21.9.2013

**Machbarkeitsstudie**

**S/Chat**

**Verschlüsselter Android Chat**

**Elias Frantar, Wolfram Soyka, Gary Ye**

|  |  |  |  |
| --- | --- | --- | --- |
|  | *Name* | *Datum* | *Unterschrift* |
| *erstellt* | *Elias Frantar* | *17.9.2013* |  |
| *geprüft* |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Version* | *Name* | *QS* | *Datum* | *Kommentar* |
| *0.1* | *Frantar* |  | *17.9.2013* | *Einführung, Ist-Zustand(Einleitung), Meilensteinplan* |
| *0.2* | *Frantar* |  | *18.9.2013* | *Projektorganisation, Technologien, Glossar* |
| *0.3* | *Frantar* |  | *21.9.2013* | *Nutzwertanalyse, Soll-Zustand* |
| *0.4* | *Ye* | *Fr.* | *21.9.2013* | *PSP* |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 

1)    Einleitung

Mit Aufdeckung des NSA-Skandals wird das Thema Datensicherheit zu einem essenziellen Bestandteil der Softwareentwicklung. Aktuell existieren nur sehr wenige Programme, die eine sichere Kommunikation ermöglichen. Von diesen wenigen sind wiederrum nur sehr wenige benutzerfreundlich und werden daher kaum von Privatpersonen eingesetzt.

Deshalb wollen wir in unserem Projekt eine benutzerfreundliche App entwickeln, die es den Usern ermöglicht, sicher Nachrichten zu versenden, ohne jegliches Grundwissen über Verschlüsselungen besitzen zu müssen.

2)    Ist-Zustand

Aktuell existieren nur sehr wenige Programme, die eine sichere Kommunikation ermöglichen. Von diesen wenigen sind wiederrum nur sehr wenige benutzerfreundlich und werden daher kaum von Privatpersonen eingesetzt.

2.1 Marktanalyse:

Es gibt derzeit zwar viele ähnliche Systeme, wie rentalcars oder check24, aber keines ist offen zugänglich oder käuflich erwerbbar und keines erfüllt alle Kriterien und Funktionen der Flitzer AG, deshalb muss ein neues individuelles System erstellt werden.

2.2 Trendanalyse:

Da in diesem Projekt nur eine einfache Website erstellt wird, soll es auch möglich sein das Bestellsystem sowie alle anderen Funktionen problemlos in ein komplexeres Design zu integrieren. Das System kann daher auch an andere Firmen weiterverkauft werden.

3)    Soll-Zustand

**Muss – Ziele:**

1. **Sicherheit**

Die Sicherheit der Nachrichten ist der wichtigste Punkt dieses Projektes, deshalb ist die Implementierung geeigneter Verschlüsselungssysteme (internationale Standards: RSA, AES, SHA3) und Protokolle essenziell.

1. **Bedienbarkeit**

Die App soll benutzerfreundlich und damit leicht bedienbar sein. Wichtig hierbei ist, dass ein User ohne Kenntnisse über Verschlüsselung die App problemlos und ohne Zusatzaufwand bedienen kann.

1. **Zuverlässigkeit**

Das Verschicken der Nachrichten soll zuverlässig funktionieren. D.h. konkret, dass bei guter Internetverbindung rund 95% aller Nachrichten beim ersten Versenden ankommen und 99% beim (automatischen) Wiederholen.

**Kann – Ziele:**

1. **Kompatibilität**

Die App soll von Android-Version 4.x bis 2.3 abwärtskompatibel sein ohne etwaige Funktionalitäten zu verlieren.

1. **Gruppen**

Die App soll nicht nur Nachrichten mit einer Person, sondern auch in Gruppen von mehreren Personen gleichzeitig austauschen können.

**Nicht – Ziele**:

1. **Design**

Ein spezielles modernes Design der Applikation für das extra angefertigte Bedienungselemente notwendig sind.

