

УТВЕРЖДАЮ
Зав. кафедрой ИБСТ

к.т.н., доцент

С.Л.Зефилов

28 02 2024 г.

ЗАДАНИЕ
на курсовую работу

по теме: Разработка приложений с сетевым взаимодействием.

1 Дисциплина: Сетевые технологии.

2 Вариант задания: 8.

3 Студент: Гусев Д. А.

4 Группа: 21ПИ1.

5 Цель: разработка сетевого приложения с архитектурой клиент-сервер.

6 Исходные данные на проектирование:

- требования к сетевому взаимодействию:
 - используемый протокол транспортного уровня – TCP;
 - в протоколе прикладного уровня должны быть предусмотрены возможности обмена информацией о возникающих ошибках при сетевом взаимодействии;
- требования к программе-клиенту:
 - программа-клиент должна обеспечивать передачу текстовых файлов на сервер если версия программы-сервера равна 1;
 - программа-клиент должна обеспечивать передачу бинарных файлов на сервер если версия программы-сервера равна 2;
 - в программе-клиенте должна быть реализована возможность определения версии программы-сервера;
 - в программе-клиенте должна быть предусмотрена возможность регистрации нового пользователя в программе-сервере ;
 - в программе-клиенте должна быть предусмотрена обработка ошибок, возникающих при сетевом взаимодействии;
- требования к программе-серверу:
 - программа-сервер должна сохранять текстовый файл, полученный от программы клиента, если версия программы-сервера равна 1;
 - программа-сервер должна сохранять бинарный файл, полученный от программы клиента, если версия программы-сервера равна 2;
 - программа-сервер должна сообщать программе-клиенту об ошибке, если передаваемый файл не соответствует версии программы-сервера;
 - в программе-сервере должна выполняться проверка IP-адреса программы-клиента – сохранение файла должно выполняться, если IP-адрес находится в диапазоне разрешенных адресов;

— в программе-сервере должна быть реализована возможность одновременного подключения к серверу нескольких программ-клиентов;

— в программе-сервере должна быть предусмотрена обработка ошибок, возникающих при сетевом взаимодействии;

— максимальное количество одновременно подключенных программ-клиентов – 3;

— при попытке подключения программы-клиента сверх максимального количества, ей должно возвращаться сообщение об ошибке подключения;

— должна быть реализована аутентификация пользователя программы-клиента при подключении к серверу;

— должен быть разработан протокол прикладного уровня, предназначенный для реализации сетевого взаимодействия между программой-клиентом и программой-сервером;

— должна быть проведена проверка работоспособности программ;

— должно быть разработано руководство пользователя.

7 Структура проекта:

7.1 Пояснительная записка (содержание работы):

— разработка протокола прикладного уровня;

— программная реализация клиента и сервера;

— проверка работоспособности разработанных программ;

— разработка руководства пользователя.

7.2 Графическая часть:

UML-диаграммы протокола прикладного уровня.

7.3 Экспериментальная часть:

Тестирование разработанных программ.

8 Календарный план выполнения работы:

— оформление задания	к 2 неделе семестра;
— разработка протокола прикладного уровня	к 5 неделе семестра;
— программная реализация клиента и сервера	к 11 неделе семестра;
— проверка работоспособности программ	к 13 неделе семестра;
— разработка руководства пользователя	к 14 неделе семестра;
— оформление отчета	к 15 неделе семестра.

Задание получил «8» февраля 2024 г.

Студент

Руководитель

Д. А. Гусев

О. В. Липилин

Разработка протокола прикладного уровня

Были разработаны диаграммы состояний сервера и клиента. Диаграммы представлена на рисунках 1 и 2. Была разработана таблица запросов-ответов между сервером и клиентом. Результат представлен на таблице 1.

Таблица 1

запрос клиент — сервер	ответ сервер — клиент
запрос на подключение	успешное подключение
	ошибка подключения
	неверный запрос
запрос на аутентификацию	успешная аутентификация
	неверный логин или пароль
	неверный запрос
запрос на регистрацию	успешная регистрация
	пользователь уже существует
	неверный запрос
запрос на передачу файла	успешная передача файла
	неверный тип файла
	неверный запрос

