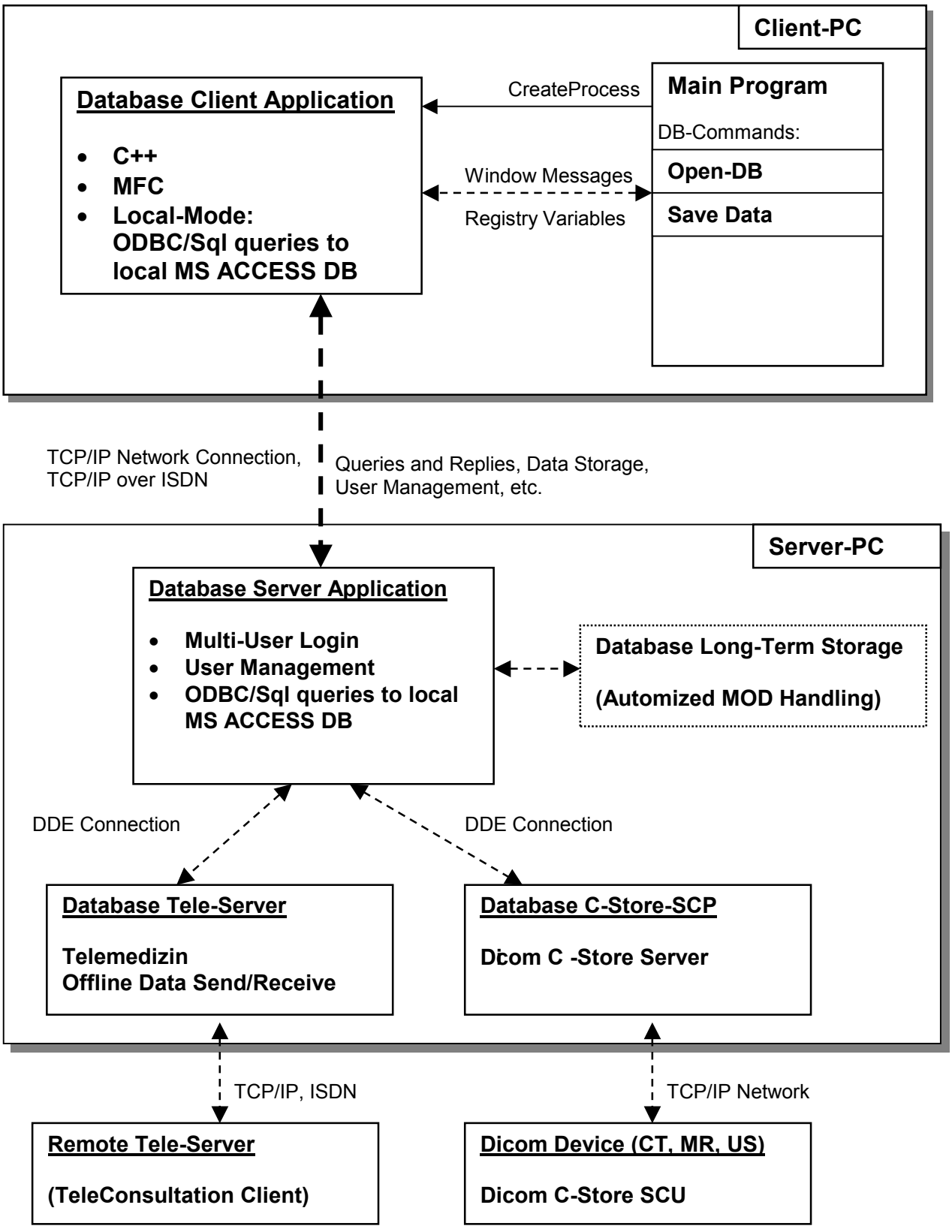
1 MedCom Client Server Database Overview



1.1 User Interface Overview (Examples)

Note: These images are taken while the software runs under german Windows-NT, therefore the UI Elements display German Text – In english NT and any other language the UI will be displayed with english text.

