

Webbasiertes Interface zu multimedialem Contact-Center

Studienarbeit

23. Dezember 2011

Autor	Eric Aebi
Betreuer	Prof. Beat Stettler
Projektpartner	Luware AG, Zürich

Inhaltsverzeichnis

0.1	Änderungsgeschichte	6
1	Abstract	7
2	Projektplan	8
2.1	Projektübersicht	8
2.1.1	Zweck und Ziel	8
2.1.2	Annahmen und Einschränkungen	8
2.2	Projektorganisation	8
2.2.1	Externe Schnittstellen	8
2.3	Management Abläufe	9
2.3.1	Projekt Kostenvoranschlag	9
2.3.2	Projektplan	9
2.3.2.1	Iterationsplanung / Meilensteine	9
2.3.2.2	Besprechungen / Meetings	9
2.3.2.3	Releases	9
2.4	Risikomanagement	9
2.5	Arbeitspakete	10
2.6	Infrastruktur	10
3	Technologieanalyse	11
3.1	Requirements	11
3.1.0.4	Plattformunabhängig	11
3.1.1	Plugins	11
3.1.2	Datenübertragung	11
3.1.3	Anbindung an MS Lync	11
3.2	Technologien	11
3.2.1	Serverseitige Technologien	12
3.2.1.1	ASP.NET	12
3.2.1.1.1	Anbindung MS Lync	12
3.2.1.2	Java	12
3.2.1.2.1	Anbindung MS Lync	12
3.2.1.3	PHP	12
3.2.1.3.1	Ajax	12
3.2.1.3.2	Anbindung MS Lync	13
3.2.2	Clientseitige Technologien	13
3.2.2.1	Java-Applets	13

3.2.2.2	Java Script	13
3.2.2.2.1	Ajax	13
3.2.2.3	Adobe Flash	13
3.2.2.4	HTML5	14
3.3	Fazit	14
3.4	Quellen	14
4	Anforderungsspezifikation	15
4.1	Allgemeine Beschreibung	15
4.1.1	Produkt Perspektive	15
4.1.2	Produkt Funktion	15
4.1.2.1	Clientseite	15
4.1.2.2	Serverseite	15
4.1.3	Benutzer Charakteristik	15
4.1.4	Einschränkung	15
4.1.5	Abhängigkeiten	15
4.2	Spezifische Anforderungen	16
4.2.1	Nichtfunktionale Anforderungen	16
4.2.1.1	Bedienbarkeit	16
4.2.1.2	Stabilität	16
4.2.1.3	Design	16
4.2.1.4	Anpassbarkeit	16
4.2.1.5	Schnittstellen	16
4.2.1.6	Verwendete Standards	16
4.3	Use Cases	17
4.3.1	Use Case Diagramm	17
4.3.2	Aktoren & Stakeholders	17
4.3.3	Use Cases Brief	17
4.3.3.1	UC01: Kontakt aufnehmen	17
4.3.3.2	UC02: Kunde führen	17
4.3.3.3	UC03: Callback Funktion	18
4.3.3.4	UC04: Kunde wechselt Webseite	18
4.3.4	UC01: Kontakt aufnehmen	19
4.3.5	UC02: Kunde führen	20
4.3.5.1	UC03: Callback Funktion	21
4.3.6	UC04: Kunde wechselt Webseite	22
5	Management Summary	23
5.1	Ausgangslage	23
5.1.1	Vorgehen, Technolgien	23
5.1.2	Ergebnisse	23
5.1.3	Ausblick	24

6	Software Architektur	25
6.1	Architektonische Darstellung	25
6.2	Architektonische Ziele & Einschränkungen	25
6.2.1	Ziele	25
6.3	Physische Sicht	25
6.4	Logische Sicht	26
6.5	Designentscheidungen	27
6.5.1	Datenhaltung	27
6.5.2	Design Packages	28
6.5.2.1	Package Client.Html	28
6.5.2.1.1	Klassen Diagramm	28
6.5.2.1.2	Beschreibung des Packages	28
6.5.2.1.3	Sequenzdiagramm Start Session	28
6.5.2.1.4	Sequenzdiagramm Send Message	29
6.5.2.1.5	Sequenzdiagramm Receive Message	30
6.5.2.1.6	Kernoperationen	30
6.5.2.1.6.1	Client.StartSession(String firstMesssage):void	30
6.5.2.1.6.2	Client.SendImMessage(String message):void	31
6.5.2.1.6.3	Client. InstantMessageReceived (InstantMessageReceivedNotification):void	31
7	GUI Analyse	32
7.1	Allgemeine Beschreibung	32
7.2	Paperprototypes	32
7.2.1	Einstieg	32
7.2.2	WebchatControl	32
7.2.3	Eingabemaske für Telefonnummer	33
7.2.4	Chat in eigenem Fenster	33
8	Technischer Bericht	35
8.1	Übersicht & Aufgabenstellung	35
8.2	Ergebnisse	35
8.3	Schlussfolgerungen	36
9	Persönlicher Bericht	37
9.1	Allgemeiner Rückblick	37
9.2	Gelernte Lektionen	37
10	Test Dokumentation	38
10.1	Voraussetzungen	38
10.2	Vorbereitungen	38
10.3	Systemtests	39
10.3.1	Test vom 16.12.2011	39

10.3.2 Test der GUI Funktionalität 22.12.2011	40
11 Benutzeranleitung	41
11.1 Wie kann ich meine Frage stellen?	41
11.2 Chat in eigenem Fenster öffnen	42
11.3 Telefonanruf auslösen	42
12 Installationsanleitung	44
12.1 Zweck	44
12.2 Installation der Webapplikation	44
12.2.1 Erstellung einer neuen Applikation	44
12.2.2 Deploy der Source Dateien	46
12.3 Installation nur des WebChatControls	46
12.3.1 Globale Registrierung des Tags	47
13 Glosar	49

0.1 Änderungsgeschichte

Datum	Version	Änderung	Autor
02.12.2011	1.0	Erstellung des Dokuments	ea
17.12.2001	1.1	Überarbeitung der Kapitel	ea

1 Abstract

Abteilung	Informatik
Name des Studierenden	Eric Aebi
Studienjahr	HS 2011/2012
Titel der Studienarbeit	Webbasiertes Interface zu multimedialem Contact Center
Examinator	Prof. Beat Stettler
Themengebiet	Internet-Technologien und -Anwendungen
Projektpartner	Luware AG, Zürich
Institut	Institut für vernetzte Systeme

Innerhalb dieser Studienarbeit wurde ein Web-Frontend für das bestehende Customer Contact Center der Firma Luware erarbeitet. Dabei handelt es sich um einen Webchat, welcher serverseitig in ASP.Net bzw. C# geschrieben wurde. Clientseitig wurden, um möglichst alle Kunden zu erreichen, sowie um die Abhängigkeit von Plug-ins zu vermeiden, nur Standard Webtechnologien (HTML, CSS) verwendet. Die Webapplikation kommuniziert via der bestehenden Consumer Integration von Luware mit einem MS Lync Server und ermöglicht zusätzlich zum Chat ein „co-browsing“. Dazu wird die URL der aktuell vom Kunden geöffneten Webseite übergeben, welche dann im Lync des Agenten angezeigt wird. Klickt der Agent auf einen Link, wird dies dem Webfrontend mitgeteilt und der Kunde kann dem Agenten folgen. Zudem kann der Kunde seine Telefonnummer angeben. Diese wird ebenfalls dem Lync Server mitgeteilt, welcher dann den Kunden zurückruft und ihn telefonisch mit dem Agenten verbindet. Diese Funktion ist aber auf Seite von Luware noch nicht vollständig implementiert. Als Vorarbeit für zukünftige Versionen wurde noch versucht, eine Videoübertragung zu implementieren, welche dann im Browser mittels HTML5 dargestellt wird. Leider ergaben sich hier aber einige Probleme mit den aktuellen browserspezifischen Implementationen des noch nicht verabschiedeten HTML5 Standards.

2 Projektplan

2.1 Projektübersicht

In dieser Arbeit soll ein Webfrontend für das bereits bestehende Contact Center der Firma Luware AG erstellt werden. Dieses soll mit offenen Technologien realisiert werden. Um eine möglichst grossen Teil der Kundschaft, unabhängig von der installierten Software und Plattform, erreichen zu können.

2.1.1 Zweck und Ziel

Ziel dieser Arbeit ist es, ein Contact Center, basierend auf Web Technologien zu implementieren, mit welchem Kunden mit dem Support der jeweiligen Webseite in Kontakt treten können. Dies ist auf mehrere Arten vorstellbar, wie zum Beispiel via Chat, IP-Telefonie, Video-Chat oder Desktop-Sharing, wobei in dieser Arbeit nicht zwingend alle Möglichkeiten implementiert werden sollen. Um die Kontaktaufnahme dem Kunden einfach zu gestalten, soll die Applikation möglichst ohne eine Installation von einer Clientsoftware oder Plugins auskommen. Zudem soll sie Plattform unabhängig sein und wenn möglich auch Mobile-Devices unterstützen.

2.1.2 Annahmen und Einschränkungen

Vom 28.11.2011 bis 16.12.2011 ist es Eric Aebi nur beschränkt möglich am Projekt zu arbeiten, da er in den Militärdienst muss. Er wird jedoch für die Schultage Urlaub beantragen, somit sollten sich die Einschränkung in vertretbaren Grenzen halten.

2.2 Projektorganisation

Das Projekt wird von Eric Aebi, einem Studenten der HSR Hochschule für Technik in Rapperswil, erarbeitet.

2.2.1 Externe Schnittstellen

- Betreuer ist Herr Prof. B. Stettler
- Industriepartner ist die Luware AG in Zürich
 - Ansprechpartner ist Philipp Beck

2.3 Management Abläufe

2.3.1 Projekt Kostenvoranschlag

Das Projekt startet am 19. September 2011 und soll am 23.12.2011 abgeschlossen werden. In dieser Zeitspanne stehen pro Projektmitarbeiter ca. 16h pro Woche zur Verfügung. Das Gesamtbudget setzt sich zusammen aus $8\text{ECTS} * 25\text{h} * 1 \text{ Teammitglied} = 200 \text{ Stunden}$.