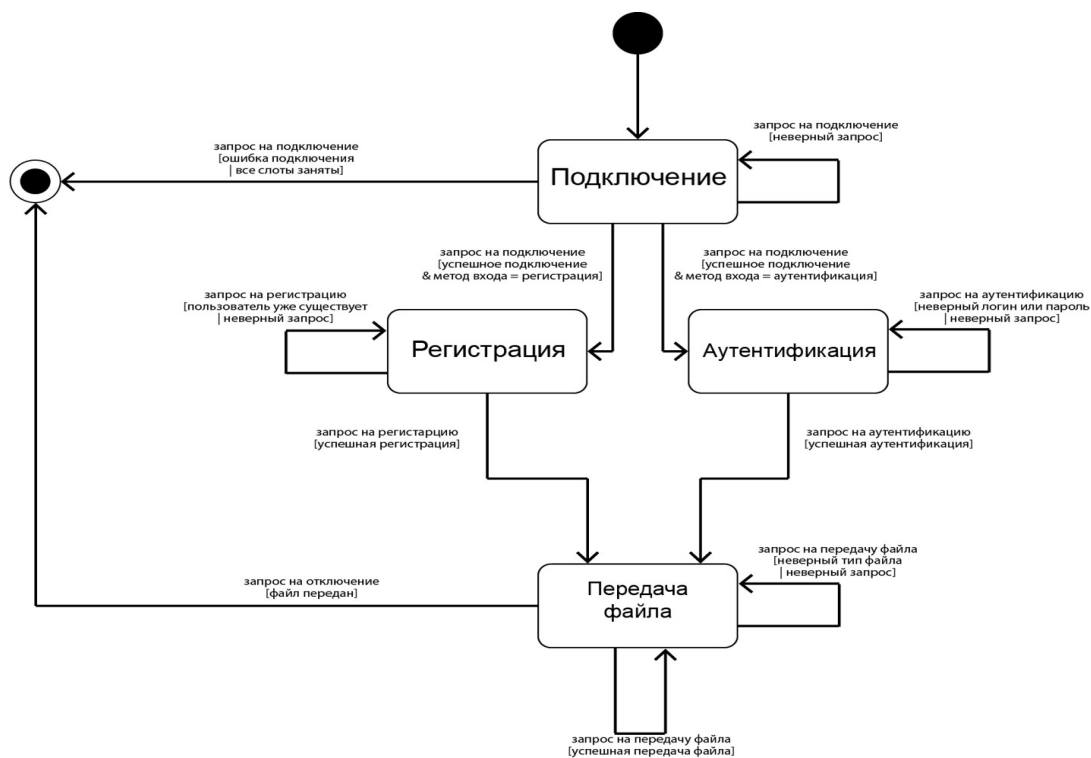


Рисунок 1 — Диаграмма состояний клиента

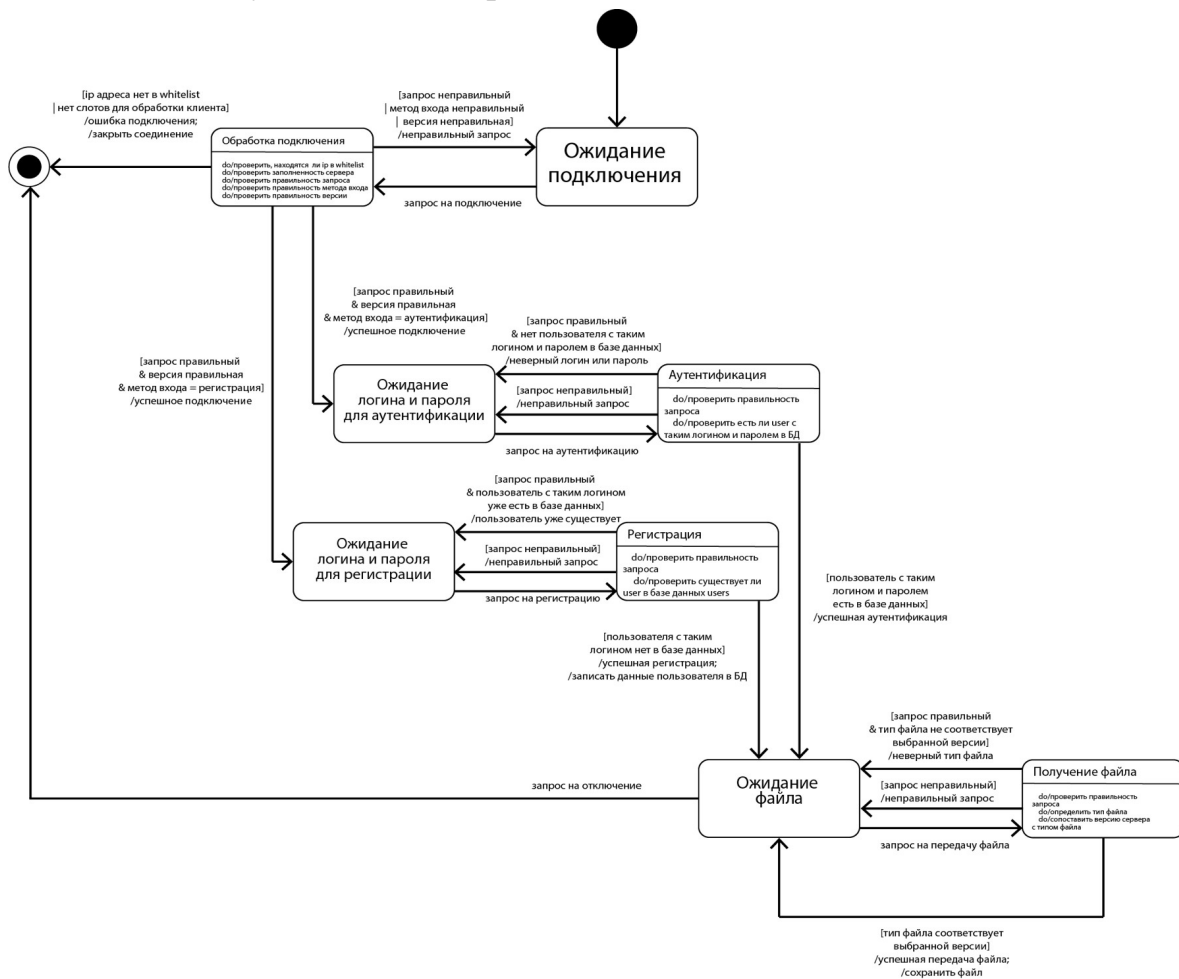


Рисунок 2 — Диаграмма состояний сервера

Были разработаны диаграммы последовательностей сетевого взаимодействия клиента и сервера. Диаграммы правильного взаимодействия представлены на рисунках 3 и 4. Диаграммы взаимодействия с ошибкой подключения представлены на рисунках 5, 6, 7. Диаграммы взаимодействия с ошибкой аутентификации и регистрации представлены на рисунках 8, 9, 10 и 11. Диаграммы взаимодействия с ошибкой передачи файла представлены на рисунках 12, 13, 14 и 15.

