Dokumentation Studienarbeit

Email Programm

im Fach

.NET Programmierung mit C#

von

Daniel Glaser

Im Studiengang

Mobile Computing

an der Hochschule für angewandte Wissenschaften Hof

Inhaltsverzeichnis

[1 Einführung 3](#_Toc473474256)

[2 Beschreibung der Aufgabenstellung 4](#_Toc473474257)

[2.1 Vorgegebene Anforderungen 4](#_Toc473474258)

[2.2 Eigene Anforderungen 4](#_Toc473474259)

[3 Implementierung: Beschreibung der Lösung 5](#_Toc473474260)

[3.1 Angaben zum Design der Oberfläche 5](#_Toc473474261)

[3.2 Angaben zur Umsetzung 5](#_Toc473474262)

[3.2.1 Beschreibung der Komponenten 5](#_Toc473474263)

[3.2.2 Verwendete Algorithmen, Bibliotheken oder Techniken 6](#_Toc473474264)

[3.2.3 Verwendeter fremder Quelltext 6](#_Toc473474265)

[4 Benutzerhandbuch 7](#_Toc473474266)

[4.1 Hauptfenster 7](#_Toc473474267)

[4.2 Optionen 7](#_Toc473474268)

[4.2.1 Account anlegen 7](#_Toc473474269)

[4.2.2 Account bearbeiten 8](#_Toc473474270)

[4.2.3 Account als Standard definieren 8](#_Toc473474271)

[4.2.4 Account löschen 8](#_Toc473474272)

[4.3 Email schreiben 8](#_Toc473474273)

[4.4 Emails abholen 8](#_Toc473474274)

[4.5 Emails anschauen 8](#_Toc473474275)

[4.5.1 Email beantworten 8](#_Toc473474276)

[4.5.2 Email weiterleiten 8](#_Toc473474277)

[4.5.3 Email als ungelesen markieren 8](#_Toc473474278)

[5 Diskussion 9](#_Toc473474279)

[5.1 Zusammenfassung der Ergebnisse 9](#_Toc473474280)

[5.2 Aufgetretene Probleme 9](#_Toc473474281)

[5.2.1 WebBrowser ruckelt 9](#_Toc473474282)

[5.2.2 Testen des SMTP Server 9](#_Toc473474283)

[5.2.3 Speichern von Anhängen 9](#_Toc473474284)

[5.2.4 InvalidOperationException in der doEmailsContainsId 9](#_Toc473474285)

[5.2.5 Emails abholen im gemeinsamen Posteingang 9](#_Toc473474286)

[5.3 Weiterentwicklung des Programmes 9](#_Toc473474287)

[6 Eidesstattliche Erklärung 10](#_Toc473474288)

# Einführung

Ein Email Programm gehört zu den grundlegendsten Programmen die es gibt, weil es so gut wieder jeder braucht. Viele Email-Programme sind aber mit Funktionen überladen und brauchen deswegen lange zum Starten oder haben eine unübersichtliche Oberfläche. Deswegen habe ich mir in dieser Studienarbeit als Ziel genommen, ein Email Programm zu programmieren, welches schnell startet und eine übersichtliche Oberfläche hat, damit es leichter zu bedienen ist. Es soll flüssig laufen und die wichtigsten Funktionen bieten.

Als Vorbild habe ich mir andere Email Programme wie z.B. Thunderbird aber auch Online-Email-Dienste wie z.B. Inbox herangezogen.