Image-1: Login Screen

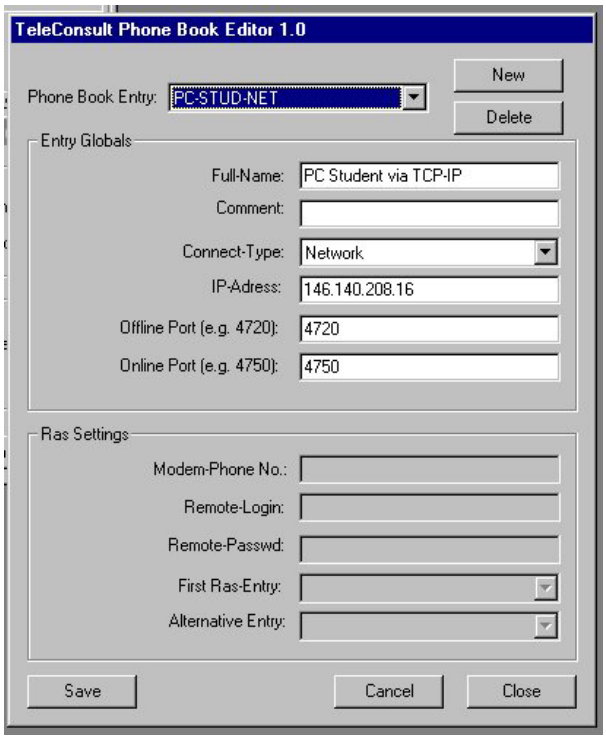


Image-2: Phonebook editor to define Server Location

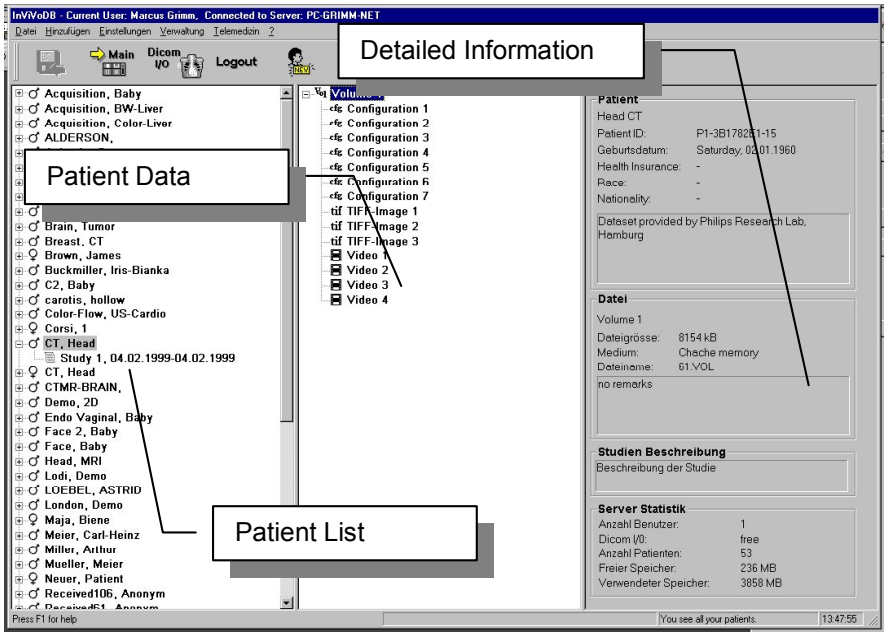


Image-3: Main Screen while connected to server

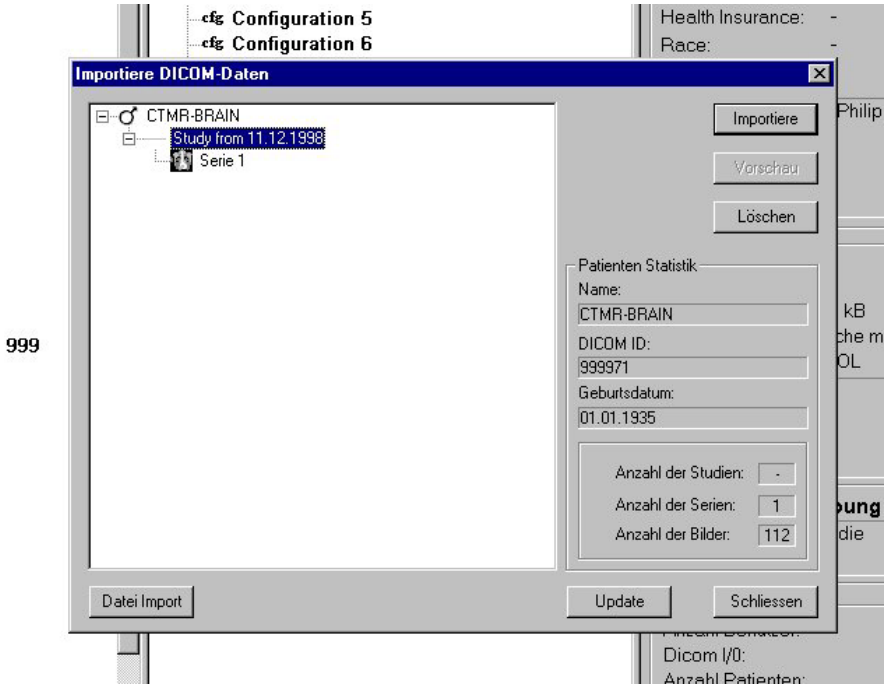


Image-4: Dicom Import Screen

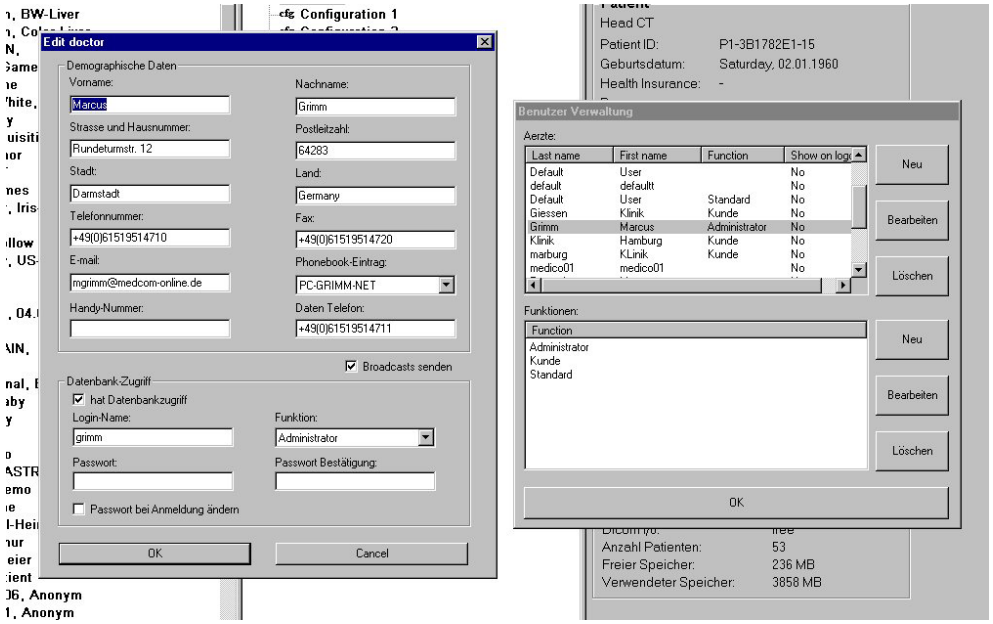


Image-5: User Manager

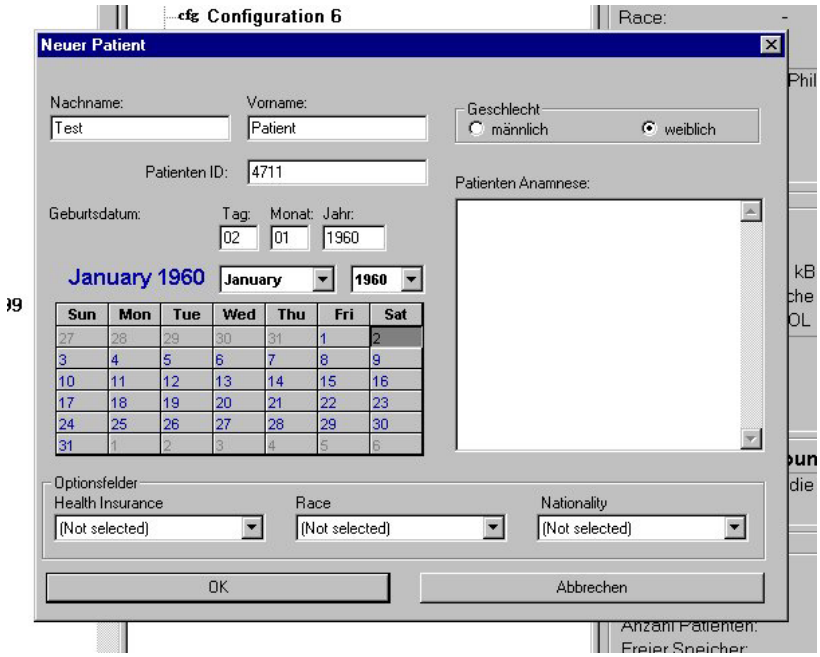


Image-6: Patient Mask

2 Communication between Database Client and Main Application

The main application (Exomio, ScanNT, etc.) starts the database during the first access (path of database executable is given in the registry database) by creating a process. If the user leave the database, the window will be hidden until the next access. The communication between the two applications is realized by Windows messages. When the user closes ScanNT a standard window message (WM_CLOSE) is send to the database to close it as well.