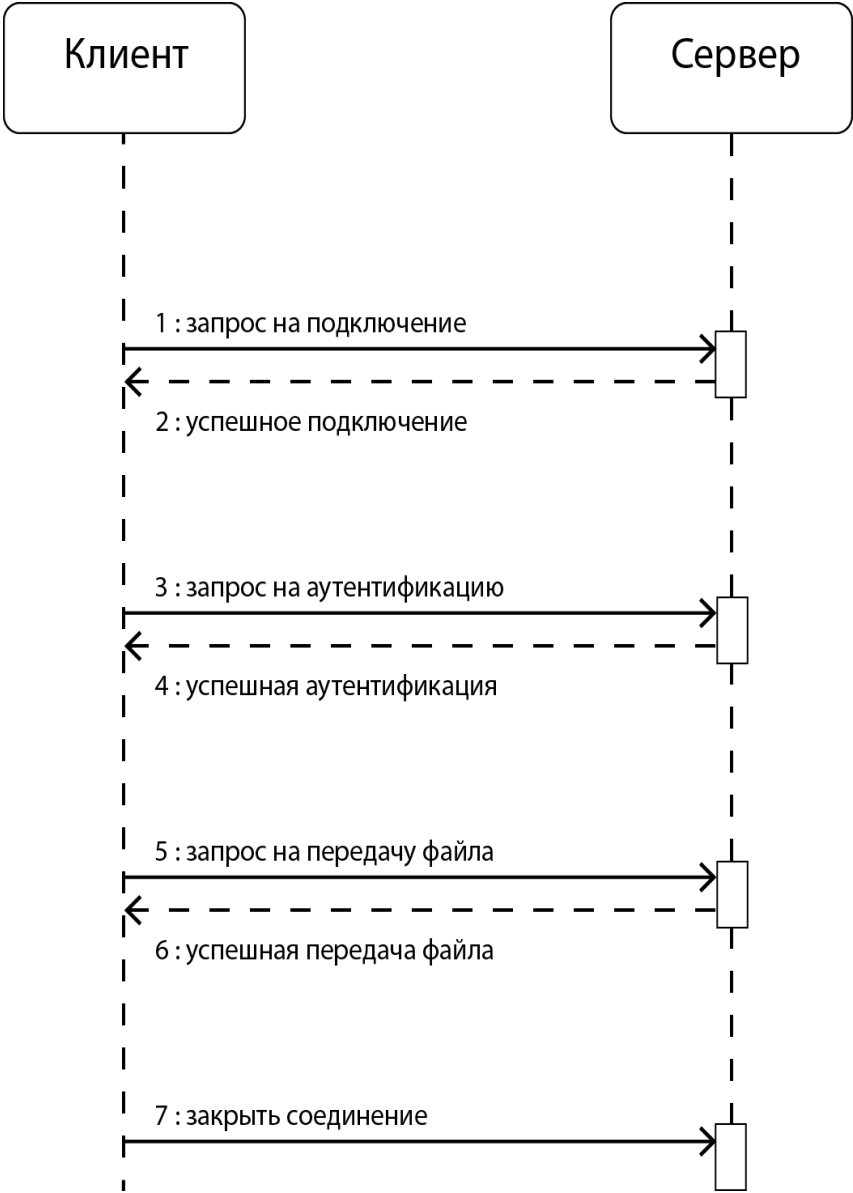


Рисунок 3 — Правильное взаимодействие (аутентификация)



Рисунок 4 — Правильное взаимодействие (регистрация)

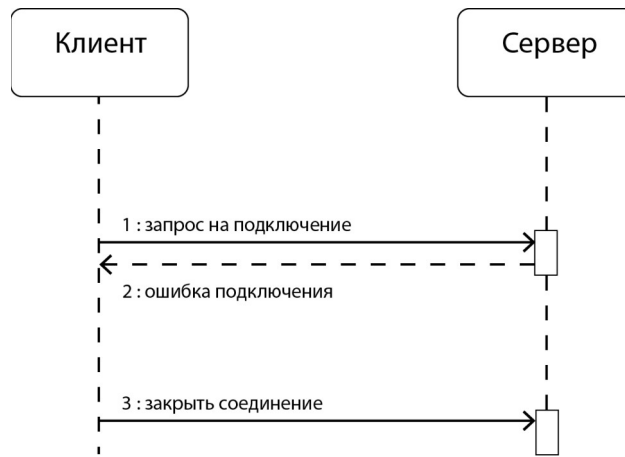


Рисунок 5 — Ошибка подключения



Рисунок 6 — Неверный запрос при подключении (аутентификация)