2.3.2 Projektplan

2.3.2.1 Iterationsplanung / Meilensteine

Meilenstein	Titel	Resultat	Datum
MS1	Technologie Analyse	Abschluss Technologie Analyse	05.10.2011
MS2	Anforderungen und Analyse	Requirements Specification, Use-Cases, Domain Analyse	19.10.2011
MS3	Ende Eloberation	Architektur Prototyp, Release 0.1	02.11.2011
MS4	Architektur Design	Release 0.1b	23.11.2011
MS5	Abgabe	Release 1.0	23.12.2011

2.3.2.2 Besprechungen / Meetings

Es sollen in regelmässigen Abständen Meetings mit allen Parteien abgehalten werden. Wenn möglich sollte es min. ein Meeting pro Meilenstein geben. Die Termine hierfür müssen jeweils im Voraus abgemacht werden. Falls spezifische Fragen geklärt werden müssen, sind auch spontane Meetings zwischen den einzelnen Parteien möglich.

2.3.2.3 Releases

Release	Typ	Requirements
0.1	Prototyp	UI-Prototyp, Chatfenster
0.1b	Beta	Verbindung zu MS Lync, Anzeige der Aktuellen URL des Clients
1.0	Final	Interaktion mit MS Lync

2.4 Risikomanagement

Siehe Risikomanagement.xls

2.5 Arbeitspakete

Siehe Arbeitspakete.xlsx

2.6 Infrastruktur

Um das Projekt zu bewältigen wird mit dem persönlichen Laptop, sowie dem von der HSR zur Verfügung gestellten Desktop Rechner gearbeitet. Für die Entwicklung sowie die Tests soll die Umgebung der Firma Luware AG benützt werden können. Der Zugriff auf die Entwicklungsumgebung erfolgt via einem VPN Client und Remote Desktop. Der Entwicklungsrechner ist eine Virtual Machine mit Windows Server 2008. Zudem ist darauf Visual Studio 2010 Ultimate installiert, welches an einen TFS von Luware angebunden ist. Dieser soll für die Source Code Verwaltung genutzt werden. Luware stellt für die Entwicklung ebenfalls die restliche benötigte Umgebung zu Verfügung. Z.B. einen MS Lync Server, sowie die Lync Client Applikation. Die Anbindung an den MS Lync Server soll mit dem Consumer Integration API von Luware bewerkstelligt werden.

Die Zeiterfassung wird mit MS Excel gemacht.

3 Technologieanalyse

3.1 Requirements

Hier werden die Requirements an die eingesetzte Technologie beschrieben, damit ein möglichst objektiver Vergleich ermöglicht wird.

3.1.0.4 Plattformunabhängig

Die Technologie soll auf möglichst vielen Plattformen lauffähig sein. Um mit der aktuellen Entwicklung mithalten zu können, wäre auch die Unterstützung von mobilen Systemen wünschenswert.

3.1.1 Plugins

Die Applikation soll "out of the box" funktionieren. Der Kunde soll nicht gezwungen werden, ein Plugin zu installieren um die Applikation nutzen zu können, denn die Akzeptanz eines Contact Center könnte stark darunter leiden, wenn zuerst auf dem Client Software installiert werden muss. Dies gilt es deshalb zu vermeiden.

3.1.2 Datenübertragung

Der Kunde soll, zu einem späteren Zeitpunkt, entscheiden können, ob er mittels eines Chats oder per Sprache kommunizieren möchte. Um die Sprachübertragung zu ermöglichen, werden Codecs benötigt. Welche werden von der Technologie zur Verfügung gestellt? Welche zwingend benötigt? Die selbe Frage stellt sich bei der Videoübertragung. Die Unterstützung solcher Datenübertragung ist wünschenswert, auch wenn die erste Version der Applikation dies nicht unterstützen wird.

3.1.3 Anbindung an MS Lync

Das Webfrontend muss mit der MS Lync Umgebung der Firma Luware interagieren können. Daher ist die Möglichkeit einer Anbindung unabdingbar. Diese Anbindung muss stabil laufen und sollte möglichst direkt erfolgen, um keine Performance Verluste in Kauf nehmen zu müssen.

3.2 Technologien

In diesem Kapitel sollen zuerst kurz die serverseitigen Technologien diskutiert werden. Die clientseitigen werden später behandelt.

3.2.1 Serverseitige Technologien

Grundsätzlich gilt für alle serverseitigen Technologien, dass eine plattformunabhängige Implementation möglich ist, da die Übersetzung bzw. die Interpretation des Codes auf dem Server bewerkstelligt wird. Der Client erhält nur die, meist aus HTML oder CSS bestehende, Antwort.

3.2.1.1 ASP.NET

Mit ASP.NET können unter Verwendung von objektorientierten Sprachen wie C# und VB.NET dynamische Webseiten erstellt werden. Damit steht eine Vielzahl von Umsetzungsmöglichkeiten zur Verfügung.

3.2.1.1.1 Anbindung MS Lync Für ASP.NET gibt es von Microsoft eine API welche den Zugriff auf MS Lync ermöglicht, damit ist eine direkte Anbindung sichergestellt. Es kann zudem auch davon ausgegangen werden, dass Microsoft das MS Lync SDK an neue Versionen von MS Lync anpassen wird. Somit ist auch in Zukunft die Verbindung zum MS Lync Server sichergestellt.

3.2.1.2 Java

Java bietet ebenfalls eine Vielzahl von Möglichkeiten dynamische Webseiten zu erstellen, wie zum Beispiel Servlets, JSP oder auch Applets.¹ Mit diesen Technologien ist eine objektorientierte Entwicklung möglich.

3.2.1.2.1 Anbindung MS Lync Der Zugriff wäre eventuell via JNI machbar. Dies würde allerdings einen erheblich höheren Aufwand bedeuten, da dies zuerst in einem proof of technology gezeigt werden müsste. Zudem hat JNI einen relativ grossen Overhead pro Aufruf nativer Methoden. Es ist möglich, dass hier Performance Probleme auftreten. Auch dies müsste vorerst in einem Test widerlegt werden.

3.2.1.3 PHP

PHP ist eine weit verbreitete Technologie zur Entwicklung von Webanwendungen. Objektorientierte Entwicklung ist zwar möglich, Kritiker erheben allerdings den Vorwurf, dass viele Bibliotheken noch prozedural gehalten sind. Weitere Kritikpunkte sind das fehlende threading sowie der nicht vorhandene Bytecode-Cache, was zur Folge hat, dass der PHP Code bei jedem Aufruf von neuem übersetzt werden muss. Dieser Bytecode-Cache kann jedoch mit Erweiterungen nachgerüstet werden.

3.2.1.3.1 Ajax PHP kann als serverseitiger Teil von Ajax eingesetzt werden. Ajax kann den Datenverkehr zwischen Server und Client stark reduzieren und ist eine viel versprechende Technologie. Siehe auch: 3.2.2.2.

¹Siehe: 3.2.2.1

3.2.1.3.2 Anbindung MS Lync Es ist in PHP möglich, auf lokale DLLs zuzugreifen und diese ins Programm einzubinden, somit wäre eine Anbindung an MS Lync theoretisch möglich. Hier müssten jedoch weitere Abklärungen gemacht werden. Zudem müsste auch dies in einem proof of technology gezeigt werden.

3.2.2 Clientseitige Technologien

3.2.2.1 Java-Applets

Um Java Applets einsetzen zu können, müsste auf dem Client-System eine JRE installiert werden, da nur wenige Browser eine integrierte JVM verfügen und diese zudem veraltet sind und nur dem Funktionsumfang von JDK 1.1 entsprechen. Daher sollte davon abgesehen werden Applets einzusetzen, auch wenn davon auszugehen ist, dass auf den meisten Rechnern² bereits eine JRE installiert ist.

3.2.2.2 Java Script

Java Script ist eine weit verbreitete Technologie um dynamische Webseiten zu erstellen, es wird von allen bekannten Browsern direkt unterstützt. Die grosse Einschränkung ist hier jedoch das Sandbox Prinzip mit welchem Java Scripts ausgeführt werden. Der Zugriff auf den Client Rechner ist damit nicht möglich. Die Umsetzung einer Sprachübertragung oder eines Video-Chats ist damit nicht machbar, da keine Möglichkeit besteht auf das Mikrophon bzw. die Webcam zuzugreifen.

3.2.2.2.1 Ajax Asynchronous JavaScript and XML ermöglicht es den Datenverkehr zwischen Client und Server zu reduzieren und wird ebenfalls von allen Browsern unterstützt. Ajax ermöglicht es, dass nur die Daten welche sich wirklich ändern an den Server bzw. den Client geschickt werden. Es muss nicht bei jedem Request die ganze Seite neu aufgebaut werden.

3.2.2.3 Adobe Flash

Flash bietet sehr viel Möglichkeiten um mit dem User zu interagieren. In einem späteren Zeitpunkt wäre auch der Zugriff auf eine Webcam möglich. Jedoch können Flash Dateien nur mit dem proprietären Adobe Flash Player ausgeführt werden. Dieser Player ist allerdings weit verbreitet und laut Adobe auf 99% der Internet fähigen Desktop Rechnern installiert, gemäss anderen Statistiken³ jedoch nur auf 57%. Eine Einschränkung würde hier zudem iOS darstellen, da diese mobilen Geräte kein Flash unterstützen. Abhilfe schafft hier nur der Adobe Flash Media Server 4.5 welcher den Flash Stream für iOS Geräte in HTTP Content umwandelt und so die Darstellung ermöglicht.

²ca. 95%

³<http://www.webhits.de>

3.2.2.4 HTML5

HTML5 wird in Zukunft weit verbreitet sein. Alle bekannten Browser unterstützen bereits einen Grossteil von HTML5, obwohl der Standard noch nicht fertig gestellt wurde. Die Anzeige von Videos ist heute bereits möglich. Um alle Clients zu erreichen müsste das Video allerdings in zwei Formaten bereitgestellt werden. Dies ist nötig, weil der Internet Explorer und der Apple Safari nur mit dem MPEG4 (H.264) Codec arbeiten. Der Mozilla Firefox sowie Google Chrome jedoch mit Ogg Theora. Mit diesen beiden Codecs können aber alle Clients erreicht werden, inklusive weniger bekannten.⁴ In naher Zukunft ist auch der Zugriff auf Client Hardware, wie zum Beispiel die WebCam möglich, was auch einen Video-Chat ermöglichen würde.