1. **Funktionen**

Bereitstellen spezielle Zusatzfunktionen, wie das Versenden von Bild oder Videodateien oder das Erstellen eines detaillierten persönlichen Kontaktprofils

4) Produktfunktionen

**4.1 An und Abmelden:**

4.1.1 Benutzer registrieren (LF010):

Jeder Benutzer kann sich selbständig unter Angabe seiner persönlichen Daten und eines Passworts im System registrieren. Danach hat er Zugriff auf das Bestellsystem.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Benutzer registrieren (LF010)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Jeder Benutzer kann sich selbständig unter Angabe seiner persönlichen Daten und eines Passworts im System registrieren. Danach hat er Zugriff auf das Bestellsystem.  *Auslöser:* Benutzer möchte sich für Bestellsystem registrieren.  *Ergebnis:* Benutzer hat eigenes Bestellkonto.  *Akteure:* neuer Benutzer  *Eingehende Informationen:* persönliche Daten (siehe LD010), Passwort  *Vorbedingungen:* Benutzer darf noch nicht registriert sein.  *Nachbedingungen:* Benutzer kann sich im System anmelden. | hoch | mittel | Must Have |

4.1.2 Benutzer anmelden (LF020):

Ein Benutzer kann sich unter Angabe seiner registrierten Email – Adresse und seines Passworts im System anmelden.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Benutzer anmelden (LF020)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein Benutzer kann sich unter Angabe seiner registrierten Email – Adresse und seines Passworts im System anmelden.  *Auslöser:* Benutzer möchte sich zum Bestellen anmelden.  *Ergebnis:* Benutzer ist angemeldet.  Akteure: registrierter Benutzer  *Eingehende Informationen:* registrierte Email - Adresse, Passwort  *Vorbedingungen:* Benutzer muss bereits registriert sein.  *Nachbedingungen:* Benutzer kann das Bestellsystem nutzen. | hoch | mittel | Must Have |

4.1.3 Benutzer abmelden (LF030):

Ein angemeldeter Benutzer kann sich jederzeit vom System abmelden.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Benutzer abmelden (LF030)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein angemeldeter Benutzer kann sich jederzeit vom System abmelden.  *Auslöser:* Angemeldeter Benutzer möchte sich von der Website abmelden.  *Ergebnis:* Benutzer ist abgemeldet. Er kann nicht mehr auf das Bestellsystem zugreifen.  *Akteure:* angemeldeter Benutzer  *Eingehende Informationen:* keine  *Vorbedingungen:* Benutzer muss im System angemeldet sein.  *Nachbedingungen:* Benutzer kann sich jederzeit wieder anmelden. | hoch | gering | Must Have |

4.1.4 Passwort vergessen (LF040):

Sollte ein Benutzer sein Passwort vergessen haben, kann er es sich jederzeit an seine registrierte Email – Adresse zuschicken lassen.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Passwort vergessen (LF040)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Sollte ein Benutzer sein Passwort vergessen haben, kann er es sich jederzeit an seine registrierte Email – Adresse zuschicken lassen.  *Auslöser:* Benutzer möchte sein vergessenes Passwort erfahren  *Ergebnis:* Benutzer erfährt sein vergessenes Passwort  *Akteure:* registrierter Benutzer  *Eingehende Informationen:* registrierte Email - Adresse  *Vorbedingungen:* Benutzer muss bereits registriert sein.  *Nachbedingungen:* Benutzer kann sich wieder anmelden. | hoch | gering | Should Have |

**4.2 Benutzerdaten**

Ein Benutzer verfügt über die persönlichen Daten, die bei der Registrierung angegeben wurden.