Рисунок 7 — Неверный запрос при подключении (регистрация)

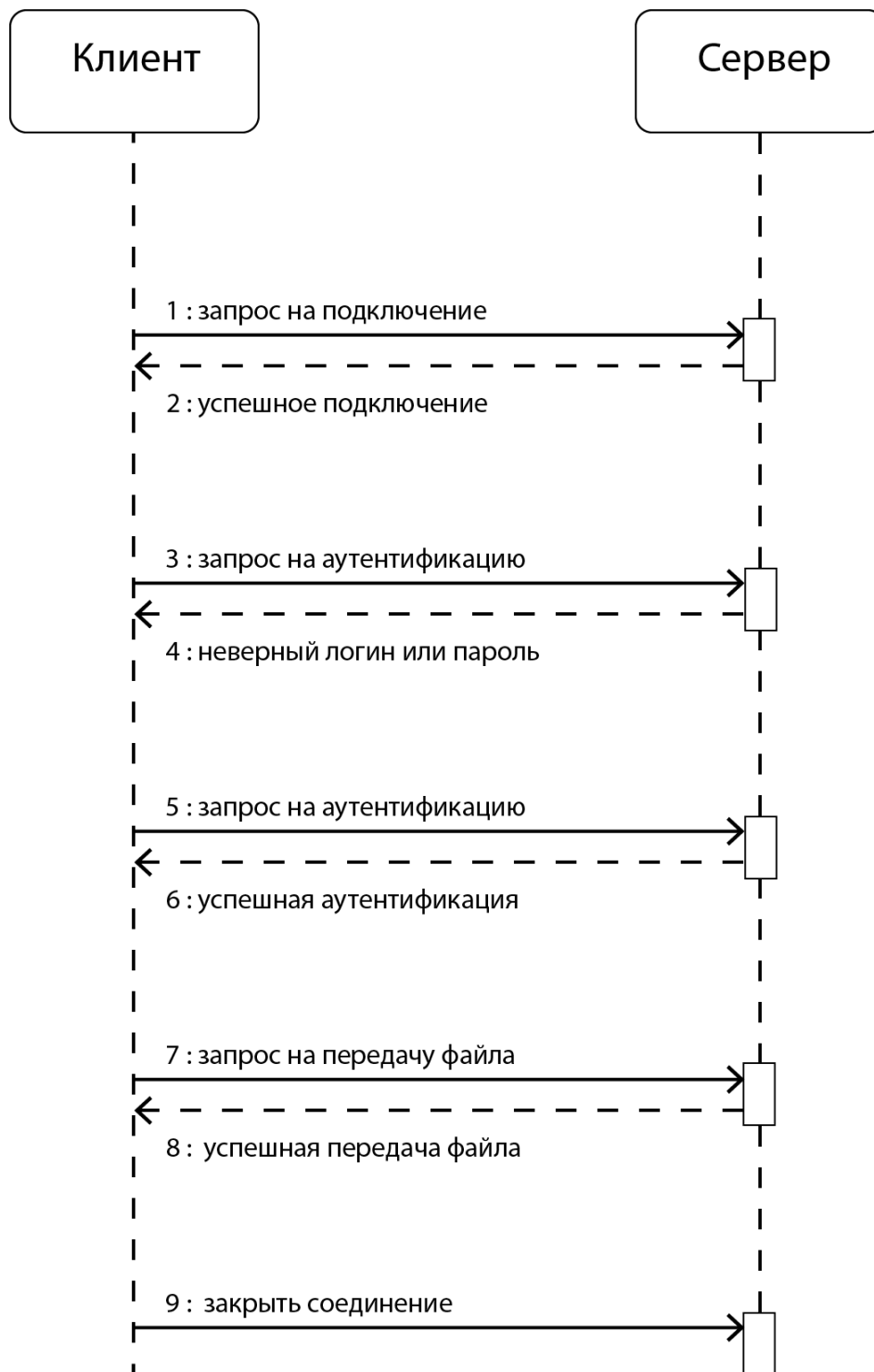


Рисунок 8 — Неверный логин или пароль (аутентификация)