3.3 Fazit

Der Vergleich der Technologien zeigt, dass eine Lösung mittels ASP.NET und HTML5, mit Javascript bzw. Ajax Elementen, die beste Lösung ist, da die Darstellung auf der Client Seite mittels Standard Webtechnologien die grösstmögliche Abdeckung bietet. Auf der Serverseite ist die Anbindung an den MS Lync Server mit dem von Microsoft bereitgestelltem SDK für MS Lync gewährleistet. Alle anderen Technologien haben entweder grosse Einschränkungen auf der Clientseite, da sie ein Plugin benötigen, oder die Anbindung an den MS Lync Server ist nicht, oder nur schwer möglich.

3.4 Quellen

Quelle	Titel	Autor	Erscheinungsdatum
1	Kompendium der Web-Programmierung dynamische Web-Sites	Thomas Walter	2008
2	http://www.wikipedia.de	-	10.2011
3	http://www.microsoft.com	-	10.2011
4	http://www.diveintohtml5.org/	Mark Pilgrim	10.2011

⁴Wie zum Beispiel Opera und Konqueror

4 Anforderungsspezifikation

4.1 Allgemeine Beschreibung

4.1.1 Produkt Perspektive

Diese Applikation soll Kunden die Möglichkeit bieten von einer Webseite aus via Consumer Integration mit dem Contact Center Kontakt aufzunehmen. Für den Agent ist es jederzeit ersichtlich wo genau auf der Seite sich der Kunde befindet. Er kann ihm so besser helfen und ihn durch die Seite führen.

4.1.2 Produkt Funktion

4.1.2.1 Clientseite

Der Kunde kann über die Consumer Integration mit dem Anbieter der Webapplikation in Kontakt treten und direkt seine Fragen stellen, ohne über eine Helpline anrufen zu müssen.

4.1.2.2 Serverseite

Der Agent erhält die Fragen des Kunden in der Lync Applikation. Es ist für ihn jederzeit ersichtlich, auf welcher URL sich der Kunde befindet. Er kann die Fragen des Kunden beantworten oder ihn durch einen Link auf eine andere Seite führen.

4.1.3 Benutzer Charakteristik

Die Applikation wendet sich an Personen, die im Umgang mit einer Webapplikation Hilfe benötigen. Ihnen soll der Umweg über eine Helpline erspart werden.

4.1.4 Einschränkung

Mit der aktuellen Technologie ist nur ein WebChat, ohne Audio oder Video Übertragung möglich. Die Möglichkeit für eine Erweiterung mit HTML5 soll aber für eine spätere Version offen gehalten werden. Die serverseitige Implementation soll in ASP.net gemacht werden. Clientseitig soll HTML mit CSS sowie Javascript und AJAX Elemente verwendet werden.

4.1.5 Abhängigkeiten

Die Applikation muss mit MS Lync interagieren können, da der Agent mit Lync arbeitet. Dies besteht bereits und soll so weiter verwendet werden.

4.2 Spezifische Anforderungen

4.2.1 Nichtfunktionale Anforderungen

4.2.1.1 Bedienbarkeit

Die Bedienung soll simpel gehalten werden, so dass sie für jedermann verständlich und intuitiv ist.

4.2.1.2 Stabilität

Die Übertragung der Nachrichten soll zu 99% funktionieren. Ausgenommen sind allfällige Netzwerkunterbrüche.

4.2.1.3 Design

Das Design soll sich am Luware Lean Design orientieren. Es soll zudem einfach mittels CSS an Kundenwünsche angepasst werden können.

4.2.1.4 Anpassbarkeit

Die Applikation muss ohne Probleme erweitert werden können. Das Grundgerüst für A/V Übertragung soll bereits integriert werden. Ein übertragenes A/V File soll im WebChat angezeigt, bzw. auf dem Server abgespeichert werden können.

4.2.1.5 Schnittstellen

Die Schnittstelle zu Lync bildet die von Luware entwickelte Consumer Integration Server Komponente die auf der MS UCMA API basiert

4.2.1.6 Verwendete Standards

HTML 5 darf für Voice und Video eingesetzt werden. Für alles andere sind Funktionen zu verwenden, die nicht HTML 5 verlangen.

4.3 Use Cases

4.3.1 Use Case Diagramm

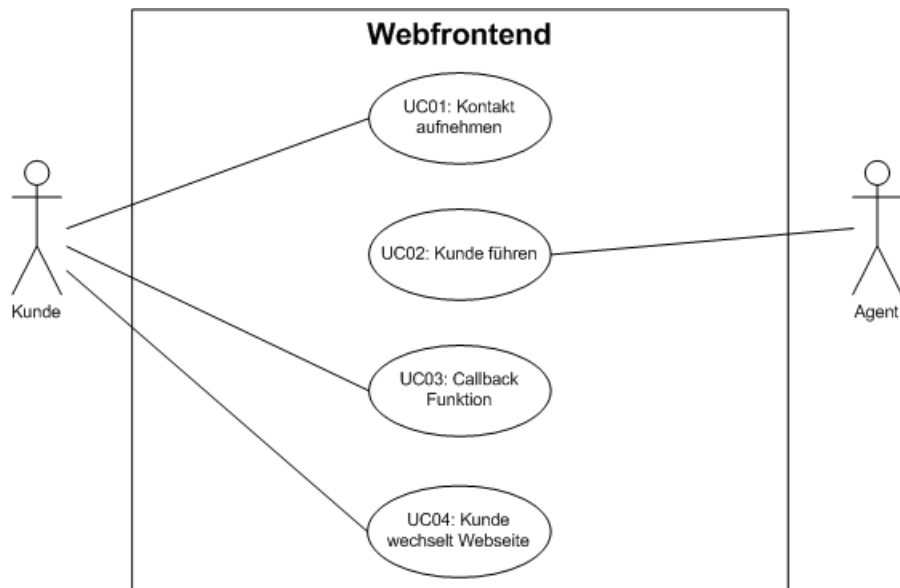


Abbildung 4.1: Use Case Diagramm

4.3.2 Aktoren & Stakeholders

Der primäre Akteur ist der Kunde. Er initiiert die Consumer Integration, indem er über das GUI auf der Seite seine Nachricht eingibt. Der zweite Akteur ist der Agent der Webapplikation, welcher das Gespräch entgegen nimmt und die Fragen des Kunden beantwortet und ihn durch die Webseite führt.

4.3.3 Use Cases Brief

4.3.3.1 UC01: Kontakt aufnehmen

Von der Consumer Integration wird nur der Kopfteil angezeigt. Der Kunde kann sie durch einen Klick öffnen und seine Frage bzw. Nachricht in die Chatbox eingeben. Zudem hat er die Möglichkeit einen optionalen DisplayName einzugeben. Der Agent erhält die Nachricht und falls gegeben den DisplayName. Zudem wird ihm sogleich die Webseite angezeigt auf welcher der Kunde sich befindet. Er kann dem Kunden direkt im Chat antworten.

4.3.3.2 UC02: Kunde führen

Der Agent stellt fest, dass sich der Kunde auf der falschen URL befindet. Er wechselt in seinem Fenster auf die korrekte URL. Dies wird dem Benutzer angezeigt und er kann

dem Agent folgen.

4.3.3.3 UC03: Callback Funktion

Wünscht der Kunde ein Telefongespräch, kann er mittels eines Buttons die Callback Funktion aufrufen. Dies öffnet eine Eingabemaske in welcher er seine Telefonnummer angibt. Das System ruft ihn anschliessend zurück. Die Telefonnummer muss im E164 Format eingegeben werden. Es soll auch eine Beispielnummer (+41751234567) angezeigt werden.

4.3.3.4 UC04: Kunde wechselt Webseite

Wechselt der Kunde die Webseite, zum Beispiel durch einen Klick auf einen Link, muss die Consumer Integration inkl. der Chat History bestehen bleiben. Der Kontakt zum Agent darf nicht verloren gehen. Die neue Seite des Kunden muss beim Agenten angezeigt werden.

4.3.4 UC01: Kontakt aufnehmen

Scope	SA Webfrontend
Level	Anwenderziel
Primary Actor	Kunde
Stakeholders & Interests	Benutzer: Einfache Kontaktaufnahme mit dem Contact Center
Preconditions	Der Kunde befindet sich auf der Webseite
Success Garentee	Nachricht wird an den Agent übermittelt
Main success scenario	<ol style="list-style-type: none"> 1. Kunde klickt auf das Consumer Integration Modul. 2. Das Modul öffnet sich, ein WebChat wird angezeigt 3. Kunde tippt seine Nachricht ein 4. Nachricht wird an den Lync Server übermittelt 5. Nachricht wird beim Agent im Lync angezeigt 6. Webseite des Kunden wird beim Agent angezeigt
Postconditions	Der Agent sieht die Nachricht sowie die Webseite in seiner Clientsoftware
Extensions	-
Special Requirements	-
Technology and data variations list	-
Frequence of occurrence	1-20 mal pro Ausführung

4.3.5 UC02: Kunde führen

Scope	SA Webfrontend
Level	Anwenderziel
Primary Actor	Agent
Stakeholders & Interests	Agent: Einfache Führung des Kunden durch die Webapplikation
Preconditions	Der Kunde befindet sich auf der Webseite und hat mit dem Contact Center Kontakt aufgenommen.
Success Garentee	Der Kunde erhält den Link des Agents und kann ihm folgen.
Main success scenario	<ol style="list-style-type: none"> 1. Agent wechselt die Seite 2. Nachricht wird an Kunden übermittelt 3. Kunde kann, nach manueller Bestätigung, dem Link folgen
Postconditions	Der Kunde befindet sich auf der vom Agent vorgeschlagenen Seite
Extensions	-
Special Requirements	-
Technology and data variations list	-
Frequence of occurrence	1-5 mal pro Ausführung

4.3.5.1 UC03: Callback Funktion

Scope	SA Webfrontend
Level	Anwenderziel
Primary Actor	Kunde
Stakeholders & Interests	Benutzer: Telefonischer Rückruf auslösen
Preconditions	Der Kunde befindet sich auf der Webseite
Success Garentee	Der Kunde wird vom System angerufen
Main success scenario	<ol style="list-style-type: none"> 1. Kunde klickt auf den Callback button 2. Kunde tippt seine Telefonnummer ein 3. Nachricht wird an Lync übermittelt 4. Lync ruft den Kunden sowie den Agenten an 5. Der Wählvorgang wird dem Kunden angezeigt
Postconditions	Das Telefongespräch wurde aufgebaut
Extensions	-
Special Requirements	-
Technology and data variations list	-
Frequence of occurrence	1 mal pro Ausführung

4.3.6 UC04: Kunde wechselt Webseite

Scope	SA Webfrontend
Level	Anwenderziel
Primary Actor	Kunde
Stakeholders & Interests	Kunde: Webseite wechseln
Preconditions	Der Kunde befindet sich auf der Webseite
Success Garentee	Der Kunde befindet sich auf neuer Webseite ohne den Kontakt zum Agent zu verlieren
Main success scenario	<ol style="list-style-type: none"> 1. Kunde chattet mit einem Agent 2. Kunde klickt auf einen Link 3. Neue Seite wird angezeigt 4. Chat bleibt inklusive History bestehen 5. Kunde kann mit Agent weiter chatten 6. Agent sieht die neue Webseite des Kunden
Postconditions	Kunde befindet sich auf neuer Webseite
Extensions	-
Special Requirements	-
Technology and data variations list	-
Frequence of occurrence	5-10 mal pro Ausführung

5 Management Summary

5.1 Ausgangslage

Die Firma Luware AG aus Zürich hat bereits eine Silverlight Applikation entwickelt, welche es einem User ermöglicht, auf eine Webseite direkt mit einem Supporter (Folgend Agent) Kontakt aufzunehmen. Der Agent arbeitet mit MS Lync und für ihn ist darin jederzeit ersichtlich auf welcher Webseite sich der User befindet. Leider hat sich Silverlight jedoch nicht durchgesetzt. Deshalb wird diese Applikation nun im Rahmen der Studienarbeit neu entwickelt, jedoch soll dabei auf standard Internet-Technologien gesetzt werden. Damit wird auch eine möglichst breite Schicht von Anwendern abgedeckt, da keine Plugins oder proprietäre Software verwendet wird.