4.2.1 Benutzerdaten anzeigen (LF110):

Ein angemeldeter Benutzer kann sich jederzeit seine aktuell gespeicherten Daten vom System anzeigen lassen.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Benutzerdaten anzeigen (LF110)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein angemeldeter Benutzer kann sich jederzeit seine aktuell gespeicherten Daten vom System anzeigen lassen.  *Auslöser:* Benutzer möchte seine derzeit gespeicherten persönlichen Daten erfahren.  *Ergebnis:* Benutzer erfährt seine aktuell gespeicherten persönlichen Daten  *Akteure:* angemeldeter Benutzer  *Eingehende Informationen:* keine  *Vorbedingungen:* Benutzer muss bereits angemeldet sein.  *Nachbedingungen:* Benutzer kann nun die Daten gegebenenfalls ändern lassen. | hoch | mittel | Must  Have |

4.2.2 Benutzerdaten ändern (LF120):

Ein angemeldeter Benutzer kann jederzeit eine Anfrage zur Änderung seiner persönlichen Daten an den Administrator senden.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Benutzerdaten ändern (LF120)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein angemeldeter Benutzer kann jederzeit eine Anfrage zur Änderung seiner persönlichen Daten an den Administrator senden.  *Auslöser:* Benutzer möchte seine derzeit gespeicherten persönlichen Daten ändern.  *Ergebnis:* Die Daten werden gegebenenfalls (bei sinnvoller Änderung) geändert.  *Akteure:* angemeldeter Benutzer, Administrator  *Eingehende Informationen:* die zu ändernden Daten  *Vorbedingungen:* Benutzer muss bereits angemeldet sein.  *Nachbedingungen:* Die Daten stehen dem Administrator zur Überprüfung und zum nachfolgenden Update zur Verfügung | hoch | mittel | Must  Have |

**4.3 Suche**

Ein Benutzer kann verschieden Suchfunktionen nutzen. Er muss dazu weder angemeldet noch registriert sein.

4.3.1 Fahrzeuge suchen (LF210):

Ein Benutzer kann Fahrzeugsuchanfragen nach folgenden Kriterien stellen: Automarke, Farbe, Preis, genauer Fahrzeugname, Anzahl an bereits gefahrenen Kilometern, Verfügbarkeit

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Fahrzeuge suchen (LF210)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein Benutzer kann Fahrzeugsuchanfragen nach verschiedenen Kriterien (siehe oben) stellen.  *Auslöser:* Benutzer möchte bestimmte Fahrzeuge finden.  *Ergebnis:* Benutzer bekommt eine Liste aller Fahrzeuge, die auf diese Kriterien zutreffen.  *Akteure:* Benutzer (muss nicht registriert oder angemeldet sein)  *Eingehende Informationen:* Suchkriterien  *Vorbedingungen:* keine  *Nachbedingungen:* Der Benutzer kann sich anmelden / registrieren und eines der angezeigten Fahrzeuge bestellen. | hoch | mittel | Must  Have |

4.3.2 Fahrzeuge sortieren (LF220):

Ein Benutzer kann sämtliche Fahrzeuge nach folgenden Kriterien sortiert anzeigen lassen:

Alphabetisch, Automarke, Anzahl an bereits gefahrenen Kilometern, Farbe

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Fahrzeuge sortieren (LF220)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein Benutzer kann sämtliche Fahrzeuge nach bestimmten Kriterien (siehe oben) sortiert anzeigen lassen.  *Auslöser:* Der Benutzer möchte bestimmte Fahrzeuge sehen und vergleichen  *Ergebnis:* Der Benutzer sieht die Fahrzeuge nach gewünschten Kriterien sortiert.  *Akteure:* Benutzer (muss weder angemeldet noch registriert sein)  *Eingehende Informationen:* Sortierkriterien  *Vorbedingungen:* keine  *Nachbedingungen:* Der Benutzer kann die Fahrzeuge vergleichen und sich anmelden / registrieren, um eines zu bestellen. | hoch | mittel | Should  Have |

**4.4 Feedback**

Ein Benutzer kann freiwillig ein Feedback über unser Unternehmen auf der Website hinterlassen. Als Gegenleistung dafür erhält er gelegentlich Sonderangebote über die er per Email informiert wird