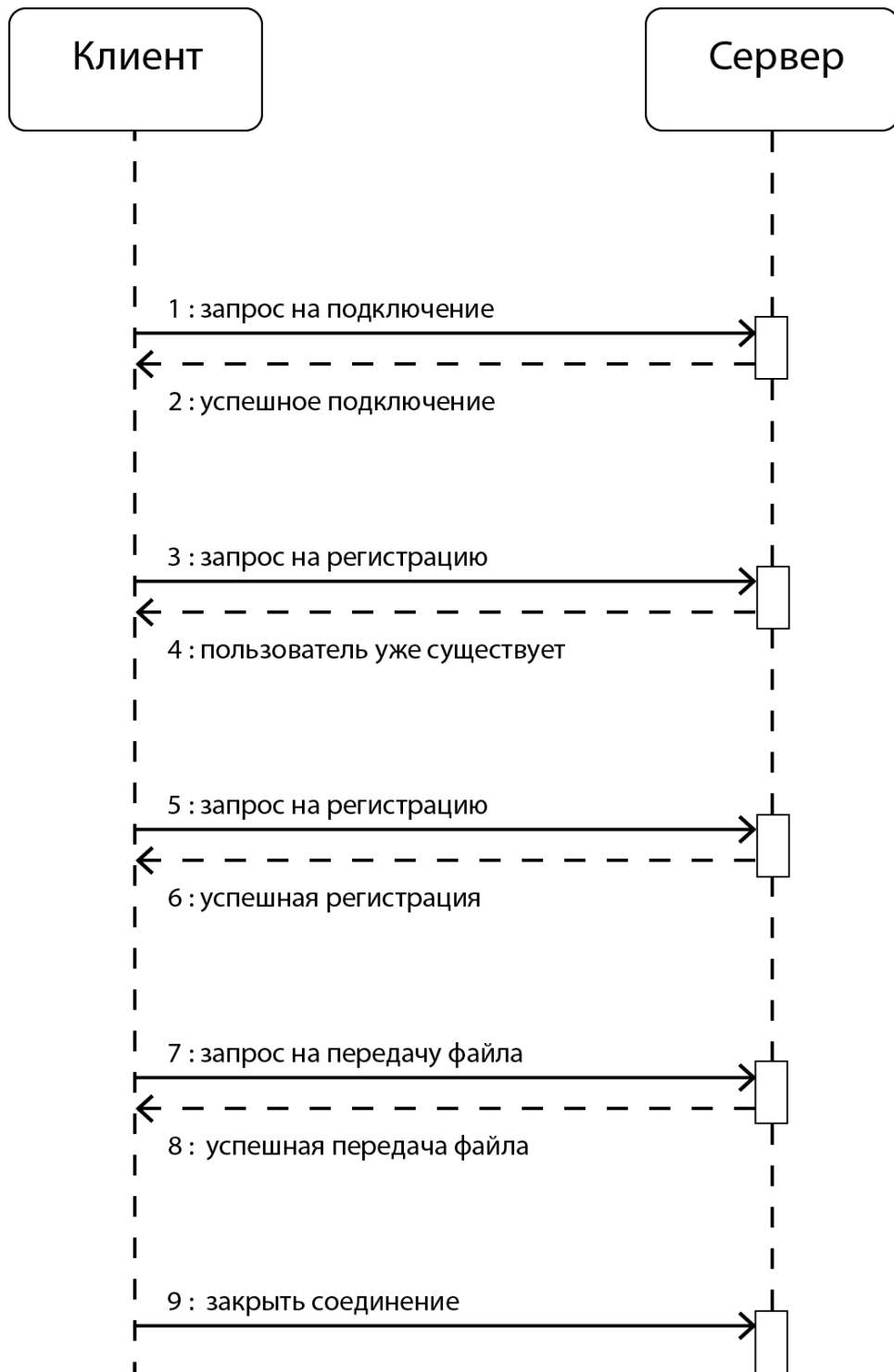


Рисунок 9 — Пользователь уже существует (регистрация)