5.1.1 Vorgehen, Technologien

Um ausfindig zu machen, mit welchen Technologien am meisten User bedient werden können, wurden in einer ersten Phase der Studienarbeit die in Frage kommenden Technologien analysiert und verglichen. Diese Analyse ist im Dokument [02-Requirements/TechnologieAnalyse.pdf] beschrieben. Dabei wurde auf die Verbreitung, die Plattformunabhängigkeit, die Möglichkeit von Datenübertragung (z.B. Audio / Video) Wert gelegt. Die Analyse kam zum Schluss, dass die geeignetste Technologie ASP.Net ist, da damit die Anbindung an MS Lync problemlos funktioniert. Die Darstellung auf dem Client Rechner soll mit standard HTML / CSS gemacht werden.

Als erstes wurde ein Webchat Prototyp mittels ASP.Net erstellt. Dieser ermöglichte das Chatten zwischen mehreren Usern, jedoch ohne eine Anbindung an MS Lync. Anschliessend wurde dieser Prototyp ausgebaut und an MS Lync angebunden. Nachdem die Übermittlung von Nachrichten an Lync und zurück möglich war, wurden die weiterführenden Funktionen, wie die Anzeige der aktuellen Webseite sowie die Callback-Funktion implementiert.

5.1.2 Ergebnisse

Die Applikation konnte gemäss der Requirements Specification implementiert werden. Sie bietet dem User die Möglichkeit, via Webchat Kontakt zum Agent aufzunehmen und von ihm geführt zu werden. Der Webchat wurde als eigenständiges WebControl entwickelt und kann somit sehr einfach in bestehende Webseiten eingebunden werden. Zudem wurden Vorarbeiten geleistet, um eine Einbindung von Videoübertragung zu erleichtern. Hierfür wurde auf HTML5 gesetzt, was von den verbreitetsten Browsern bereits teilweise unterstützt wird. Jedoch birgt der Standard bzw. die Implementation in den einzelnen

Browsern noch einige Tücken, weshalb die Übertragung eines Videostreams nicht vollständig gelang.

5.1.3 Ausblick

In einer späteren Version könnten dem User noch mehr Möglichkeiten der Kommunikation geboten werden. Vorstellbar wäre eine Art Desktop-Sharing oder Video-Telefonie. Hierfür sollen die Möglichkeiten von HTML5 genutzt werden. Der Standard ist aber noch nicht fertiggestellt und die Unterstützung der Browser noch sehr unterschiedlich.

6 Software Architektur

6.1 Architektonische Darstellung

Die Applikation soll sehr simpel aufgebaut werden. Sie besteht grundsätzlich aus zwei Teilen, dem Web UI sowie der Anbindung via WCF an die bestehende Lync Schnittstelle von Luware.

6.2 Architektonische Ziele & Einschränkungen

6.2.1 Ziele

- Die UI Komponenten sollen klar vom Business Layer getrennt werden.
- Die Applikation soll möglichst einfach, (z.B. via eines Tags) in eine bestehende Webseite eingebunden werden können.
- Die Benutzeroberfläche muss intuitiv bedienbar sein. Da sie sich in erster Linie an Personen richtet, die beim Bedienen einer Webapplikation Hilfe benötigen.

6.3 Physische Sicht

Das hier entwickelte System befindet sich auf dem Webserver des Kunden von Luware. Das Webfrontend kommuniziert via der Luware Consumer Interaction mit einem MS Lync Server des Kunden. Zudem wird für die Anzeige der aktuellen Kunden Webseite eine Silverlight Applikation verwendet. Diese ist ebenfalls als externer Service zu betrachten. Folgende Abbildung (6.1) zeigt den Aufbau schematisch.

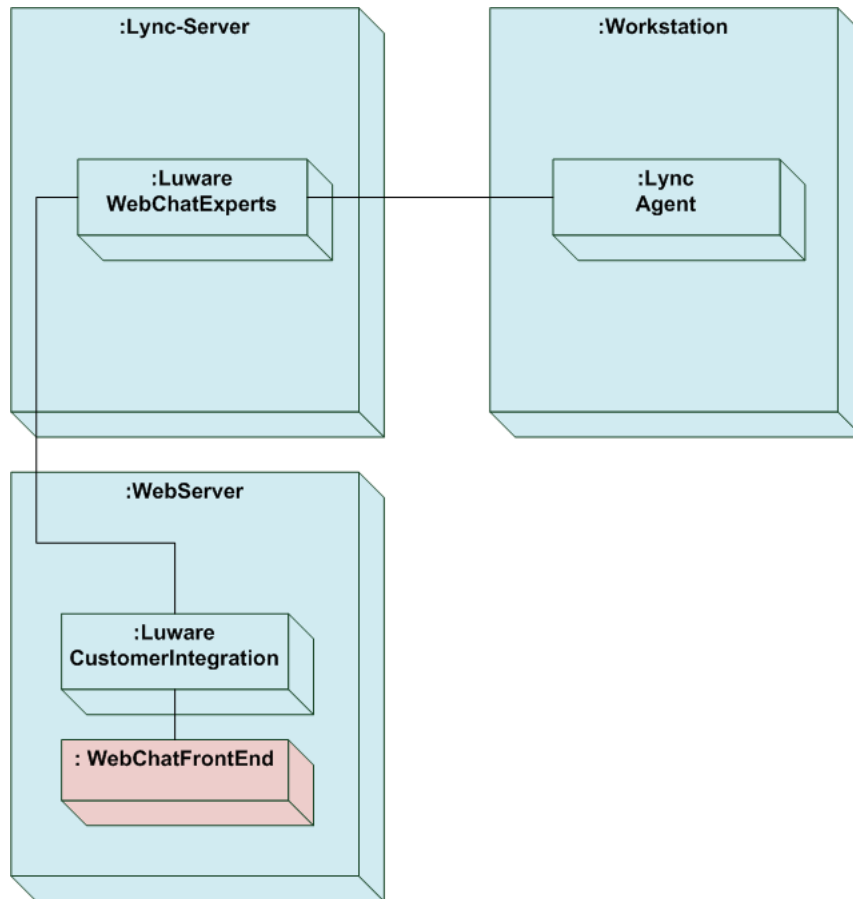


Abbildung 6.1: Physische Sicht

6.4 Logische Sicht

Die gesamte Applikation befindet sich im Namespace `WebChat.Client.Html`. Eine Ausnahme bilden die Interfaces welche für die Kommunikation mit MS Lync verwendet werden. Sie befinden sich im Package `WebChat.Service`. Dies weil die Interfaces von allgemeiner Bedeutung sind und nicht nur in dieser Applikation verwendet werden. Zudem wurden die Interfaces nicht im Rahmen dieser Arbeit erstellt. Die Applikation ist folgend (Abbildung 6.2) schematisch dargestellt und wird so unterteilt.

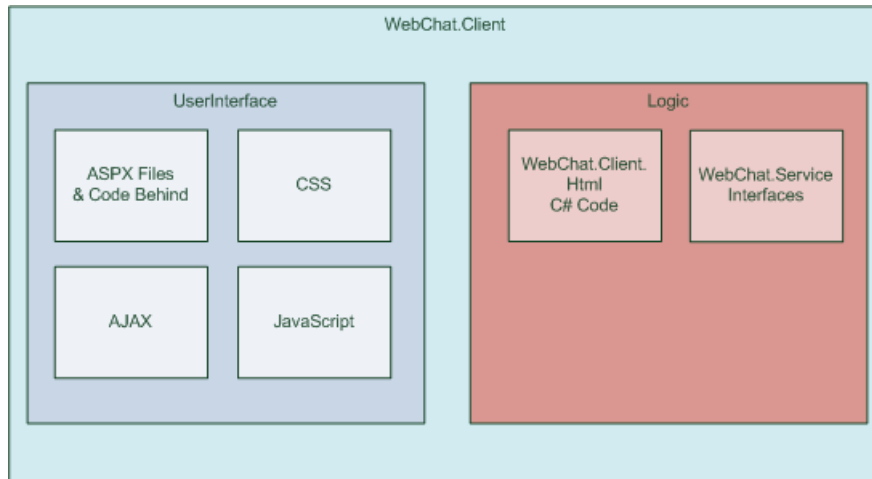


Abbildung 6.2: Logische Sicht

Die Applikation besteht aus den User-Interface Komponenten, welche sich in den HTML, CSS sowie APSX Dateien und den dazugehörigen Code-Behind Elementen befinden. Die eigentliche Business-Logik der Anwendung befindet sich im Namespace `WebChat.Client.Html`. Dazu kommen Interfaces welche zur Kommunikation via WCF mit der bestehenden Consumer Integration Schnittstelle von Luware verwendet werden. Diese befinden sich im Namespace `WebChat.Service.Interfaces`.