4.4.1 Feedback abgeben (LF310):

Ein angemeldeter Benutzer kann nach Rückgabe eines Fahrzeuges ein für alle Webseitenbenutzer sichtbares Feedback über das Unternehmen hinterlassen.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Feedback abgeben (LF310)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein Benutzer kann nach Rückgabe eines Fahrzeuges ein für alle Webseitenbenutzer sichtbares Feedback über das Unternehmen hinterlassen.  *Auslöser:* Der Benutzer möchte seine Meinung übe unser Unternehmen äußern.  *Ergebnis:* Das Feedback ist für alle Webseitenbenutzer sichtbar.  *Akteure:* angemeldeter Benutzer nach Rückgabe eines Fahrzeugs  *Eingehende Informationen:* das Feedback  *Vorbedingungen:* Benutzer muss angemeldet sein und gerade ein gemietetes Fahrzeug zurückgebracht haben.  *Nachbedingungen:* Der Benutzer ist für das Sonderangebotsprogramm registriert. | mittel | gering | Nice To  Have |

4.5 Bestellung:

Ein angemeldeter Benutzer kann jederzeit ein verfügbares Fahrzeug bestellen oder ein nicht-verfügbares reservieren. Die Datenbank wird daraufhin automatisch upgedatet.

4.5.1 Fahrzeug bestellen:

Ein angemeldeter Benutzer kann ein Fahrzeug auf Knopfdruck bestellen. Diese Bestellung ist nach einer Bestätigung verbindlich. Die Datenbank wird daraufhin automatisch upgedatet.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Fahrzeug bestellen (LF410)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein Benutzer kann ein Fahrzeug auf Knopfdruck bestellen.  *Auslöser:* Der Benutzer möchte ein Fahrzeug mieten und online bestellen.  *Ergebnis:* Der Benutzer hat das Fahrzeug verbindlich bestellt.  *Akteure:* angemeldeter Benutzer  *Eingehende Informationen:* Dauer der Fahrzeugmietung  *Vorbedingungen:* Benutzer muss angemeldet sein  *Nachbedingungen:* Der Benutzer kann das Fahrzeug nun, wie im Bestellvertrag vereinbart, abholen. | hoch | gering | Must Have |

4.5.2 Fahrzeug reservieren:

Ein Benutzer kann ein derzeit nicht verfügbares Fahrzeug auf Knopfdruck reservieren, um bei Verfügbarkeit automatisch darüber reserviert zu werden.

|  |  |  |  |
| --- | --- | --- | --- |
| Funktion | Nutzen | Aufwand | Priorität |
| UseCase:  *Name:* Fahrzeug reservieren (LF420)  *Art:* Anwendungsfall  *Kurzbeschreibung:* Ein Benutzer kann ein derzeit nicht verfügbares Fahrzeug reservieren.  *Auslöser:* Der Benutzer möchte ein derzeit nicht verfügbares Fahrzeug reservieren, um es sobald es wieder verfügbar ist bestellen zu können.  *Ergebnis:* Der Benutzer hat das Fahrzeug unverbindlich reserviert.  *Akteure:* angemeldeter Benutzer  *Eingehende Informationen:* keine  *Vorbedingungen:* Benutzer muss angemeldet sein  *Nachbedingungen:* Der Benutzer wird, sollte das Fahrzeug wieder verfügbar sein automatisch per Email darüber informiert und kann es bestellen. | hoch | gering | Must Have |

5) Machbarkeit

5.1 Technische Machbarkeit

5.1.1 Technologien:

5.1.1 Verschlüsselungsalgorithmen:

Für dieses Projekt werden kryptografische Algorithmen verschiedenster Art verwendet werden. Konkret werden ein Verfahren für den Schlüsselaustausch, ein symmetrisches Verschlüsselungsverfahren, ein digitales Signaturverfahren und eine kryptografische Hashfunktion benötigt.  
Alle im folgenden Abschnitt behandelte Algorithmen gelten als sicher, d.h. sie werden schon über ein längeren Zeitraum verwendet, von Experten analysiert und es wurden bisher noch keine Sicherheitslücken entdeckt.   
  