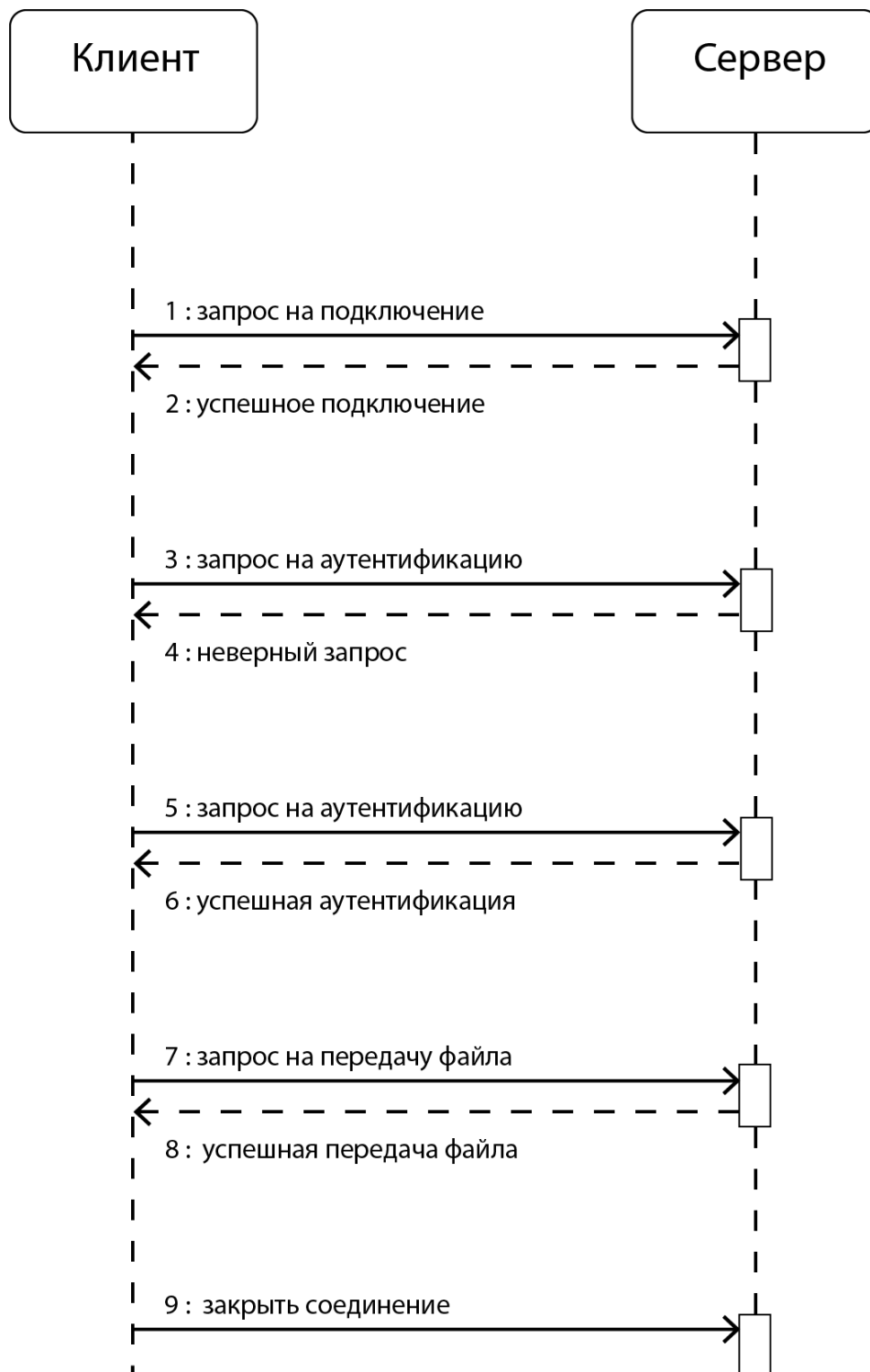


Рисунок 10 — Неверный запрос при аутентификации

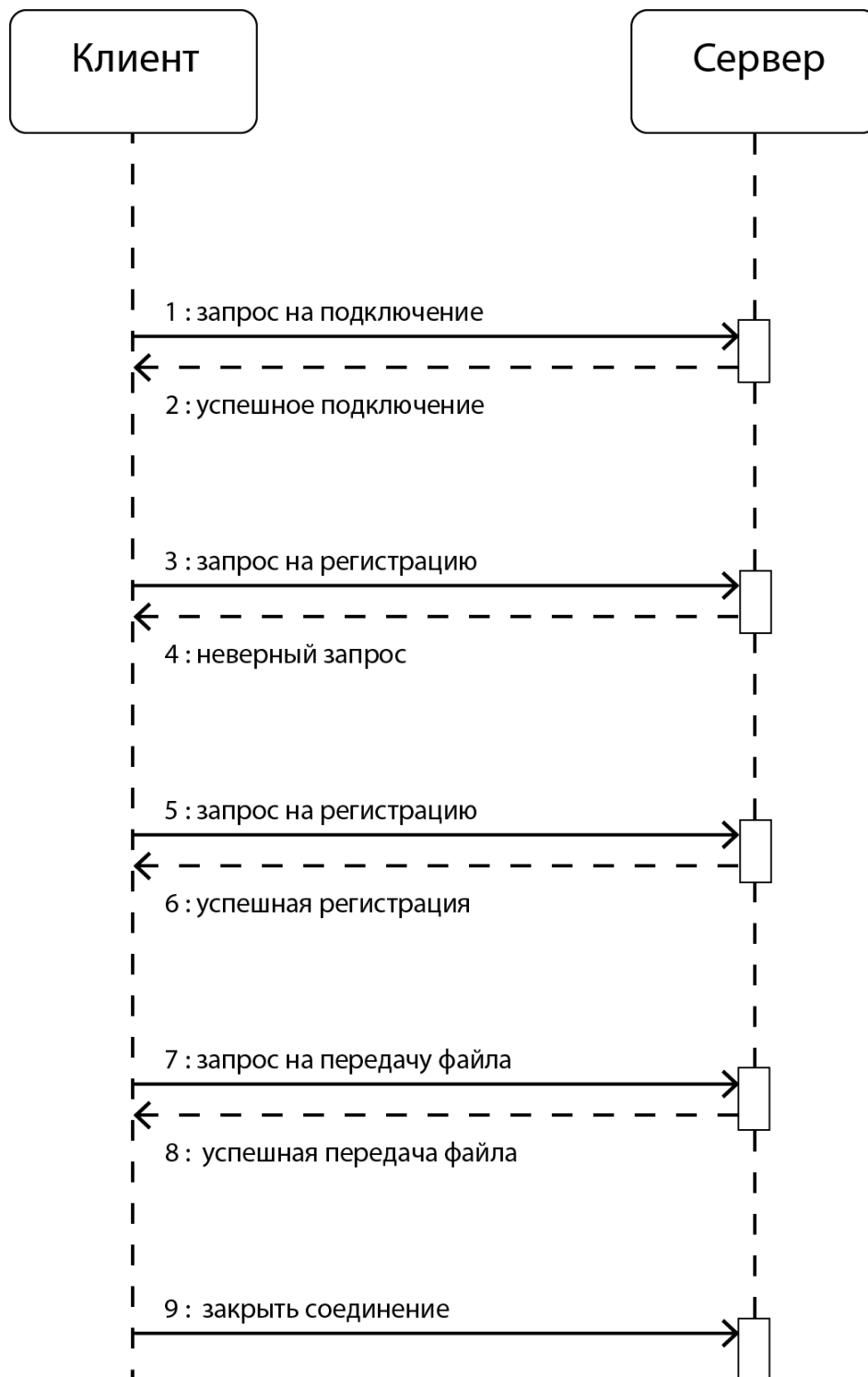


Рисунок 11 — Неверный запрос при регистрации

