

Beschreibung Projekt zur OOS im SS16

Aufgabenstellung

Funktionale Anforderungen

FR1: Es soll eine Nutzerverwaltung implementiert werden, die

1. neuen Benutzern ermöglicht, sich einzutragen, wobei sie eine UserId und ein Passwort wählen können,
2. bereits eingetragenen Benutzern ermöglicht, sich unter Eingabe der UserId und des Passwortes anzumelden.

FR2: Der Zugang zum System soll sowohl remote, als auch auf dem lokalen Rechner erfolgen.

Technische Anforderungen

TR1: Das System muss auf allen Plattformen ausführbar sein.

TR2: Außerdem darf für die Zugangssysteme lediglich folgendes vorausgesetzt werden:

1. Es ist ein (java-fähiger) Web-Browser installiert.
2. Das System ist mit dem Internet verbunden.

Design- und Implementierungsentscheidungen

FR1 und FR2 implizieren, dass das System als 3-Schichten-Architektur (Client, Server und Persistenzschicht) implementiert wird:

- Client/Server-Architektur, da das System auch remote verwendet werden soll.

- Persistenzschicht, da die eingetragenen Benutzer persistent zu halten sind und

somit auch nach Wiederaufnahme des Systems verfügbar sein müssen.

Wegen **TR1**, wird als Implementierungsumgebung der Java SDK verwendet, da dieser eine möglichst große Plattformunabhängigkeit gewährleistet.

Zur Erfüllung von **TR 2** wird das Zugangssystem als Java-Applet implementiert.

Neben den Entscheidungen, die sich aus den Anforderungen ergeben, wird folgendes festgelegt:

1. Einem Systemadministrator wird ermöglicht, bereits eingetragene Benutzer zu löschen.
2. Außerdem muss die Möglichkeit bestehen, die Datenhaltung in der Persistenzschicht zu initialisieren, um ein neues System aufzusetzen.

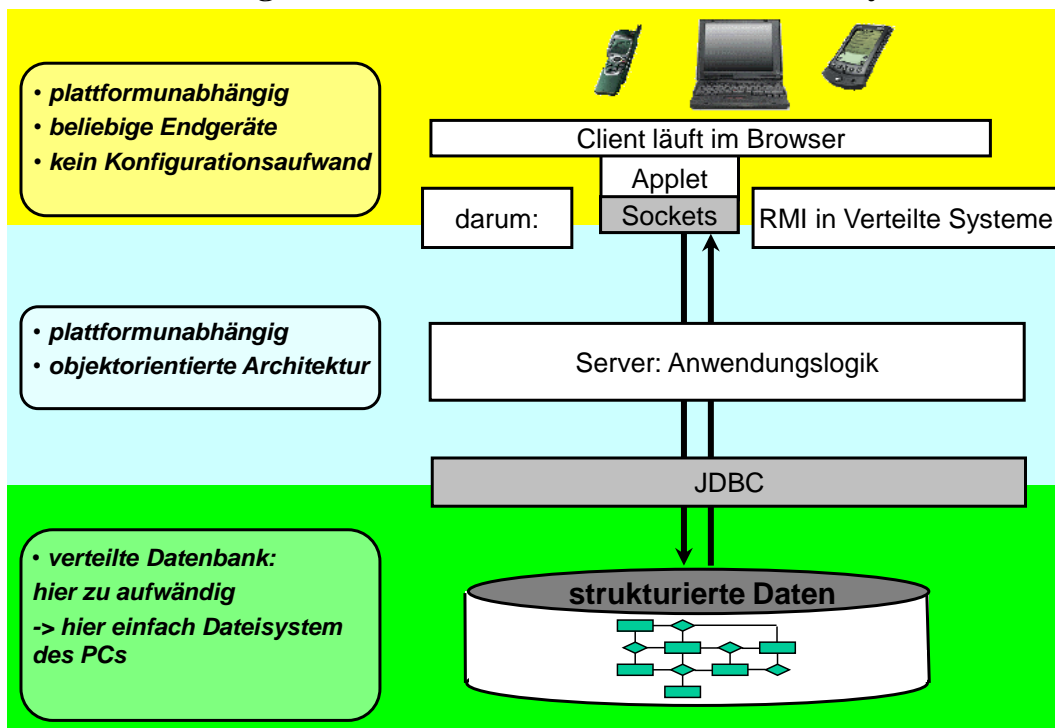
Auch diese Funktionalität darf nur einem Systemadministrator zur Verfügung gestellt werden.

Grobarchitektur

3-Schichten-Architektur

Die 3-Schichten-Architektur wird wie in der folgenden Abbildung dargestellt realisiert:

3-schichtige Architektur moderner Informationssysteme



Server: Nutzerverwaltung

Implementierung der Funktionalität, die den Benutzern zur Verfügung gestellt wird.

Ferner Implementierung der Funktionalität, die nur einem Systemadministrator zur Verfügung steht.

Persistenzschicht

Da keine Datenbank verwendet werden soll, um den Fokus auf die Verwendung von Java nicht zu verlieren, wird die Nutzerverwaltung unter Verwendung der Java-Mechanismen Serialisieren und Deserialisieren in einer Datei auf dem Dateisystem des Servers persistent gehalten.

Client: Benutzerschnittstelle

Der Client soll als Applet auf einer html-Seite implementiert sein. Hierzu wird eine Fensterumgebung unter Verwendung der Java-Swing-Klassen implementiert, die auf Systemereignisse, wie Mouse-Klick oder Eintragen von Texten in Textfeldern reagiert.

Praktikum zur OOS

Versuche 2 – 5

SoSe16

Seite 3 von 3

Kommunikation zwischen Client und Server

Um der Vorlesung "Verteilte Systeme" nicht vorzugreifen, wird nicht die nahe liegende Java-Technik RMI verwendet, sondern eine direkte Client-Server-Verbindung über Sockets implementiert.

Kommunikation zwischen Server und Persistenzschicht

Da die Persistenzschicht nicht als Datenbank realisiert wird, wird nicht die nahe liegende Java-Technik JDBC verwendet, sondern die Datei direkt durch Abspeichern von Streams persistent gehalten.

WebService

Zusätzlich soll das System als Webservice bereitgestellt werden, um einen standardisierten Zugriff und eine einfache Integration in eine Service OrientedArchitecture zu ermöglichen.