*(Alle im folgenden Abschnitt behandelte kryptografische Begriffe können im Glossar gefunden werden)*5.1.1.1 Schlüsselaustauschverfahren:

Grundsätzlich gibt es für den sicheren Schlüsselaustausch zwei verschieden Möglichkeiten: ein *asymmetrisches Verschlüsselungsverfahren* oder der *Diffie-Hellman-Schlüsselaustausch*.  
Letzterer hat den entscheidenden Nachteil, dass mit jedem einzelnen Kommunikationspartner ein eigener gemeinsamer Schlüssel generiert werden muss. Noch dazu kommt, dass ein asymmetrisches Verfahren gleichzeitig zur Verschlüsselung, *Signierung* und *Authentifizierung* verwendet werden kann. Deshalb wird ein asymmetrisches Verfahren verwendet werden.

Hier gibt es eigentlich auch nur zwei infrage kommende Möglichkeiten: der bewehrte und standardisierte *RSA* oder ein Verfahren auf elliptischen Kurven basierendes Verfahren (*ECC*).  
Erstere wird schon seit den 90iger-Jahren von zahlreichen Kryptoprodukten eingesetzt und hat bisher noch keine (bei fehlerfreier Implementierung) sicherheitskritische Schwäche offenbart. Ein Nachteil dieses Verfahrens ist aber, dass sehr lange Schlüssel benötigt werden (2048bit) um eine hohe Sicherheit zu garantieren. Des Weiteren ist die Schlüsselgenerierung keinesfalls trivial, da hierzu sehr lange voneinander unabhängige zufällige Primzahlen gefunden werden müssen. Im Gegensatz dazu hat ein ECC-Verfahren den entscheidenden Vorteil, dass es vergleichbare Sicherheit mit wesentlich geringerer Schlüssellänge und damit wesentlich höherer Performence ermöglicht. Die Kryptografie auf Basis elliptischer Kurven ist aber relativ neu und wurde bisher kaum eingesetzt und ist daher vergleichsweise weniger analysiert und damit weniger bewehrt als RSA. Außerdem gibt es in diesem Bereich noch kaum international anerkannte Standards. Aus diesem Grund werden wir in unserer Applikation das RSA-Verfahren einsetzen, um ein höheres Vertrauen der User in unsere Applikation zu erlangen.

5.1.1.2 Symmetrisches Verschlüsselungsverfahren:

Grundsätzlich gibt es *Blockchiffren* und *Stromchiffren*, wobei erste viel bewehrter sind und ein Schlüssel (unter richtiger Anwendung) ohne Sicherheitsrisiko mehrmals verwendet werden kann. Deshalb werden wir eindeutig ein symmetrisches Verfahren implementieren.

Hier gibt es sehr viele Möglichkeiten, wobei die bekanntesten und damit bewehrtesten wohl die Finalisten der *AES* (Advanced Encryption Standard)-Wettbewerbs sind. Da der Sieger dieses Wettbewerbs seit 2001 in zahlreichen Kryptoprodukten implementiert wurden und sich auch nach dem Wettbewerb trotz zahlreicher Analyseversuche keine Sicherheitslücken gefunden wurden, werden wir auch auf diesen Algorithmus (AES) setzen, da dieser wahrscheinlich auch bei den Usern auf die höchste Akzeptanz stößt.

5.1.1.3 Digitale Signatur:

Da der sowieso verwendet RSA auch als bewehrtes Signaturverfahren eingesetzt werden kann und die beste Alternative der *DSA* (Digital Signature Standard) einen weiteren Schlüssel pro Kommunikationspartner erfordert, werden wir auch hier RSA verwenden.