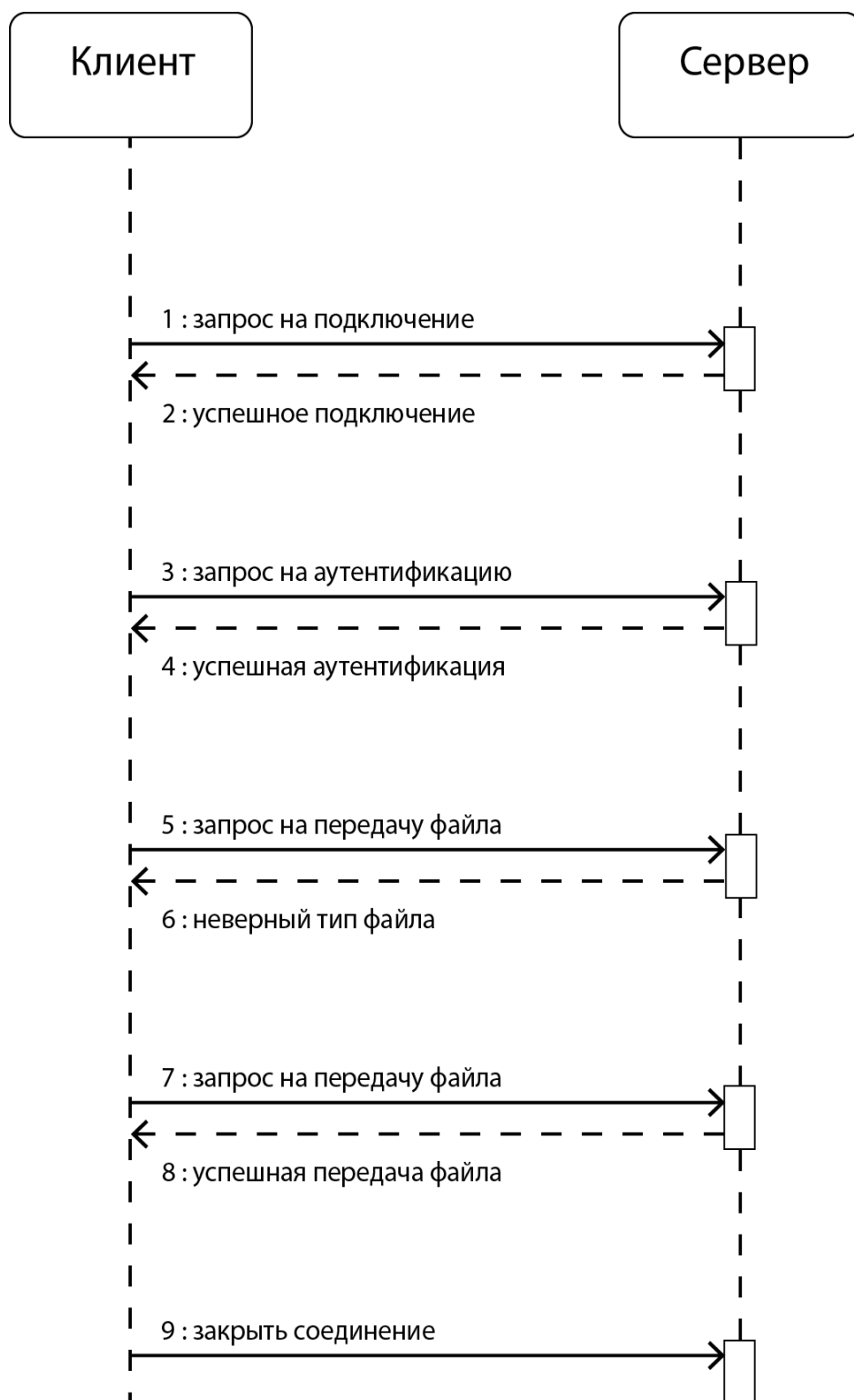


Рисунок 12 — Неверный тип файла (аутентификация)



Рисунок 13 — Неверный тип файла (регистрация)

