OTH HOCHSCHULE AMBERG-WEIDEN

Hochschule für Technik und Wirtschaft

Fakultät Elektrotechnik, Medien und Informatik / Master IT & AUT

Studienarbeit: Technologien verteilter Systeme

Studiengang: Master IT & AUT
Fach: Technologien verteilter Systeme
Erster Demotermin: 15.1.2018
Zweiter Demotermin: 22.1.2018
Letzter Abgabetermin: 23.1.2018
Prüfer: Prof. Dr. Josef Pösl
Abgabe: Programmdateien (elektr.+gedruckt)
Hilfsmittel: alles

Name:	
Matrikelnummer:	
Bearbeitete Aufgabe und Lösungsform [3] :	

Hinweise:

1. Geben Sie die Ihre Lösungen bitte sowohl **in gedruckter** (Programmcode) **als auch in elektronischer Form** ab. Tragen Sie Ihren Namen in jeder von Ihnen erstellten Datei als **Autorennamen** ein. Verwenden sie das Deckblatt der Angabe auch als Deckblatt für ihre Lösung und heften Sie alle Blätter der gedruckten Lösung zusammen. Die elektronische Version ihrer Lösung schicken Sie bitte per EMail in einem Archiv (bspw. 7z, kein Zip, **ohne Debug-Verzeichn., < 1MB**) an den Dozenten (EMail-Adresse: J.Poesl@oth-aw.de).
2. Tragen Sie Name und Matrikelnummer in das Deckblatt der Lösung ein!
3. Sie brauchen jeweils nur eine der Aufgaben in zwei der angegebenen möglichen Lösungsformen zu bearbeiten. Jede der Aufgaben wird dabei exklusiv von einer Gruppe (max. 2 Personen) bearbeitet, jedes Gruppenmitglied bearbeitet genau zwei Lösungsformen. Die verteilten Schnittstellen sind dazu von der Gruppe (im Vorfeld) gemeinsam festzulegen. Die **Verteilung** der Aufgaben und Lösungsformen auf die Bearbeiter erfolgt nach Ausgabe der Arbeit vorläufig am 27.11.2017 und endgültig **am 4.12.2017** in der Vorlesung.
4. Sollten sich zwischen den Gruppen ähnliche Teillösungen durch Zusammenarbeit ergeben, ist dies bei der Lösung anzugeben (und natürlich bei den „Autoren“ zu vermerken).
5. Überlegen Sie sich für die Abgabe bitte eine **kleine Demo**, mit der Sie Anwendungs- und Testfälle Ihrer Arbeit vor dem letzten Abgabetermin vorführen.

Viel Spaß bei der Studienarbeit !

Schnittstelle, Orthogonales (gemeinsam)	
Funktionalität	
Anwendung der verteilten Technik	
Programmstruktur	
Gesamtnote	

Studienarbeit: Master IT & AUT / Technologien verteilter Systeme

Zu implementieren ist eine verteilte Anwendung mit dem Visual Studio 2015, deren Funktionsumfang in einer der folgenden Aufgabenstellungen beschrieben ist. Die Aufgabenstellungen beschreiben dabei die einzelnen Funktionen die zu unterstützen sind. Die Art, in der diese Funktionen dem Benutzer angeboten werden, hängt von der jeweiligen Lösungsform ab.

Die vier möglichen Lösungsformen sind:

A. RPC-Lösung

Bei dieser Lösung ist die Verteilung über RPC zu implementieren. Die Funktionen des Clients sind in einem C/C++-Kommandozeilenprogramm anzubieten, das nach Aufruf mit der Netzwerkadresse des Servers (optionaler Aufrufparameter) gestartet wird. Das Programm soll nach dem Start mit dem Server (ebenfalls in C/C++) Kontakt aufnehmen und bei Erfolg auf der Kommandozeile Befehle entgegennehmen, bis ein Abbruchbefehl eingegeben wird. Die Befehle sind dabei durch die Funktionen gegeben, die bei den einzelnen Aufgabenstellungen beschrieben sind.

B. COM-Lösung

Diese Lösung ist analog der RPC-Lösung zu implementieren, wobei die Verteilung über COM zu realisieren ist.

C. .NET-Lösung

Bei dieser Lösung sind Client und Server in Visual C# .NET zu realisieren. Die Funktionen des Clients sollen dabei über eine benutzerfreundliche graphische Oberfläche angeboten werden. Client und Server sollten über einen der .NET-Remotingdienste (WCF oder WEB-Services) verbunden sein. Der C#-Client sollte außerdem durch Austausch des verwendeten Remoting-Objekts/-Typs (bspw. bedingte Compilierung) in der Lage sein, mit dem COM-Server aus der Lösungsform B zu interagieren.

