

Protocol					
Request type	Purpose	Format	Example	Response	
				Success	Failure
The start of communication	The communication begins with the client sending this greeting message to the server. The client must specify their name. The maximum length is 988 bytes. In response, the server sends the greeting message with the client's name back to the client. This exchange serves as the acknowledgment stage of the communication.	Hello, server! How are you? This is + name.	Hello, server! How are you? This is Marharyta	Hello, Marharyta! This is the server.	If the connection hasn't occurred, the client waits or closes their application.
	In the older version the greeting message is different. The client doesn't specify their name.	Hello, server! How are you?	Hello, server! How are you?	Hello, client! This is the server.	
INFO	Retrieve detailed information about a specified file, including its size in bytes, the date of its last modification, the absolute path to the file, and its owner permissions (read, write, execute).	INFO <file_name>	INFO "HEART.txt"	Information about the file specified (size, last modification date, permissions)	"Request denied. Specified path does not exist." - if the path is invalid or if the specified path does not refer to a regular file.
LIST	Get a list of all files in the client's directory. 4 bytes is the length of this request, since it doesn't take any parameters.	LIST	LIST	List of files in a client's deidcated directory	"The folder is empty!"
REMOVE	Remove a specific folder	REMOVE <dir_name>	REMOVE Margo	"Successful removal!"	"The directory could not be deleted!" - if something went wrong.
					"Request denied. Specified path does not exist." - if the path is invalid or if the specified path does not refer to a regular file.
DELETE	Delete a specific file.	DELETE <file_name>	DELETE "HEART.txt"	"Successful removal!"	"The file could not be deleted!" - if something went wrong.
					"Request denied. Specified path does not exist." - if the path is invalid or if the specified path does not refer to a regular file.
PUT	Upload a file to the server	PUT <file_name>	PUT "C:\Margo\HEART.txt"	"File transfer completed!"	The invalidness of the file is the client's responsibility.
GET	Request a specific file from the server	GET <file_name>	GET "HEART.txt"	"OK" - the file is ready to be transferred.	"Request denied. Specified path does not exist." - if the path is invalid or if the specified path does not refer to a regular file.
				"File transfer completed!"	