



DV-PROJEKT

ZENTRAL VERWALTETE MULTI USER TO-DO LISTE

INHALT

- Anforderungen
- Aufbau
- Server
- Client
- Kommunikation
- Funktionen
- Anwendung
- Quellenverzeichnis

ANFORDERUNGEN

- Zentrale Verwaltung der Einträge
- Abfrage und Anzeige des Erledigungsstatus und Datum
- Von mehreren Benutzern les- und bearbeitbar
- ToDos Serverseitig in einer Speicherdatei verwaltet
- Aufbereitete Benutzereingabe von Client für Server Weiterverarbeitung

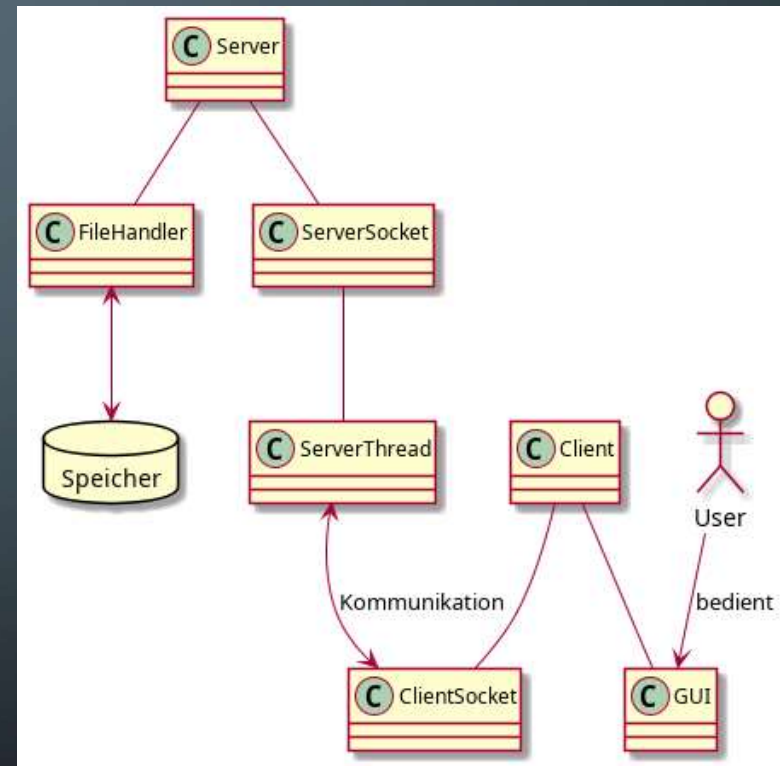
AUFBAU

Server (Linux):

- verwaltet Daten in Datenbank

Client (Windows):

- Oberfläche
- Anwendungs- bzw. Steuerungslogik



CLIENT

- Liste (mit Datum und To-Do Eintrag)
- Erledigungsbutton
- Hinzufügebutton für neue To-Do's
- Eingabemaske für neue To-Dos

KOMMUNIKATION

- Server-Client-Kommunikation über selbe „Sprache“
- Client schickt Datenanfrage
- Server liefert Daten

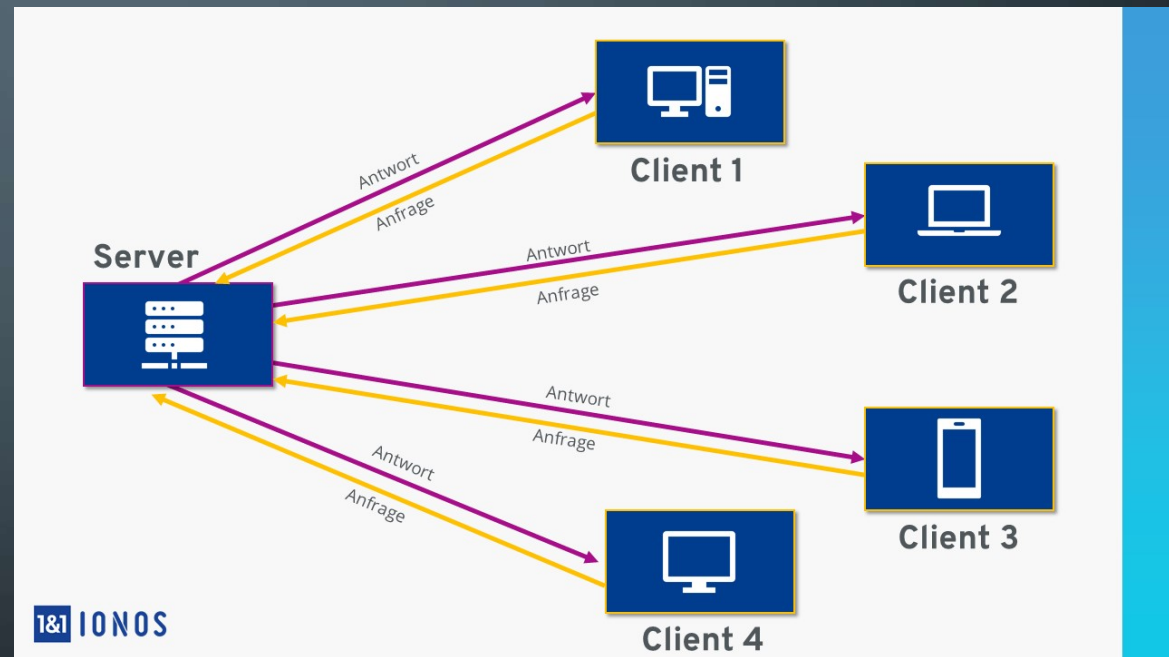
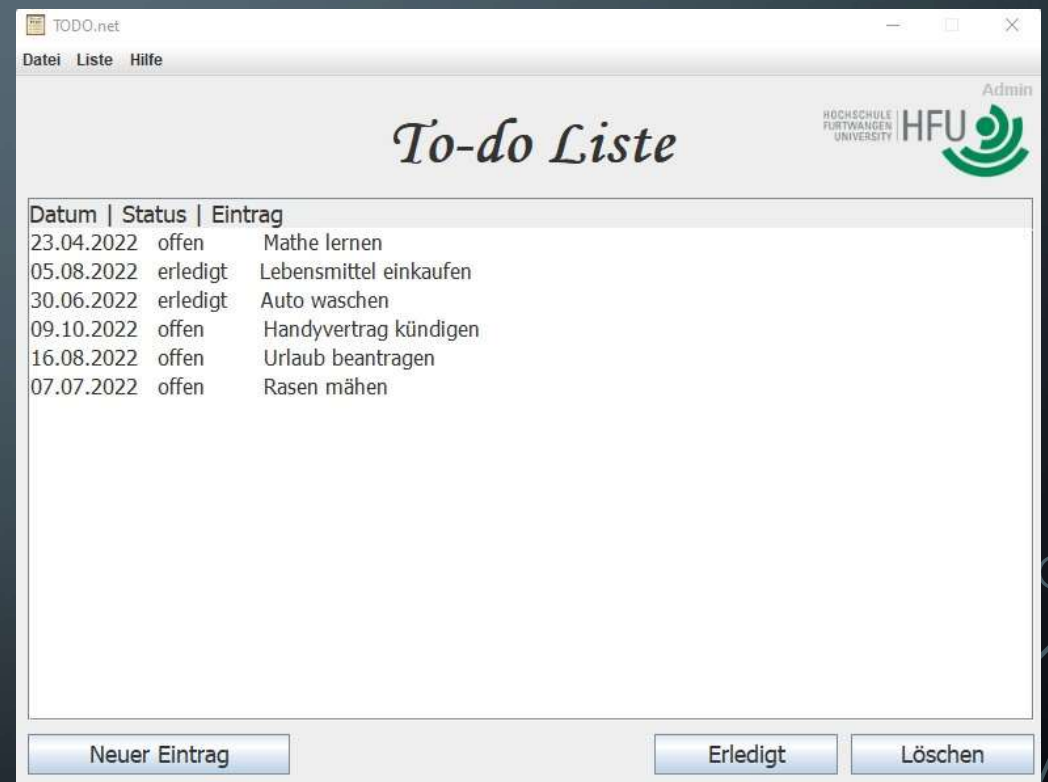