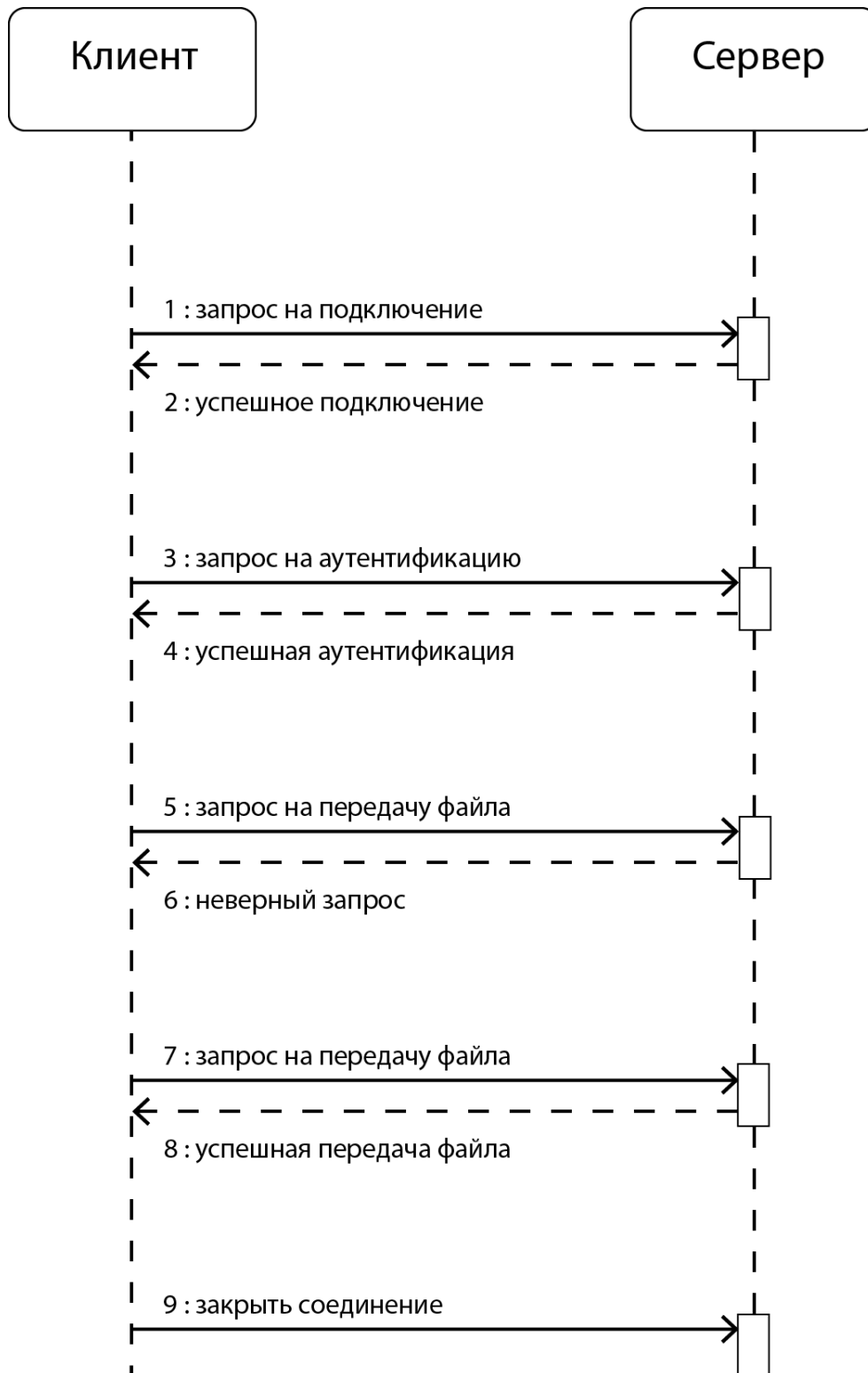


Рисунок 14 — Неверный запрос при передаче файла (аутентификация)



Рисунок 15 — Неверный запрос при передаче файла (регистрация)

Был разработан формат сообщений.

Название: запрос клиента на подключение к серверу

Формат:

connect
user_agent: <значение> separator: <значение> method: <значение> separator: <значение> version <значение>

Значения полей:

connect – заголовок запроса, строка символов в кодировке UTF-8;

separator: <значение> - строка байт, поле значение содержит разделитель сообщений (4 байта);

user_agent: <значение> - строка символов в кодировке UTF-8, поле «значение» содержит идентификатор программы-клиента (от 1 до 32 символов английского алфавита), используется для определения сессии каждой программы-клиента;

method: <значение> - строка символов в кодировке UTF-8, поле «значение» содержит информацию о методе входа (от 3 до 4 символов английского алфавита), используется для определения метода входа (регистрация или аутентификация).

version: <значение> - целое число типа int, поле «значение» содержит информацию о версии программы сервера (1 или 2);

Название: запрос клиента на аутентификацию

auth
login: <значение> separator: <значение> password: <значение>

Формат:

Значения полей:

auth – заголовок запроса, строка символов в кодировке UTF-8;

separator: <значение> - строка байт, поле значение содержит разделитель сообщений (4 байта);

login: <значение> - строка символов в кодировке UTF-8, поле «значение» содержит username пользователя (от 4 до 16 символов);

password: <значение> - строка символов в кодировке UTF-8, поле «значение» содержит пароль пользователя (от 6 до 64 символов);

Название: запрос клиента на регистрацию

reg
login: <значение> separator: <значение> password: <значение>

Формат:

reg – заголовок запроса, строка символов в кодировке UTF-8;

separator: <значение> - строка байт, поле значение содержит разделитель сообщений (4 байта);

login: <значение> - строка символов в кодировке UTF-8, поле «значение» содержит username пользователя (от 4 до 16 символов);

password: <значение> - строка символов в кодировке UTF-8, поле «значение» содержит пароль пользователя (от 6 до 64 символов);

Название: запрос клиента на передачу файла на сервер

upload
file_name: <значение> separator: <значение> content: <значение>

Формат:

upload – заголовок запроса, строка символов в кодировке UTF-8;

separator: <значение> - строка байт, поле значение содержит разделитель сообщений (4 байта);

fileNmae: <значение> - строка символов в кодировке UTF-8, поле «значение» содержит имя файла (от 1 до 128 символов английского алфавита, кириллицы или цифр от 0 до 9);

content: <значение> - байтовая строка, содержит файл для передачи;