6.5 Designentscheidungen

6.5.1 Datenhaltung

Das Hauptziel dieser Applikation besteht darin, Kunden auf einfache Art und Weise Hilfe mit einer Webapplikation bieten zu können. Eine History wird hierfür nicht benötigt. Zudem besteht die Möglichkeit die History Funktion von MS Lync zu nützen. Der Einsatz einer Datenbank um den Chatverlauf aufzuzeichnen ist somit fragwürdig und der damit Verbundene Overhead kann nicht gerechtfertigt werden.

6.5.2 Design Packages

6.5.2.1 Package Client.Html

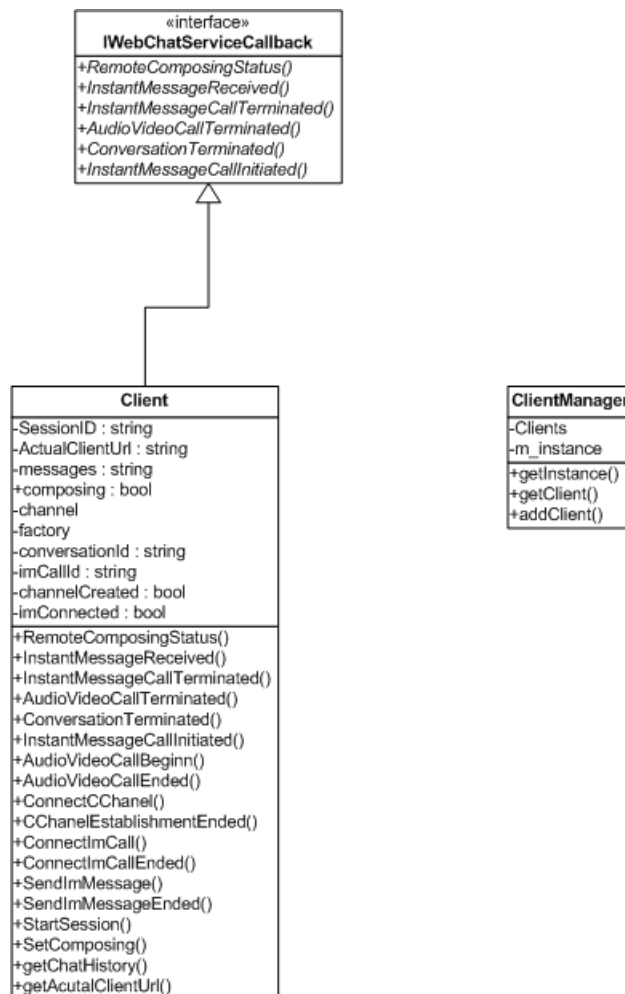


Abbildung 6.3: Klassendiagramm Client.Html

6.5.2.1.1 Klassen Diagramm

6.5.2.1.2 Beschreibung des Packages In diesem Package wird die gesamte Logik der Applikation behandelt.

6.5.2.1.3 Sequenzdiagramm Start Session Folgendes Sequenzdiagramm (Abbildung 6.4) zeigt, wie die Session und damit die Verbindung zum MS Lync Server aufgebaut wird. Dies wird jeweils vor dem Senden der ersten Nachricht, welche der Kunde im WebChat eingibt, ausgeführt. Dabei wird die Verbindung nur bis zum WCF Channel gezeigt, da

ab dem Channel das Framework die Verbindung übernimmt. Zudem wurde zugunsten einer besseren Übersicht darauf verzichtet, die Erstellung einzelner DTOs, wie sie die Methoden des Channels erwarten, darzustellen.

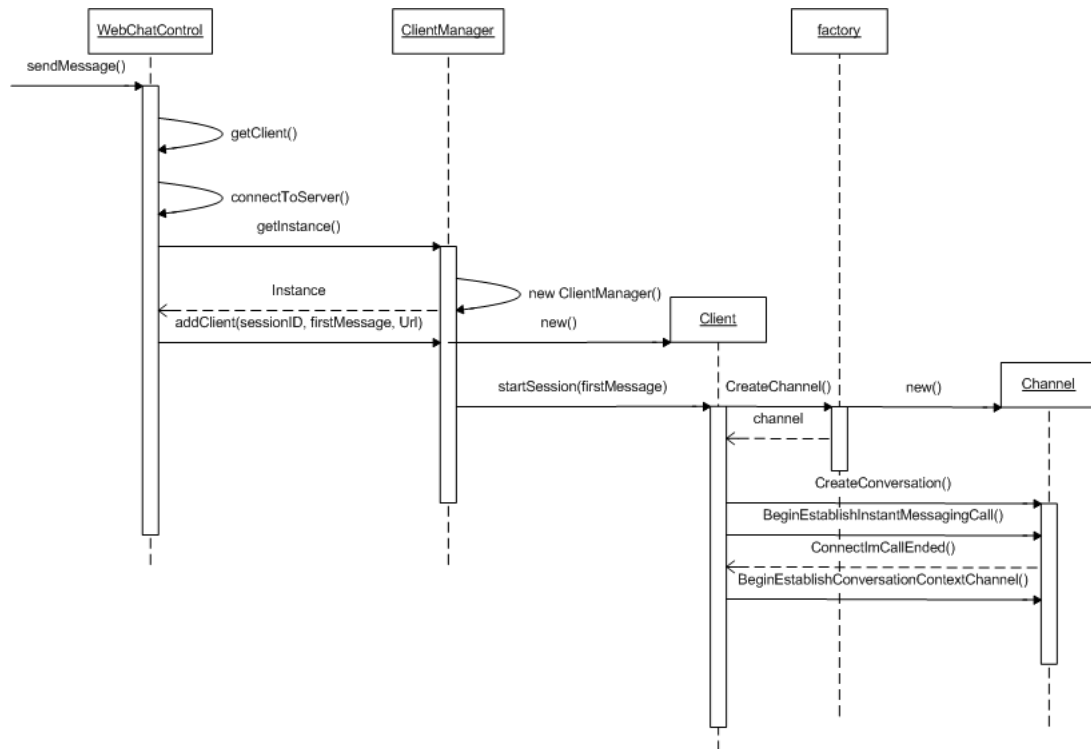


Abbildung 6.4: Sequenzdiagramm Start Session

6.5.2.1.4 Sequenzdiagramm Send Message In diesem Diagramm (Abbildung 6.5) wird gezeigt, wie das Versenden einer Nachricht vom Kunden zum Agenten abläuft. Dies wird bei jeder versendeten Nachricht durchlaufen. Auch hier wird der Vorgang nur bis zum WCF Channel dargestellt. Ebenfalls wurde bei den Aufrufen der Channel Methoden auf die Darstellung von DTOs und Parameterlisten verzichtet, um eine bestmögliche Übersicht zu gewährleisten.

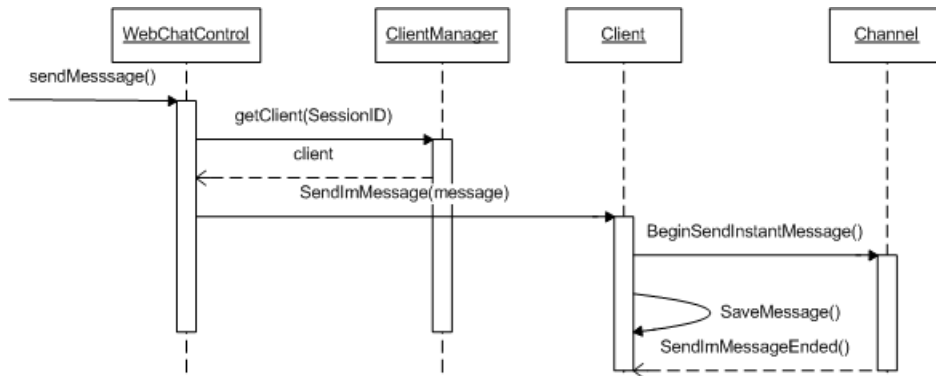


Abbildung 6.5: Sequenzdiagramm Send Message

6.5.2.1.5 Sequenzdiagramm Receive Message Folgend (Abbildung 6.6) wird gezeigt, wie das Empfangen von Nachrichten abläuft. Der Agent gibt seine Nachricht im MS Lync ein, der Lync Server ruft dann via dem Callback Channel den Client auf und übergibt die Nachricht. Dargestellt wird der Vorgang wiederum nur ab dem WCF Channel, welcher beim jeweiligen Client die Methode `InstantMessageReceived()` aufruft. Eine Besonderheit dieses Ablaufes ist der Update des WebChatControl. Hierfür wird ein PartialUpdate mittels eines Ajax Timers benützt. Der Timer ruft periodisch die `updateMessages()` Methode des WebChatControls auf. Das Control wiederum holt sich die aktuelle ChatHistory vom Client und stellt diese dar.

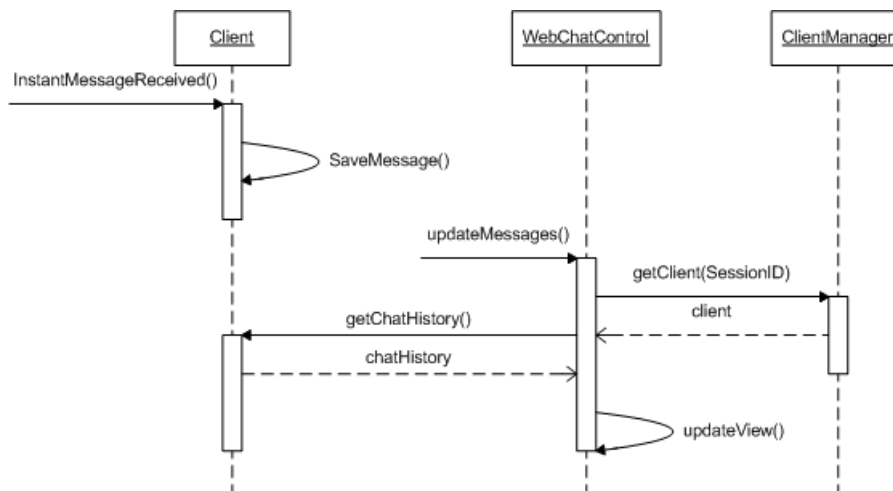


Abbildung 6.6: Sequenzdiagramm Receive Message

6.5.2.1.6 Kernoperationen

6.5.2.1.6.1 Client.StartSession(String firstMessage):void Dies ist eine Zentrale Methode der Applikation. Sie baut die Verbindung zu MS Lync auf. Dazu wird mit einer

DuplexChannelFactory ein WCF Channel konstruiert. Anschliessend wird auf diesem eine Conversation gestartet. Hierbei wird das Subjekt der Conversation, der Name des Absenders, sowie seine SIP URI festgelegt. Gelingt der Aufbau der Conversation wird anschliessend ein InstantMessageCall aufgebaut. Dieser Call benötigt die ConversationId der zuvor aufgebauten Conversation, zudem die SIP URI des Empfängers und die erste Nachricht. Gelingt auch dies, wird durch den Aufruf von ConnectImCallEnded() durch den WCF Channel der Aufbau des zweiten Channels ausgelöst. Dieser zweite Channel, der ContextChannel, wird für die Übertragung der aktuellen Webseite des Kunden genutzt. Dem ContextChannel werden beim Start die ApplicationId der WebChatExperts Applikation, die ConversationId, sowie die URL der aktuellen Webseite des Kunden übergeben. Damit ist die Verbindung zwischen dem WebChatControl und MS Lync vollständig aufgebaut und Nachrichten können übertragen werden.