# Beschreibung der Aufgabenstellung

## Vorgegebene Anforderungen

Als Studienarbeit wird ein mit C# und XAML erstelltes Programm mit grafischer Oberfläche entwickelt, dass folgende Anforderungen erfüllt:

* Programm verwendet bzw. verarbeitet Daten, die aus einer beliebigen Quelle (z.B. Webservice, Datenbank, Datei) stammen können
* Daten müssen gespeichert und geladen werden können
* Nutzung von WPF für die GUI
  + Ansprechendes Design der GUI
  + Verwendung passender Steuerelemente
  + Layout der GUI passt sich an Größenänderungen des Fensters an
* Oberfläche wird in XAML-Dateien definiert, nicht im Code erzeugt
* Konsequente Nutzung des MVVM-Patterns, beachten Sie dazu:
  + Aufteilung in Model, View und ViewModel
  + Datenbindung
  + Code-Behind enthält nur den nötigsten Code
* Trennung in 2 Teilprojekte:
  + (Portable) oder Class Library)
    - Enthält Model-Klassen, Klassen für Datenzugriff sowie sonstige Klassen (z.B. für Utilities)
    - Enthält ViewModels
    - Targets für Portable Class Library (mind.): sh. Abbildung 1
  + Application)
    - Referenz auf Projekt „Common“
    - Enthält nur die Views sowie direkt dazugehörigen Code
    - Target Framework: 4.5 bzw. 4.6 (passend zum anderen Projekt!)
* Aufteilung der Quellcodedateien in eine sinnvolle Klassenhierarchie, d.h. Verwendung von Namespaces. Ein einfaches, nicht vollständiges Beispiel sh. Abbildung 2.
* Objektorientierte Programmierung
* Anwendung asynchroner Programmierung bei länger dauernden Aufgaben
* Mindestens 1500 Lines of Code

## Eigene Anforderungen

Es soll ein Email Programm entstehen das zuerst einmal die Grundfunktionen Email schreiben und Emails lesen ermöglichen soll. Dabei sollen HTML E-Mails richtig dargestellt werden.

Es soll möglich sein zu erkennen welche Emails bereits gelesen wurden und auch eine E-Mail wieder als ungelesen markieren können.

Die Emails sollen auch gelöscht werden können. Lokal aber auch auf dem Server.

Eine E-Mail soll man weiterleiten können und darauf antworten können. Auch soll es möglich sein allen zu antworten an die die E-Mail ging.

Bei Emails mit Anhang soll der Anhang zu öffnen oder speichern sein. Auch soll man bei einer Email einen Anhang mit verschicken können.

Es soll die Möglichkeit geben mehrere Accounts anzulegen und zu verwalten.

# Implementierung: Beschreibung der Lösung

## Angaben zum Design der Oberfläche

Beim Design wurde sich an Thunderbird orientiert aber trotzdem ein eigenes Design erstellt.

Als Farben wurde weiß oder blau verwendet, da es zu dem modernen Flat Design passen soll.

## Angaben zur Umsetzung

### Beschreibung der Komponenten

#### Email

Die Email-Klasse ist eine selbst erstellte Klasse die eine E-Mail repräsentiert. Sie beinhaltet alles was für dieses E-Mail-Programm von einer E-Mail benötigt wird.

#### Account

Diese Klasse repräsentiert einen E-Mail-Account mit allen Parametern die ein E-Mail-Account in diesem Programm benötigt.

Es beinhaltet eine Liste aller Emails die zu diesem Account gehören.

#### EmailService

Der EmailService kümmert sich darum Emails zu senden und zu empfangen. Außerdem hat er noch Methoden mit dehnen der IMAP, POP3 und SMTP Server getestet werden können.

##### Emails senden

Zum Senden von Emails wird der SmtpClient von .Net verwendet.

##### Emails abholen

Für das Abholen von Emails wie entweder der ImapClient oder der Pop3Client des MailKit verwendet je nachdem welche Verbindung der Nutzer eingegeben hat.

##### Server testen

Zum Testen der Server werden die 3 Clients des MailKits verwendet um sich mit dem jeweiligen Server zu verbinden. Dabei wird überprüft ob es erfolgreich klappt oder nicht und dem Nutzer dann grafisch dargestellt.

#### EmailViewModel

Das EmailViewModel beinhaltet alles was auch eine E-Mail beinhaltet aber die Listen sind hier ObservableCollections.

#### AccountViewModel

Das AccountViewModel beinhaltet alles was auch ein Account beinhaltet aber die Listen sind hier ObservableCollections.

