

Fach: Verteilte Informationsverarbeitung
Studienheft: VII01-03
Projekt: Client-Server-Malprogramm
Dokumenttitel: Phase 1: Analyse und Definition
Dokumentart: Analyse/ Definition
Erstellt von: Martin Blankenstein
Erstelldatum: 26.08.2010

Inhaltsverzeichnis:

1 Einleitung.....	2
2 Kundenanforderungen.....	2
2.1 Beschreibung des gewünschten Produktes	2
2.2 Rückfragen an den Kunden.....	2
3 Anforderungsanalyse anhand des Kundenwunsches und der Rückfragen	3
3.1 Bedingungen an die Umgebung:.....	4
4 Eigene Anforderungen	4
4.1 Eigene äußere Anforderungen	4
4.2 Eigene innere Anforderungen	4

1 Einleitung

Als Client-Server-Anwendung habe ich mir ein einfaches Malprogramm überlegt, in dem mehrere Leute gleichzeitig zeichnen können. Dabei soll es einen Server geben der den Malbereich und die Beteiligten verwaltet, sowie einen Client, der die Oberfläche für den Zeichenvorgang darstellt.

Die Erstellung der Anforderungen erarbeite ich durch ein Kundengespräch. Da es für die erstellte Software keinen realen Auftraggeber gibt, wird das Kundengespräch fiktiv durchgeführt. Das (virtuelle) Kundengespräch erlaubt es mir Rückfragen zu stellen und die Anforderungen zu konkretisieren.

2 Kundenanforderungen

2.1 Beschreibung des gewünschten Produktes

Eine Software mit der mehrere Personen gleichzeitig an einem einfachen Bild malen können. Das Malen soll in einer Oberfläche erfolgen, in der man mit der Maus Linien zieht. Jede beteiligte Person soll dabei eine Farbe haben. malt eine Person an ihrem Rechner einen Strich, so soll dieser Strich bei den beteiligten Personen erscheinen. Somit ist das gemeinsame Malen an einer gemeinsamen Zeichnung möglich. In der Anwendung sollen alle gerade beteiligten Personen mit einem Alias und ihrer Farbe angezeigt werden.

Beim Start der Anwendung soll erst eine Auswahl erscheinen, ob man sich bei der malerischen Tätigkeit eines anderen beteiligen möchte, oder ob man eine neue Zeichnung beginnen möchte. Zusätzlich soll beim Start ein Name und eine Farbe ausgewählt werden.

Wenn man sich entschließt eine neue Malerei zu beginnen, soll man die Größe der Leinwand in Pixeln angeben. Die Leinwand soll zu Beginn weiß sein.

Entschließt man sich zur Beteiligung an einer Malerei, so muss man den Namen oder die IP des Rechners angeben sowie den Port, wo die Malerei gestartet wurde. Ist unter dem angegebenen Ziel eine Malerei aktiv, so erfolgt die Beteiligung an dieser. Beteiligt man sich an einer Malerei, soll die Leinwand in der für den Server angegebenen Größe erstellt und mit dem aktuellen Inhalt gefüllt werden.

2.2 Rückfragen an den Kunden

- Darf sich jeder in einem Netzwerk ohne Authentifizierung an gemeinsamen Malereien beteiligen?
☐ Ja
- Wie viele Personen sollen gleichzeitig an einer Malerei arbeiten können?
☐ max. 10
- Darf eine Person mehrere Malereien starten?
☐ Ja