6.5.2.1.6.2 Client.SendImMessage(String message):void Diese Methode ist vergleichsweise simpel, aber nicht weniger wichtig. Sie behandelt das Versenden von Nachrichten via dem zuvor aufgebauten WCF Channel. Hierfür wird dem Channel ein SendInstantMessageRequest übergeben. Dieser Request enthält die ConversationId, die Nachricht, sowie eine eindeutige RequestId. Anschliessend wird die Nachricht in der ChatHistory des Clients gespeichert.

6.5.2.1.6.3 Client.

InstantMessageReceived

(InstantMessageReceivedNotification):void Dies ist die Callback Methode für den WCF Channel. Sie wird vom Channel aufgerufen, wenn dieser von MS Lync eine Nachricht für den Client erhalten hat. Dem Client wird eine InstantMessageReceivedNotification übergeben. Diese Notification enthält die ImCallId, die erhaltene Nachricht, den Absender sowie eine MessageID. Die Nachricht wird mit dem Sender in die ChatHistory des Clients gespeichert.

7 GUI Analyse

7.1 Allgemeine Beschreibung

Die User Interface soll, gemäss dem Industriepartner Luware, lean gehalten werden. Lean bedeutet soviel wie dünn oder knapp. Deshalb wurde Wert darauf gelegt, dass GUI möglichst simpel zu halten. Es soll für jeden verständlich und intuitiv bedienbar sein und ohne unnötige Effekte auskommen. Die Chatfunktion von Google wurde dabei, auf Wunsch von Luware, als Inspiration verwendet. Dabei sind folgende Paperprototypes entstanden.

7.2 Paperprototypes

7.2.1 Einstieg

Da es sich bei der dieser Applikation um eine Hilfe Leistung handelt, welche nicht von allen Kunden in Anspruch genommen werden wird. Soll die Möglichkeit bestehen, sie platzsparend darstellen zu können. Deshalb soll es möglich sein, nur die Titelleiste des Webchats anzuzeigen. Diese kann durch einen Klick vergrössert werden.



Abbildung 7.1: Webchat minimiert

7.2.2 WebchatControl

Wird von einem Kunden die Hilfe in Anspruch genommen, wird der ganze Webchat angezeigt. Das Design soll dabei jedoch einfach gehalten werden. Das WebchatControl besteht nur aus einer Textbox für die Eingabe der Nachrichten sowie eine für die Darstellung der History. Dem Kunden soll die Möglichkeit geboten werden, das WebchatControl selbst zu platzieren. Er kann es per Drag & Drop verschieben, sowie durch einen Klick auf die Titelleiste wieder minimieren wenn er es nicht mehr benötigt. In der Titelleiste sollen die Buttons für zusätzliche Funktionen, wie das Anfordern eines telefonischen Gesprächs und der Auslagerung des Chats in ein eigenes Fenster, untergebracht werden.

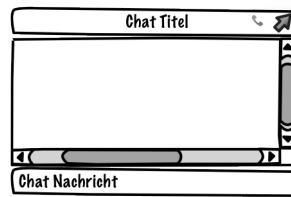


Abbildung 7.2: Webchat Control

7.2.3 Eingabemaske für Telefonnummer

Durch einen Klick auf das Telefon in der Titelleiste, öffnet sich eine Eingabemaske in welcher die eigene Telefonnummer eingegeben werden kann. Die Eingabemaske soll über dem WebchatControl zu liegen kommen, um keine Inhalte der restlichen Webseite zu verdecken. Auch hierbei soll wiederum auf ein simples Design geachtet werden. Die Eingabemaske besteht deshalb nur aus dem nötigsten.



Abbildung 7.3: Eingabemaske für Telefonnummer

7.2.4 Chat in eigenem Fenster

Falls der Kunde den Chat in einem eigenen Fenster haben möchte, kann er dies durch einen Klick auf den Pfeil in der Titelleiste erreichen. Um Verwirrungen zu vermeiden, soll sich das WebchatControl innerhalb der Webseite minimieren sobald der Chat ausgelagert wird. Ebenso soll sich das Control wieder vergrössern, sobald der Kunde das Fenster schliesst. Klickt der Kunde auf die minimierte Titelleiste des Controls, während dem der Chat ausgelagert ist, soll das Fenster des Chats den Fokus erhalten, ohne dass sich das Control innerhalb der Webseite vergrössert.

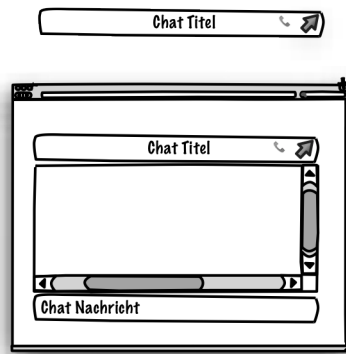


Abbildung 7.4: Ausgelagerter Chat

8 Technischer Bericht

8.1 Übersicht & Aufgabenstellung

Das Ziel dieser Studienarbeit war es, ein webbasiertes User Interface für das bereits bestehende Contact Center der Firma Luware AG zu entwickeln. Die Applikation soll Kunden die Möglichkeit bieten, direkt auf einer Webseite Hilfe anzufordern. Der Kunde soll seine Frage via einem Webchat direkt einem Agent stellen können. Für den Agent muss ersichtlich sein, auf welcher URL sich der Kunde befindet. Er soll ihn zudem durch ein "co-browsing" durch eine Webseite führen können. Der Agent arbeitet mit MS Lync, für welches Luware bereits ein Plugin entwickelt hat, welches dem Agent die Webseite des Kunden anzeigt. Ebenfalls wurde ein Interface zum MS Lync Server bzw. zum UCMA API, über welches die Applikation an MS Lync angebunden werden kann, von Luware entwickelt.

Die Applikation soll mit standard Webtechnologien entwickelt werden, um eine möglichst grosse Anzahl von Kunden erreichen zu können. Aus dem selben Grund soll auch die Verwendung von Plugins vermieden werden.

8.2 Ergebnisse

Der Webchat konnte vollständig implementiert werden. Er bietet dem Kunden die Möglichkeit mit einem Agent zu chatten und seine Fragen zu stellen. Der Webchat kommuniziert über die Customer Integration Schnittstelle von Luware mit MS Lync, mit welchem der Agent arbeitet. Diese Anbindung funktioniert gut, der Agent erhält die Nachricht des Kunden sowie die URL auf welcher der Kunde sich aktuell befindet. Der Kunde kann zudem über eine Eingabemaske seine Telefonnummer angeben und so ein telefonisches Gespräch mit einem Agent anfordern. Auch dies konnte korrekt implementiert werden.

Beim User Interface wurde darauf geachtet, das es intuitiv und schlank gehalten wurde. Es kann dank der Auslagerung der Darstellung in eigene CSS Files jedoch einfach an Kundenwünsche angepasst werden. Es wurde mittels aktuellen Technologien aufgebaut und bietet die Möglichkeit, per Drag & Drop an jede gewünschte Stelle innerhalb der Webseite verschoben zu werden. Zudem kann es auch in einem eigenen Fenster geöffnet werden. Neue Nachrichten des Agent werden per AJAX geladen, somit wird nur das WebChatControl neu geladen und ein Refresh der ganzen Webseite vermieden.

Leider konnten einzelne Funktionen nicht vollständig getestet werden, da die WebChat Experts Applikation von Luware in der Testumgebung nicht richtig funktionierte oder die Funktion noch nicht vollständig implementiert ist. So konnte das Auslösen eines Telefongesprächs nicht getestet werden, der Request wird in der jetzigen Version korrekt an die

Customer Integration Schnittstelle übergeben. Das Auslösen des eigentlichen Anrufes ist aber innerhalb der Customer Integration Applikation von Luware noch nicht implementiert. Die Anzeige der aktuellen Webseite des Kunden konnte ebenfalls nicht vollständig getestet werden. Die URL wird an MS Lync übergeben und dort als Link dargestellt. Dies ist gemäss Herrn Beck von der Luware soweit auch korrekt. Die Darstellung der Webseite innerhalb von MS Lync funktionierte allerdings in der Testumgebung nicht, da die WebChat WebSite Applikation von Luware beim Aufruf abstürzt.

8.3 Schlussfolgerungen

Die meisten Ziele der Arbeit konnten korrekt implementiert werden. Der Webchat ist auf einem guten Stand und das User Interface ist gelungen und bietet dem User wie auch dem Administrator der Webseit die Möglichkeiten es den eigenen Wünschen anzupassen. Der Administrator kann es durch die CSS Datei einfach an das Design der entsprechenden Webseite anpassen und der User kann es während der Laufzeit minimieren, verschieben oder in ein eigenes Fenster auslagern.

Es ist allerdings schade, dass einige Funktionen nie richtig getestet werden konnten und die Probleme in der Testumgebung leider nicht behoben werden konnten. Rückblickend hätten die Probleme durch eine intensivere Zusammenarbeit mit Luware früher erkannt und eventuell behoben werden können. Ein Möglichkeit wäre auch gewesen, einzelne Tage in den Räumlichkeiten von Luware zu arbeiten und so das Zusammenspiel zu verbessern.