#### AccountListViewModel

Das AccountListViewModel beinhaltet eine Liste aller AccountViewModels. Sie hat zudem Methoden mit dehen sie auf die MailService Klasse zugreift um die Funktionen daraus zu nutzen.

### Verwendete Algorithmen, Bibliotheken oder Techniken

#### MailKit

Für alles was mit Emails abholen, senden, löschen, als gelesen markieren usw. zu tun hat wurde das MailKit verwendet. Es bietet einen ImapClient, Pop3Client und SmtpClient.

Der ImapClient und Pop3Client wurden für das Abholen der E-Mails verwendet und für die Testfunktion beim Account anlegen.

Der SmtpClient wurde nur zum Testen genommen da er dafür einfacher einzusetzen war als der SmtpClient von .Net oder ein TcpClient. Für das Senden von Emails wurde der SmtpClient von .NET verwendet.

#### WebBrowser

Die Anzeige von HTML-Emails ist durch den WebBrowser von CSharp umgesetzt worden.

### Verwendeter fremder Quelltext

#### Dispatcher

Vom Herr Rill habe ich zur Hilfe folgenden Code erhalten um eine E-Mail in einem Task zu den Emails eines Accounts hinzuzufügen:

var currentAccount = account;

var currentEmail = email;

dispatcher.BeginInvoke((Action)(() =>

{

currentAccount.Emails.Add(currentEmail);

}));

Den Code habe ich auch noch an anderen Stellen verwendet wo dieselbe Situation enstand.

#### DataServices

Die Klasse DataServices habe ich von Herr Rill seinem Beispiel Projekt „DataBindingAndMVVM“ aus der Solution „11 MVVMContactManager“ kopiert und an für mein Programm angepasst.

#### Emails abholen

Der Code um Emails abzuholen wurde von <http://www.mimekit.net/docs/html/T_MailKit_Net_Imap_ImapClient.htm> und <http://www.mimekit.net/docs/html/T_MailKit_Net_Pop3_Pop3Client.htm> verwendet. Er wurde aber kombiniert und erweitert.

#### Emails senden

Der Code um Emails zu senden wurde von <https://code-bude.net/2011/06/14/emails-versenden-in-csharp/> und <http://www.mimekit.net/docs/html/T_MailKit_Net_Smtp_SmtpClient.htm> kombiniert und abgeändert.

#### Kleinere Code Schnipsel

Kleinere Code Schnipsel wie z.B. zum Sortieren einer ListView wurden im Code markiert. Aber zu kleine Code Schnipsel wo nur nachgeschaut wurden wie etwas heißt oder funktioniert wurden nicht markiert.

# Benutzerhandbuch

Bitte starten Sie als erstes das das Programm „Email Programm“.

## Hauptfenster

Im Hauptfenster können Sie ihre Email sehen. Links können Sie zwischen den Posteingängen Ihrer angelegten Accounts wechseln oder wie in der Standardansicht den Gemeinsamen Posteingang sehen, in dem sich alle Emails aus all Ihren Accounts befinden.

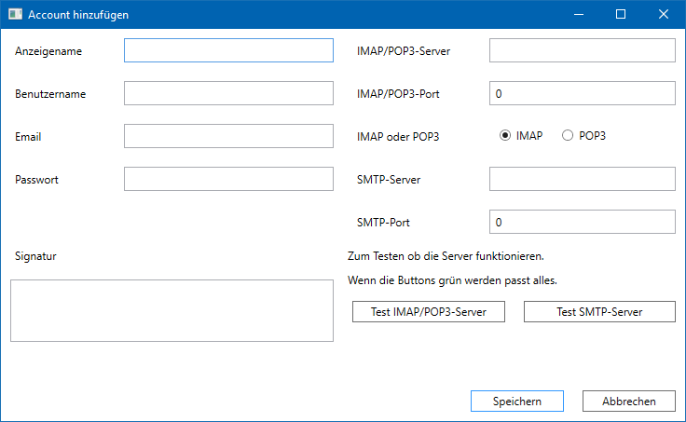
## Optionen

In den Optionen verwalten Sie Ihre Accounts.