The database operates after its initialization in two different modes. The first mode is a loading mode. This means the user wants to load some previously acquired data to ScanNT. So the user has the possibility to select a special dataset. This mode also allows to moving data to MO and doing administrative stuff, like adding a new user.

The other mode is called saving mode. In this case the user can store a single dataset do the database. Depending on the kind of data the user has to enter a new patient name or select one out of a list. In this mode the data will be moved to the database and the database application will return control to ScanNT.

For the physical data transfer the database applications store the selected dataset to a temporary file on hard disk and write the required information (i.e. file selected, filename, etc.) to the registry.

2.1 Windows messages between Main and Client Application

The messages are defined in both applications. We distinguish now between:

	used messages	defined in module (ScanNT)	defined in class (InViVo database)
Initialisation	MC_DBINITCOMPLETE	FILE-IO fileio_init.c	CInViVoTestDoc
Loading mode	MC_PUSHDATA	FILE-IO fileio_init.c	CMainFrame
	MC_DBREADY	FILE-IO fileio_init.c	CInViVoTestDoc
Saving mode	MC_SAVEDATA	FILE-IO fileio_init.c	CMainFrame
	MC_DBREADY	FILE-IO fileio_init.c	CInViVoTestDoc
Hide Database UI	MC_HIDEDATA	FILE-IO fileio_init.c	CMainFrame
Close applications	WM_CLOSE	Standard windows message	

The communication flow is shown in the figure below:

	ScanNT		**InViVoDB**
Initialisation	Create process	---------->	Database initialise
	Activate database access	MC_DBINITCOMPLETE <----------	Send InitComplete
Loading mode	Send PushData	MC_PUSHDATA ---------->	Show database
	Wait for DBready		Wait for user interaction
			Hide database
	Evaluate user interaction	MC_DBREADY <----------	Send DBready
Saving mode	Send SaveData	MC_SAVEDATA ---------->	Ask additional informations
	Wait for DBready		Save data
	Check for success	MC_DBREADY <----------	Send DBready

Hide UI	Send HideData	WM_CLOSE →	Hide main window
Close	User close ScanNT	WM_CLOSE →	Database close

Figure 1: Message flow between ScanNT and database

During the initialization, the database writes its main window handler to the registry. By reading this entry ScanNT is able to send messages directly to the database. This has the advantage that ScanNT can determine whether the database is still running by checking the access of sending a message. The respond messages are realised as broadcasts.

2.2 Data exchange via Registry keys

ScanNT and the database application exchanges data information beside the Windows Messages via the registry as well. Therefore the following path is used:
`HKEY_CURRENT_USER/Software/MedCom/DataBase/ExChange`

The information are the location of the chosen dataset and additional application dependant settings. The table below describes the different strings used by both applications:

Key name	Description
Database action successful	The database application informs ScanNT if storing of data was successful.
DB window handle	The database stores the handler of its main application window.
Examinationdate	The database application informs ScanNT when the examination took place.
PatientName	The database application tells ScanNT the patient name.
PictureSelected	The database application informs ScanNT if any data has to be loaded.
SavePossible	ScanNT informs the database application if anything has to be stored.
FromDBName	This is the directory the database use for send data write access for database, read access for ScanNT
ToDBName	This is the directory the ScanNT use for send data read access for database, write access for ScanNT
FromDBType	Specifies the type of data to be exchanged. R/W in both applications.

3 Server Functionality

The Database Server implements virtually the same functionality as the Database Client but without any Userinterface elements. The server is controlled and respons via dedicated Network Commands between him and connected clients. The number of connected clients is theoretically limited only by the number avaialble lines in the network subsystem, however internally we currently limit the number of open lines to 25.

3.1 Client Server Communication

The Protocoll (i.e. commands) used to drive the communicatiuon between Client and Server is MedCom internal defintion. These commands are mainly:

- Commands to Query the Patient Table, Study Table, Data Table, etc.
- Commands to Download files
- Commands to Maintain the User accounts
- Commands to query the dicom Store-SCP server
- Login Mechanism
- Global Maintainance: Version Checking, Software Update handling, etc.