9 Persönlicher Bericht

9.1 Allgemeiner Rückblick

Ich habe bei dieser Arbeit sehr viel gelernt. Was einerseits darauf zurück zu führen ist, dass viele der verwendeten Technologien für mich neu waren, es war mein erstes ASP.net Projekt, was für mich sehr interessant war, jedoch auch einige Kopfschmerzen bereitete. Andererseits war es auch das erstmal, dass ich bei einem grösseren Projekt auf mich allein gestellt war. Ich habe dadurch den Austausch im Team, welcher bis jetzt immer möglich warm, umso mehr schätzen gelernt. Es gibt immer wieder Situationen in welchen man nicht weiter weiss, da ist eine zweite Sichtweise meist sehr hilfreich. Die Zusammenarbeit mit Luware wie auch mit Prof. Stettler war meines Erachtens immer professionell und offen. Leider gab es allerdings manchmal längere Antwortzeiten, aufgrund von meiner Abwesenheit im Militärdienst sowie Herrn Becks starker Belastung durch seine Firma wie auch die Schulungen welche er zudem noch leitet.

9.2 Gelernte Lektionen

Da mir das Programmieren und Tüfteln mit den neuen Technologien sehr viel Spass bereitete, ist leider die Dokumentation der Arbeit etwas ins Hintertreffen geraten, weshalb die Doku auch zu relativ weit vortgeschrittener Projektzeit noch sehr mager war. Ich werde in Zukunft, speziell auch bei der Bachelorarbeit, darauf achten, dass die Dokumentation von Anfang an geführt wird. Zudem muss ich mehr Zeit in die genaue Planung einer Arbeit investieren, und mich dann auch bemühen, mich an diese zu halten. Ich habe zu Beginn der Arbeit eine Planung erstellt, sie dann jedoch beim Tüfteln an den programmtechnischen Problemen etwas aus den Augen verloren. Allerdings konnte ich rückblickend feststellen, dass meine Planung meistens mit der effektiv geleisteten Arbeit übereinstimmte. Es gilt jedoch in Zukunft dies, wie auch die Abweichungen, besser zu dokumentieren.

10 Test Dokumentation

10.1 Voraussetzungen

Damit ein Systemtest korrekt durchgeführt werden kann, muss die Applikation auf einem IIS Web Server installiert werden. Der Webserver muss eine Verbindung zur Customer Integration Konsolen-Applikation von Luware aufbauen können. Zudem muss die Applikation so konfiguriert werden, dass die Verbindung zur Customer Integration korrekt aufbaut. Der Lync Account des Agent muss bei der Customer Integration als Agent bzw. WebChat Expert registriert sein.

10.2 Vorbereitungen

Die Customer Integration Applikation muss gestartet und eine Verbindung zum MS Lync Server muss aufbauen werden können. Auf der Workstation des Agent muss der MS Lync Client gestartet sein. Zudem muss der Agent selbst mit dem korrekten User eingeloggt sein.

10.3 Systemtests

10.3.1 Test vom 16.12.2011

Testbeschreibung	Erwartetes Resultat	Resultat	Bemerkung
Nachricht von Webchat an Agent	Agent erhält Nachricht	OK	
Antwort von Agent an Webchat	Kunde erhält Nachricht	OK	
Anzeige der Webseite des Kunden beim Agenten	Webseite wird beim Agent angezeigt	Nicht OK	Context Channel wird korrekt aufgebaut und Webseite übergeben. Die WebChat Experts Applikation von Luware hat noch Probleme
Kunden führen	Link wird beim Kunden angezeigt	Nicht OK	Die WebChat Experts Applikation von Luware hat noch Probleme
Kunde wechselt Webseite	Webseite wird beim Agent angezeigt	Nicht OK	Wird beim Agenten nicht angezeigt Im WebchatControl aber implementiert
Callback Funktion	Anruf wird ausgeführt	Nicht OK	Aufruf erfolgt korrekt. Die eigentliche Anruf Funktion wurde aber seitens Luware noch nicht implementiert

10.3.2 Test der GUI Funktionalität 22.12.2011

Testbeschreibung	Erwartetes Resultat	Resultat	Bemerkung
Klick auf minimierte Titelleiste	WebchatControl wird vollständig angezeigt	OK	
Klick auf Titelleiste von maximiertem Control	Control minimiert	OK	
Control durch Drag & Drop verschieben	Control kann verschoben werden	OK	
Klick auf Telefon in Titelleiste	Eingabemaske für Telefonnummer wird angezeigt	OK	
Klick auf Pfeil in Titelleiste	Control wird in eigenes Fenster ausgelagert	OK	
	Control in Webseite wird minimiert	OK	
Klick auf minimierte Titelleiste bei ausgelagertem Chat	Fenster vom Chat erhält den Fokus	OK	
Fenster von ausgelagertem Chat schliessen	Fenster schliesst sich, Control in Webseite wird vollständig angezeigt	OK	

11 Benutzeranleitung

11.1 Wie kann ich meine Frage stellen?

Sie finden auf der Webseite folgende Abbildung.



Abbildung 11.1: WebChat minimiert

Durch einen Klick auf die Titelleiste wird die Chat Applikation geladen. Anschliessend können Sie in der unteren Textbox einfach ihre Frage eintippen und mittels der Enter Taste abschicken. Sie sehen ihre Nachricht sobald diese versendet wurde in der oberen Textbox, in welcher Sie auch die Antwort des Agent sehen werden. Dies wird in der nächsten Abbildung gezeigt.

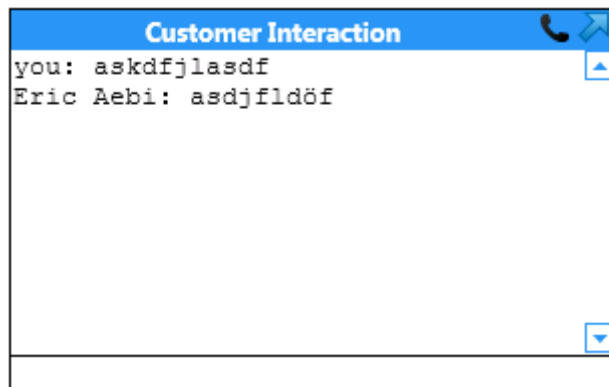


Abbildung 11.2: WebChat mit Antwort vom Agenten

Der WebChat kann mittels eines Klicks auf die Titelleiste auch wieder minimiert werden (siehe Abbildung 11.1). Auf die gleiche Weise kann der Chat anschliessend auch wieder vergrössert werden. Durch ein Klicken und halten der Titelleiste kann der WebChat auf eine beliebige Position innerhalb des Browserfensters gezogen werden.

11.2 Chat in eigenem Fenster öffnen

Durch einen Klick auf den Pfeil in der rechten oberen Ecke des WebChats, kann dieser in ein eigenes Fenster ausgelagert werden. Dabei wird der Chat im Hauptfenster minimiert. Sie können ohne Unterbruch weiter chatten. Sobald Sie das Fenster des Chats wieder schliessen, öffnet sich der Chat im Hauptfenster wieder. In folgenden Bild (11.3) wird der bereits ausgelagerte Chat dargestellt.

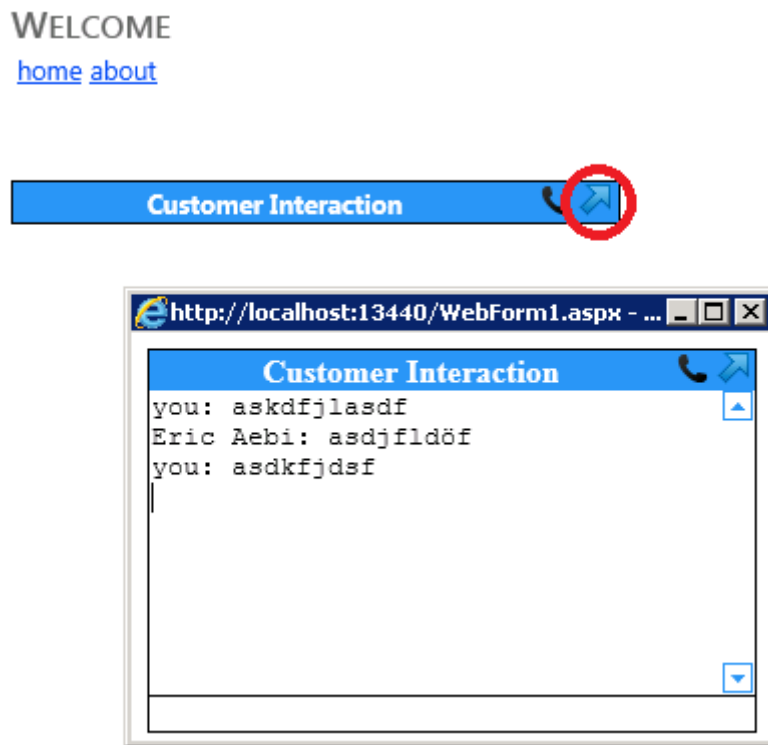
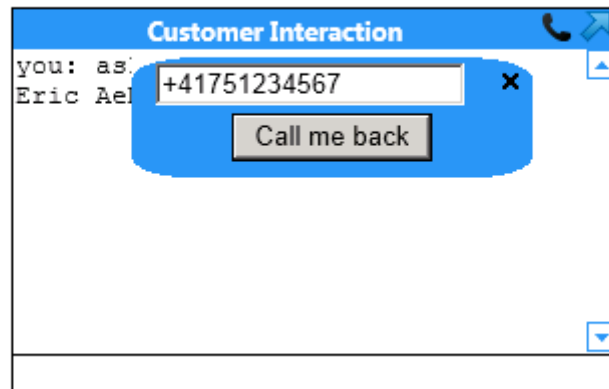


Abbildung 11.3: WebChat in eigenem Fenster

11.3 Telefonanruf auslösen

Durch einen Klick auf das Telefon in der rechten oberen Ecke des WebChats kann dem Contact Center mitgeteilt werden, dass Sie ein telefonisches Gespräch mit einem Agent wünschen. Hierfür öffnet sich ein Eingabefenster in welchem Sie Ihre Telefonnummer eintragen müssen. Bitte geben Sie die Nummer mit Vorwahl ein (z.B. +447905544). Lösen Sie den Anruf anschliessend durch einen Klick auf den "Call me back" Button aus. Der Wählvorgang wird Ihnen sogleich angezeigt. Wenige Minuten später wird sich ein Agent bei Ihnen melden. Folgende Abbildung (11.4) zeigt das Eingabefenster für die Telefonnummer. Das Fenster kann durch einen Klick auf das x wieder geschlossen werden.



The image shows a screenshot of a web application window titled "Customer Interaction". The window has a blue header bar with a telephone handset icon and a close button (X) on the right. Below the header, there is a text area containing the text "you: as:" and "Eric Aebi". A text input field is positioned over the text area, containing the phone number "+41751234567". Below the input field is a button labeled "Call me back". The window also features a scroll bar on the right side.