Zum Öffnen von diesen gehen Sie bitte in der Menüleiste auf „Optionen“ und klicken auf „Optionen“. Dort können Sie einen Account anlegen, bearbeiten und löschen. Außerdem können Sie Ihren Standard Account für den Email Versand wählen.

### Account anlegen

Klicken Sie bitte im Optionen-Fenster auf den Button „Hinzufügen“ um einen neuen Account anzulegen. Daraufhin öffnet sich ein neues Fenster in dem Sie folgende Informationen eingeben müssen:

* Anzeigename
* Username
* Email
* Passwort
* IMAP/POP3-Server
* IMAP/POP3-Port
* IMAP oder POP3
* SMTP-Server
* SMTP-Port

Zum Testen ob die eingegebenen Server und Ports richtig sind, stehen Ihnen zwei Buttons zur Verfügung. Wenn diese grün werden sind die eingebenden Server und Ports richtig. Falls etwas nicht stimmen sollte werden sie rot und es steht eine Fehlermeldung unter den Buttons die Ihnen helfen soll den Fehler zu finden.

Wenn Sie fertig sind können Sie mit einem Klick auf den „Speichern“-Button den Account speichern.

### Account bearbeiten

Zum Bearbeiten eines Accounts gehen Sie bitte in die Optionen. Im Optionen-Fenster wählen Sie bitte den zu bearbeitenden Account aus und klicken auf den Button „Bearbeiten“. Alternativ können Sie auch auf den Account doppelt klicken. Darauf hin öffnet sich ein Fenster in dem Sie den Account bearbeiten können. Die Vorgehensweiße zum Bearbeiten ist genauso wie beim Account anlegen.

### Account als Standard definieren

Um einen Account als ihr Standard Account zum Versenden von E-Mails zu definieren gehen Sie bitte in die Optionen und wählen den Account in der Liste aus den Sie als ihr Standard Account haben möchten und klicken Sie auf „Als Standard“.

### Account löschen

Um einen Account zu löschen gehen sie bitte in die Optionen und wählen den Account in der Liste aus den Sie löschen möchten und klicken Sie auf „Löschen“.

## Email schreiben

Um eine E-Mail zu schreiben gehen Sie bitte in das Hauptfenster und klicken auf den Button „Email schreiben“. Alternativ können Sie auch über die Menüleiste dorthin navigieren. Dazu in der Menüleiste den Punkt „Email“ auswählen und dann „Email schreiben“.

Es öffnet sich ein Fenster in dem Sie die Sendeadresse aus den Accounts die Sie angelegt haben auswählen können. Im nächsten Feld können Sie die Empfangsadresse/n (bei mehreren mit “,“ getrennt) eingeben an die die E-Mail gehen soll. Sie haben auch noch die Möglichkeit unter CC und BCC E-Mail-Adressen einzugeben. Diese funktionieren wie standardmäßig vorgesehen.

In dem großen Eingabefeld können Sie nun Ihre Nachricht eingeben die Sie schicken möchten. Sie können auch noch über den Button „Anhänge hinzufügen“ Dateien auswählen die Sie mitschicken möchten.

## Emails abholen

Bitte gehen Sie in das Hauptfenster der Anwendung und wählen Sie entweder einen Account aus oder Gemeinsamer Posteingang und klicken Sie auf „Emails abholen“. Dann werden im Hintergrund die entsprechenden Emails abgeholt und im Hauptfenster aufgelistet.

## Emails anschauen

Bitte klicken Sie im Hauptfenster links auf den jeweiligen Posteingang aus welchem Sie die Emails sehen möchten. In der Liste klicken Sie nun auf die E-Mail die Sie sehen möchten doppelt um sie zu öffnen.