ABB. 1

FUNKTIONEN

- Erstellen
- Löschen
- Wiederherstellen
- Speichern
- Öffnen
- Drucken
- Aktualisieren
- Handbuch



ToDo.TODOs
+ statDatum(Datum : String) : boolean
+ getDate(Datum : String) : String

ToDo.DateiHandler
- dateiname : String
- writer : FileWriter
- leser : BufferedReader
- timeStamp : java.util.Calendar
+ DateiHandler(dateiname : String)
+ openDatei(write : boolean)
+ write(data : String)
+ read() : String
+ close()
+ delete()
+ getStrings(elements : int) : String[]
+ writeNewStrings(str : String[], write : boolean)
+ writeToLegacy(eventstr : String)
+ writeErr(data : String)

ToDo.ClientParser
+ setStatus(eventstr : String, erledigt : boolean) : String
+ getStatus(str : String) : boolean
+ getNewStatus(str : String) : boolean
+ getDatum(str : String) : String
+ getAdmin(str : String) : boolean

ToDo.Fenster
- list : javax.swing.JList
- index : int
- Eintrag : String
- frmToDoListe : javax.swing.JFrame
- tFtagErled : javax.swing.JTextField
- tFmonatErled : javax.swing.JTextField
- tFJahrErled : javax.swing.JTextField
- socket : java.net.Socket
- buffer : String
- authkey : String
- port : int
- eint : java.util.ArrayList
- Eintraege : javax.swing.DefaultListModel
- bufferedReader : BufferedReader
- printWriter : PrintWriter
- pars : ClientParser
- todo : TODOs
- offeneListe : String
- Adminpriv : String
- priv : boolean
- AdminBox : boolean
- Drucktext : java.util.ArrayList
- dateToStr : String
- date : java.util.Date
- counter : int
- logo : java.awt.Image, BufferedImage
- Zwischenspeicher : String
- timer : javax.swing.Timer
- Key : DateiHandler
- Port : DateiHandler
- ip : DateiHandler
+ main(args : String[])
+ Fenster()
+ initialize()
+ NeuEintrag(Datum : String)
+ leeren()
- schreibeNachricht(socket : java.net.Socket, nachricht : String)
- leseNachricht(socket : java.net.Socket) : String
+ getTodos(socket : java.net.Socket)
+ sendTodos(socket : java.net.Socket, Datum : String, Eintrag : String, Status : boolean)
+ loescheTodo(socket : java.net.Socket, i : int)
+ fuelleListe()
+ loescheListe()
+ Aktualisieren()
+ Wiederherstellen()
+ aendereStatus()
SpeichernAlle()
- saveTextalle(file : File)
SpeichernOffene()
- saveTextoffene(file : File)
Laden()
- showText(file : File)
+ drucken()
+ Zeitstempel()

The background is a dark blue gradient. In the corners, there are white, stylized circuit-like lines with small circles at the ends, resembling a network or data flow diagram.

VORSTELLUNG DER ANWENDUNG

QUELLENVERZEICHNIS

- Abb.1: Server-Client:

<https://www.ionos.de/digitalguide/server/knowhow/was-ist-ein-server-ein-begriff-zwei-definitionen/>, Abrufdatum: 16.06.22