Abbildung 11.4: Eingabefenster für Telefonnummer

12 Installationsanleitung

12.1 Zweck

Dieses Kapitel erläutert die Installation des WebChat Controls auf einem WebServer. Dabei wird vorausgesetzt, dass der Internet Information Services (IIS) korrekt auf dem Rechner installiert ist.

Damit die ganze WebChat Applikation korrekt funktioniert, müssen noch weitere Applikationen installiert und gestartet werden. Wie zum Beispiel die Consumer Integration von der Luware AG, sowie ein MS Lync Server. Diese Applikationen müssen zudem auch korrekt konfiguriert werden. Dies ist jedoch nicht Teil dieser Installationsanleitung.

12.2 Installation der Webapplikation

In diesem Kapitel wird beschrieben, wie die gesamte Webapplikation auf einem IIS Webserver installiert wird. Der WebChat läuft anschliessend als eigenständige Applikation auf dem IIS. Hierfür müssen folgende Schritte unternommen werden.

12.2.1 Erstellung einer neuen Applikation

Öffnen Sie den IIS Manger (unter "Start" -> "All Programs" -> "Administrative Tools"). Klicken Sie mit der rechten Maustaste auf die Site innerhalb welcher Sie die Webapplikation installieren möchten. Wählen Sie anschliessend "Add Application..." im Kontextmenü, wie in folgender Abbildung (12.1) dargestellt.

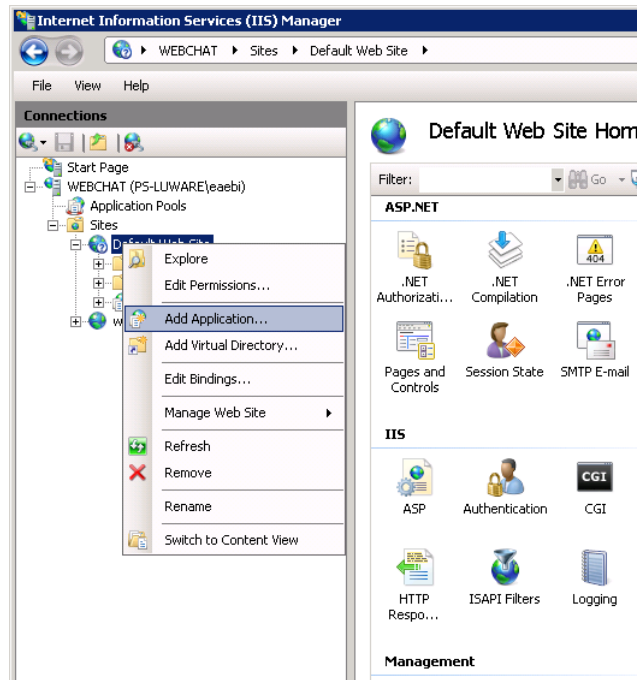


Abbildung 12.1: IIS Manager

Geben Sie im Eingabefenster den gewünschten Alias ein unter welchem Sie die Applikation später erreichen möchten. Wählen Sie zudem den Pfad, unter welchem Sie die Source Dateien der Applikation installieren möchten. Bestätigen Sie Ihre Eingaben mittels OK.

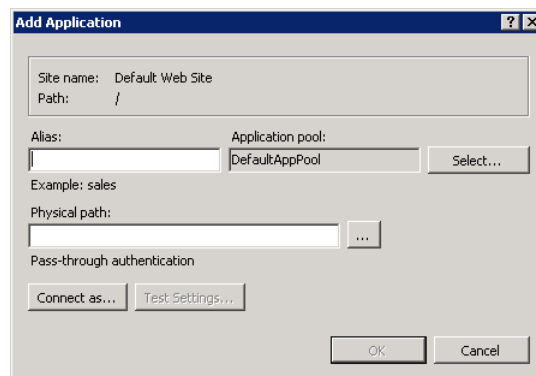


Abbildung 12.2: Eingabefenster Add Application

12.2.2 Deploy der Source Dateien

Ist die Applikation im IIS erstellt, müssen noch die Source Dateien deployed werden. Hierfür muss das ZIP File unter dem in der IIS Applikation eingegebenen Pfad extrahiert werden (siehe Abbildung 12.2). Das ZIP File mit den vollständigen Sourcen finden Sie unter 13-Deliverables/complete.zip.

12.3 Installation nur des WebChatControls

Soll nur das WebChatControl in eine bestehende Webapplikation eingebunden werden, kann das ZIP File 13-Deliverables/WebChatControl.zip verwendet werden. Es enthält nur das WebChatControl, ohne jegliche Webseite, welche das Control verwendet. Es stellt also keine funktionsfähige Webapplikation dar. Das ZIP File kann in einen beliebigen Ordner einer bestehenden Webapplikation extrahiert werden.

Anschliessend muss das Control in jede Webseite, welche es verwenden soll, eingebunden werden. Dies kann wie in der folgenden Abbildung (12.3) gemacht werden.

```

1  <%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
2  CodeBehind="Default.aspx.cs" Inherits="WebChat.Client.Html_Default" %>
3
4  <%@ Register TagPrefix="luware" TagName="chat" Src="WebChat.ascx" %>
5  <asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
6  </asp:Content>
7  <asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
8
9      <h2>
10         Welcome</h2>
11
12         <table>
13             <tr>
14                 <td>
15                     <a href="Default.aspx">home</a>
16                 </td>
17                 <td>
18                     <a href="about.aspx">about</a>
19                 </td>
20             </tr>
21         </table>
22         <p>
23             &nbsp;   <luware:chat runat="server" />
24         </p>
25     </asp:Content>

```

Abbildung 12.3: Einbindung des WebChatControls

Die erste Zeile registriert den Tag für diese Webseite. Dabei ist darauf zu achten, dass der korrekte Pfad zu den Source Dateien im Parameter Src angegeben wird. Das TagPrefix sowie der TagName kann frei gewählt werden. Nun kann das WebChatControl wie in der zweiten Zeile, an einer beliebigen Stelle, eingebunden werden. Das Prefix und der TagName müssen mit den im Register gewählten übereinstimmen.

Das WebChatControl kann über die folgenden zwei optionalen Parameter noch leicht den eigenen Bedürfnissen angepasst werden.

- chatTitle: Der hier übergebene String wird in der Titelleiste angezeigt.
- cssStyle: Hier können CSS Befehle an das WebChatControl übergeben werden.
 - Das WebChatControl ist eine relativ komplexe Struktur, dieser Parameter funktioniert nur für simple Angaben wie die Hintergrundfarbe. Detail anpassungen müssen im Stylesheet des WebChatControls vorgenommen werden.

Es ist theoretisch möglich, mehr als ein ChatControl in eine einzelne Webseite einzubinden. Da der Chat aber an die Browsersession gebunden ist, ist es nicht möglich verschiedene Chats in einer Webseite zu haben. Alle Controls werden immer dieselbe Session und damit auch dieselben Nachrichten erhalten. Daher gilt es, diese Duplizierung zu vermeiden.

12.3.1 Globale Registrierung des Tags

Um zu vermeiden, dass die Registrierung bei einer grösseren Webapplikation in jeder einzelnen Webseite vorgenommen werden muss, kann das WebChatControl auch global in der Web.Config Datei der Webapplikation eingetragen werden. Dies kann wie in der folgenden Abbildung (12.4) gemacht werden.

```
6  <configuration>
7  <system.web>
8  <pages>
9  <controls>
10 <add src="WebChat.ascx"
11     tagName="chat"
12     tagPrefix="luware" />
13 </controls>
14 </pages>
15 </system.web>
16 </configuration>
```

Abbildung 12.4: Globale Registrierung des WebChatControls

Auch hier gilt es darauf zu achten, dass der korrekte Pfad zu den Source Dateien in Parameter src eingetragen wird. tagName und tagPrefix sind frei wählbar und sind für die gesamte Webapplikation gültig.

13 Glosar

Begriff	Beschreibung
Agent	Angestellter des Contact Centers
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
ASP.Net	Active Server Page .Net
bs	Beat Stettler
Channel	Verbindung zwischen zwei Geräten im Netz
Consumer Integration	Von Luware entwickeltes Interface zu MS Lync
Contact Center	Support Organisation des Anbieters der Webseite
CSS	Cascading Style Sheets
DLL	Dynamic Link Library
DTO	Data Transfer Object
ea	Eric Aebi
H.264	Standard zur effizienten Videokompression
Helpline	Telefonische Hotline
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IIS	Internet Information Services
iOS	iPhone Operating System
JDK	Java Development Kit
JRE	Java Runtime Environment
JVM	Java Virtual Machine
LUCS	Lean Unified Customer Services
Lync	Von Microsoft entwickelte Kommunikations Plattform
MPEG4	Moving Pictures Expert Group
MS	Microsoft
Ogg	Container-Dateiformat für Multimedia-Dateien
pb	Phillip Beck
PHP	An C und Perl angelehnte Skriptsprache
Plugin	Software welche eine Applikation (z.B. einen Browser) erweitert
SDK	Software Development Kit
Silverlight	WebTechnologie für dynamische Webseiten von Microsoft
SIP	Session Initiation Protocol
TFS	Team Foundation Server
UCMA	Unified Communications Managed API

Begriff	Beschreibung
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
Vorbis	Freier Video Code
WCF	Windows Communication Foundation
WebChat Experts	Plugin für MS Lync von der Luware AG
WebChat WebSite	Plugin für MS Lync von der Luware AG

Abbildungsverzeichnis

4.1	Use Case Diagramm	17
6.1	Physische Sicht	26
6.2	Logische Sicht	27
6.3	Klassendiagramm Client.Html	28
6.4	Sequenzdiagramm Start Session	29
6.5	Sequenzdiagramm Send Message	30
6.6	Sequenzdiagramm Receive Message	30
7.1	Webchat minimiert	32
7.2	Webchat Control	33
7.3	Eingabemaske für Telefonnummer	33
7.4	Ausgelagerter Chat	34
11.1	WebChat minimiert	41
11.2	WebChat mit Antwort vom Agenten	41
11.3	WebChat in eigenem Fenster	42
11.4	Eingabefenster für Telefonnummer	43
12.1	IIS Manager	45
12.2	Eingabefenster Add Application	46
12.3	Einbindung des WebChatControls	47
12.4	Globale Registrierung des WebChatControls	48