### Email beantworten

Bitte öffnen Sie eine E-Mail und klicken Sie auf „Antworten“. Es öffnet sich das Fenster zum Email schreiben. Schreiben Sie ihre Nachricht und klicken Sie auf „Senden“. Wenn Sie allen antworten möchten klicken Sie bitte auf „Alle antworten“.

### Email weiterleiten

Bitte öffnen Sie eine E-Mail und klicken Sie auf „Weiterleiten“. Es öffnet sich das Fenster zum Email schreiben. Füllen Sie die Felder nach Ihren Belieben aus. Wenn Sie fertig sind klicken Sie auf „Senden“.

### Email als ungelesen markieren

Bitte öffnen Sie eine E-Mail und klicken Sie auf „Ungelesen“.

# Diskussion

## Zusammenfassung der Ergebnisse

Die Grundfunktionen wie Senden und Empfangen von Text E-Mails funktionieren. Die Anzeige von HTML-E-Mails funktioniert. Das Antworten, alle antworten und weiterleiten von E-Mails funktioniert. Es ist möglich mehrere Accounts mit IMAP oder POP3 anzulegen und zu bearbeiten. Zwischen den Posteingängen der Account kann gewechselt werden und alle E-Mails aller Accounts können unter Gemeinsamer Posteingang ansehen werden. E-Mails können lokal gelöscht werden und werden lokal auch als gelesen oder ungelesen markiert.

Was noch nicht funktioniert ist das Löschen von E-Mails auf dem Server. Das E-Mails gelesen wurde wird bei IMAP noch nicht an den Server übermittelt. Weiteres siehe Aufgetretene Probleme.

Ich würde gerne noch einiges beheben oder verbessern, aber ich habe leider keine Zeit mehr.

## Aufgetretene Probleme

### WebBrowser ruckelt

Der WebBrowser ruckelt bei der Änderung der Größe des Fensters. Konnte aber keine Lösung gefunden werden. Liegt vermutlich daran das die WebBrowser View zu oft rendert.

### Testen des SMTP Server

Das Testen des SMTP Server funktioniert noch nicht ganz so wie gewollt. Näheres im Code (EmailService Zeile 133)

### Speichern von Anhängen

Das Speichern von Anhängen funktioniert noch nicht. Da ich von dem MailKit einen Anhang als MimeEntity bekomme der wiederrum einen MimePart beinhaltet und ich diesen nicht serialisieren kann. Mögliche Lösung: Ich könnte die Anhänge gleich beim Abholen speichern und den Nutzer diese dann nur verschieben lassen.

### InvalidOperationException in der doEmailsContainsId

In der doEmailsContainsId tritt ab und zu eine InvalidOperationException auf. Auch obwohl ich nun eine neue List der ID erstelle scheint sie immer noch ab und zu aufzutreten. Habe sie nun abgefangen und eine Ausgabe in der Konsole eingebaut falls sie auftritt. Die Methode liefert dann das die E-Mail nicht enthalten sei und das Programm läuft einfach weiter.

### Emails abholen im gemeinsamen Posteingang

Die Emails tauchen erst auf wenn man noch mal auf einen anderen Account und zurück wechselt. Das liegt daran das die Emails hier aus den ganzen Accounts gesammelt werden und beim Klicken auf den „Gemeinsamer Posteingang“-Button als ItemSource gesetzt werden.

Mögliche Lösung: Man müsste alle Emails in einer ObservableCollection haben oder eine eigene ObservableCollection für alle Emails. Ich habe zwar eine ObservableCollection für die alle Emails erstellt aber nur in der Code Behind ohne das INotifyPropertyChanged Interface.

## Weiterentwicklung des Programmes

Als erstes könnten noch die Probleme behoben werden die noch vorhanden sind und dann könnte das Programm noch um weitere Email Funktionen erweitert werden. Dazu zählen zum Beispiel:

* Anhänge wieder abwählen können
* Einen Namen eingeben können der zu der Email Adresse gehört
* Senden von HTML Emails
* Email als Entwurf speichern

# Eidesstattliche Erklärung

Ich versichere, dass ich die Studienarbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ort, Datum Unterschrift