D. WP8 .NET-Lösung

Bei dieser Lösung ist der Client in Visual C# .NET im Rahmen des Windows Phone SDK 8.0 zu realisieren, so dass die Anwendung auf einem Mobilgerät mit dem Betriebssystem Windows Phone 8.0 lauffähig ist. Die Funktionen sollen über eine benutzerfreundliche graphische Oberfläche angeboten werden. Der Server sollte auf einem Desktop-Rechner laufen und in Visual C# .NET realisiert sein. Die Kommunikation sollte über einen der .NET-Remotingdienste (mit einer anderen Alternative als unter C) erfolgen.

Anmerkungen

- Wird eine Aufgabenstellung nur von einem Studenten bearbeitet („Gruppe“ aus einer Person) und u.A. die Lösungsform C gewählt, B aber nicht, dann sind sowohl die WCF- als auch die WEB-Service-Lösung zu realisieren.
- Lösungsform D kann zunächst mit einem Emulator der SDK programmiert werden, ist aber für die Demo auf einem realen Gerät vorzuführen.
- Sollte es bei Aufgabestellungen zu unlösbaren oder schwer zu beherrschenden Hardwareproblemen kommen, kann die Aufgabenstellung nach Absprache abgeändert werden.

Folgende Aufgabenstellungen sind exklusiv zu bearbeiten:

1. MyBAY

Zu implementieren ist eine verteilte Anwendung zur Verwaltung und Durchführung von Auktionen auf einem Auktionsserver. Folgende Funktionen sollen auf Client-Seite angeboten werden:

a. Zugang: **`user`** <Name> <Passwort>

   Damit wird überprüft, ob der Benutzer berechtigt ist, diese Form des Rechnerzugangs zu erhalten. Der Server führt eine Liste von gültigen Benutzern und Passworten (Verwaltung bspw. über Textdatei) und gestattet die Ausführung der übrigen Befehle nur nach erfolgter Autorisierung. Beachten Sie dabei, dass sich im Allgemeinen mehrere Clients zum Server verbinden können und jeweils individuell autorisiert werden müssen. Achten Sie außerdem darauf, alle Funktionen auf Serverseite so abzusichern, dass ein Aufruf ohne vorherige Autorisierung nicht möglich ist.

b. Einstellen eines Artikels: **`offer`** **`"`**<Artikelname>**`"`** <Mindestgebot>

   Einstellen eines Artikels zur Auktion unter Angabe eines Mindestgebots für den Verkauf. Die dadurch eröffnete Auktion erhält auf Serverseite eine eindeutige Nummer („Auktionsnummer“). Einige der folgenden Funktionen (*) dürfen nur vom Anbieter für die jeweilige Auktion aufgerufen werden.

c. Statusverfolgung eines Artikels: **`interested`** <Auktionsnummer>

   Automatische Verfolgung des Status der angegebenen Auktion: Gebote und Abschluss der Auktion werden automatisch auf Clientseite angezeigt, zusammen mit der jeweiligen Gebotssumme. Die Verfolgung endet mit Ende der Auktion. Das Einstellen eines Artikels (s. 1.b) impliziert automatisch (ohne Aufruf von `interested`) eine derartige Statusverfolgung, wobei in diesem Fall zusätzlich der Name des jeweiligen Bieters angezeigt wird.

d. Auflisten der Auktionen: **`listauctions`** [`-a` | `-A`] [<Artikelnameteil>]

   Auflistung aller Auktionen deren Status verfolgt wird, zusammen mit der Auktionsnummer, dem Artikelnamen, der Zahl der aktuellen Gebote und dem Höchstgebot. Bei Angabe der Option „`-a`“ werden alle Artikel in offenen Auktionen angezeigt, bei Angabe von „`-A`“ zusätzlich die Artikel in bereits beendeten Auktionen. Durch Angabe der ersten Buchstaben eines Artikelnamens kann die Ausgabe auf diejenigen Artikel beschränkt werden, deren Namen mit der entsprechenden Buchstabenfolge beginnt.

e. Gebot: **`bid`** <Auktionsnummer> <Geldbetrag>

   Abgeben eines Gebots zur angegebenen Auktionsnummer. Alle an der Auktion Interessierten (s. 1.c) werden über das Gebot informiert.