- Darf eine Person gleichzeitig an mehreren Malereien beteiligt sein?
 - ☐ Ja
 - ☐ Darf eine Person sich mehrmals an einer Malerei beteiligen?
 - ☐ Ja
 - ☐ Darf es mehrere beteiligte Personen mit der gleichen Farbe und/oder dem gleichen Alias geben?
 - Ja
- Wie groß darf die Leinwand maximal und wie groß muss sie mindestens sein?
 - ☐ max. 1000*500
 - ☐ min. 500*200
- Darf man die Größe nach dem Beginn der Malerei ändern?
 - ☐ Nein
- Soll man die Malerei speichern und für einen späteren Start verwenden können?
 - ☐ Nein
- Soll man die Möglichkeit besitzen ein Foto der aktuellen Leinwand zu machen um es in einer Datei zu speichern?
 - ☐ Ja
- Besteht das einzige Malmittel aus einer Analogie zu dem Stift in Paint?
 - ☐ Ja
 - ☐ Eine Art Radiergummi ist somit ebenfalls nicht gewünscht?
 - ☐ Ja, nur Malen und Übermalen, nicht Löschen
- Ist ein Farbwechsel während des Malen gewünscht?
 - ☐ Nein
- Sind unterschiedliche Strichstärken gewünscht?
 - ☐ Nein
- Auf welchen Betriebssystemen soll die Anwendung ausgeführt werden können?
 - ☐ Windows XP, Windows7
- Was halten Sie von "PaintTogether" als Namen für die Anwendung?
 - ☐ in Ordnung, nehmen wir

3 Anforderungsanalyse anhand des Kundenwunsches und der Rückfragen

Server:

- Start mit angegebener Größe (500-1000)*(200-500) unter einem Aliasnamen und Port
- Besitzt aktuelles Gesamtbild für neue Clients
- Kennt von Clientverbindungen den Aliasnamen sowie die Farbe
- Muss bis zu 10 gleichzeitige Clientverbindungen unterstützen
- Muss Malen von Clients empfangen und an die verbundenen Clients weiterreichen (und für Gesamtbild merken)

Client:

- GUI mit Malbereich
- Malen mit einem einfarbigen Stift im Malbereich

- Anzeige aller mit dem Server verbundener Clients inkl. Alias und Farbe

Startanwendung:

- Abfrage des Alias und der Farbe vom Anwender
- Abfrage ob eigene Malerei oder Beteiligung
 - ☐ bei eigener Malerei
 - ☐ Größe der Leinwand für Server vom Anwender abfragen
 - ☐ Port für den Server abfragen (als Standardport soll Port 6969 definiert sein)
 - ☐ Server mit der Größe der Leinwand und dem Alias des Nutzers starten
 - ☐ weiter wie bei "bei Beteiligung", nur das als Zielrechner schon "localhost" intern festgelegt wird
 - ☐ bei Beteiligung
 - ☐ Abfrage des Zielrechner (Standardeingabe "localhost") inkl. Port (Standardport anzeigen) vom Anwender
 - ☐ Client mit Farbe, Alias, Zielrechner und Port starten

Bedeutung für die grobe Architektur:

Es gibt im Ergebnis drei ausführbare Anwendungen:

- Server mit Parametern für Leinwandgröße und Aliasnamen
- Client mit Parametern für Alias, Farbe, Servername/IP, Port
- Startanwendung für einfachen Start von Server und Client

3.1 Bedingungen an die Umgebung:

- stabile Netzwerkverbindung
- Windows XP/7 als Betriebssystem
 - ☐ Zur Ausführung der Anwendung muss der Anwender das Recht haben der Anwendung die für die Kommunikation nötigen Rechte zu geben
- .Net-Framework 3.5 als Basis

4 Eigene Anforderungen

4.1 Eigene äußere Anforderungen

Damit man besser sieht, was der Server überhaupt macht, möchte ich für den Server eine Windowskonsole erstellen. Sie soll keine Befehle verarbeiten und nur die primären Tätigkeiten des Servers loggen/ausschreiben. Diese umfassen:

- Client-Verbindungs-Auf- und -abbau
- Malaktion durch einen Client

4.2 Eigene innere Anforderungen

Die inneren Anforderungen dienen der Codequalität und damit vor allem der Evolvierbarkeit und Verständlichkeit der Software/ des Quellcodes.

- hohe Testabdeckung der funktionalen Komponenten durch automatische Unit-Tests
- Einsatz eines Versionkontrollsystems wie Subversion