f. Auktionsdetails*: **`details`** <Auktionsnummer>

   Ausgabe aller Bieter zu einer gegebenen Auktionsnummer, zusammen mit Ihren jeweiligen Geboten in chronologischer Reihenfolge.

g. Abschluss einer Auktion*: **`endauction`** <Auktionsnummer>

   Beendigung einer Auktion, wobei alle an der Auktion Interessierten (s. 1.c) dreimal kurz hintereinander über dieses Ende informiert werden und damit nochmals die Möglichkeit einer zwischenzeitlichen Gebotsabgabe bekommen. Nach Abschluss der Information und der Annahme evtl. weiterer Gebote werden der zustandegekommene Preis und der Käufer ausgegeben.

h. Abbruchbefehl: **`bye`** (Client-Befehl)

   Beendigung des Clients.

2. MyCASINO

Zu implementieren ist eine verteilte Anwendung zum Betreiben eines Kasinos. Folgende Funktionen sollen auf Client-Seite angeboten werden:

a. Zugang: **`user`** <Name> <Passwort>

   Siehe **`user`** bei MyBAY. Ein Benutzer (Name: „Casino“) nimmt dabei die Rolle des Kasinobetreibers ein und darf als Einziger die Funktionen (*) zur Annahme einer Einzahlung und zum Ziehen der Gewinnzahlen ausführen. Zu einer Zeit darf höchstens ein Kasinobetreiber eingeloggt sein. Ist kein Kasinobetreiber eingeloggt, ist das Abgeben von Wetten (s. Aufgabenteil 2.c) nicht möglich. Benutzer werden bei Eingabe eines beliebigen Kommandos darüber informiert, wenn kein Betreiber eingeloggt ist.

b. Einzahlung*: **`payin`** <Name> <Geldbetrag>

   Einzahlung eines ganzzahligen Geldbetrags (bspw. in Euro) für einen Benutzer. Das Guthaben des Benutzers erhöht sich um diesen Betrag.

c. Wetten: **`bet`** <Geldbetrag> <Zahl1> <Zahl2>

   Setzen eines Geldbetrags auf zwei (ganze) Zahlen zwischen 1 und 5, wobei Zahl1 < Zahl2. Wetten dürfen vom gleichen Benutzer mehrfach abgegeben werden und bleiben gleichzeitig bestehen, wenn sie für unterschiedliche Zahlen abgegeben wurden. Werden die gleichen Zahlen angegeben, die beim Benutzer schon für eine Wette stehen, wird der bisherige Geldbetrag überschrieben. Bei einem Geldbetrag von 0 wird eine ggf. bestehende Wette für die gegebenen Zahlen gelöscht. Eine Wette wird nur dann akzeptiert, wenn im Falle eines Gewinnes (s. Aufgabenteil 2.e) das Guthaben des Kasinos ausreichend für die Begleichung der Wettschuld ist.

d. Wettanzeige: **`showbets`**

   Auflistung aller Wetten, die derzeit für alle Benutzer bestehen, zusammen mit den möglichen Gewinnen (s. Aufgabenteil 2.e).

e. Zahlen „ziehen“*: **`draw`** [<Zahl1> <Zahl2>]

   Ziehen zweier zufälliger unterschiedlicher Zahlen zwischen 1 und 5. Zu Testzwecken können Zahlen optional als Parameter übergeben werden. Stimmt eine Zahl einer abgegebenen Wette mit einer der gezogenen Zahlen überein, erhält der Spieler den doppelten Einsatz zurück. Hat ein Spieler zwei „Richtige“, erhält er zusätzlich den kompletten gesetzten Geldbetrag aller bestehenden Wetten. Nach „Auszahlung“ der Wettbeträge bzw. nach Gutschrift des verbleibenden Wetteinsatzes auf das Kasinokonto oder dessen Belastung werden alle Wetten gelöscht.

f. Statusanzeige: **`showstatus`**

   Anzeige aller bisherigen Wetten des Benutzers zusammen mit den gezogenen Zahlen, dem jeweiligen Gewinn bzw. Verlust und dem jeweiligen aktuellen Guthaben. Außerdem werden alle Einzahlvorgänge angezeigt.

g. Abbruchbefehl: **`bye`** (Client-Befehl)

   Beendigung des Clients. Dabei werden alle Wetten des Benutzers zurückgenommen. Handelt es sich beim Benutzer um den Kasinobetreiber, werden alle Wetten geschlossen und die Spieler können keine Wetten mehr abgeben.