5.1.1.4 Hashfunktionen:

Für das Projekt wird zuerst einmal eine *schlüsselabhängige Hashfunktion* verwendet. Grundsätzlich kann jede Blockchiffre(in unserem Fall AES) in einer speziellen Betriebsart als schlüsselabhängige Hashfunktion eingesetzt werden. Deshalb werden wir auch hier den AES einsetzen, um auf zusätzliche redundante Algorithmen verzichten zu können.

Für eine schlüsselunabhängige Hashfunktion gibt es viele verschiede gute Algorithmen, wie z.B. *SHA2, SHA3, SHA-512*, die sich nicht wesentlich voneinander unterscheiden. Daher kann das Verfahren nach persönlicher Präferenz ausgewählt werden. Wir haben uns für den neuesten, 2012 gewählten Sieger der SHA(Secure Hash Algorithm) Challenge, *Keccak* bzw. SHA3, da wir nur standardisierte und von vielen Experten geprüfte Algorithmen verwenden wollen.

Die genauen Entscheidungskriterien für die einzelnen Verschlüsselungsalgorithmen sind der Nutzwertanalyse zu entnehmen.

5.1.2 Umsetzung:

Es gibt zwar schon unzählige ähnliche Systeme in der Marktwirtschaft, aber keines erfüllt alle Kriterien und Funktionalitäten, deshalb wird ein genau auf die Wünsche der Firma Flitzer AG abgestimmtes, neues Produkt entwickelt.

5.1.2.1 Betriebssystem:

Für das Betriebssystem des Servers gibt es drei verschiedene Varianten, die in Frage kommen würden: Windows-Server, Linux-Server, Unix-Server

Die Vorteile des Windows Servers liegen eindeutig in der einfachen Bedienung und der Erfahrung der Teamitglieder mit diesem System. Der Nachteil wiederum sind aber die relativ hohen Anschaffungskosten.

Der eindeutige Vorteil des Linux Servers ist natürlich, dass er opensource und kostenlos, ist. Der Nachteil aber wiederrum ist die schwierigere Bedienung und die geringere Erfahrung der Teammitglieder mit diesem System. Adi Shamir hat aber bereits einige Erfahrungen mit diesem Betriebssystem.

Der Unix Server wird oft bei professionellen Systemen eingesetzt. Er kommt für dieses Projekt aber eher weniger in Frage in Frage, da er erstens Geld kostet, zweitens das Team kaum Erfahrung damit hat (ausgenommen Adi Shamir mit Linux) und drittens er für unser System nicht notwendig ist, da es nicht mit übermäßig hohen Datenmengen operieren muss.

Die Wahl des Serverbetriebssystems ist aus der Nutzwertanalyse zu ersehen. Die vom Betriebssystem unabhängigen Protokolle werden im folgenden Abschnitt behandelt.

5.1.2.2 Software / Protokolle:

Die Website wird mit Adobe Dreamweaver entwickelt. Der Vorteil gegenüber reinem HTML besteht in der einfacheren und schnelleren Webseitenkreation und der sehr einfach und schnell zu erlernenden Bedienung. Zusätzlich werden weitere Webseitenfunktionen in PHP eingebaut. Der Vorteil gegenüber JavaScript ist, dass sämtliche Berechnungen am Server durchgeführt werden, was bei der relativ großen Datenbank vor allem bei nicht leistungsstarken Endgeräten eine wichtige Rolle für die Performance spielt. Außerdem kann PHP in einem Webbrowser nicht unbeabsichtigt deaktiviert werden. PHP ist aber definitiv schwerer zu erlernen und anzuwenden. Als Verschlüsselungssysteme werden die internationalen Standards AES, RSA, SHA3 eingesetzt. Da diese auch für Dokumente der höchsten Geheimhaltungsstufe verwendet werden, sind sie definitiv für unser Projekt ausreichend sicher. Dafür sind diese Algorithmen natürlich etwas langsamer als ältere weniger starke Verschlüsselungssysteme, wie DES und SHA2. Die Datenbank wird mit PostgreSQL entwickelt. Dieses Programm ist opensource und unser Entwicklungsteam besitzt schon einige Erfahrungen damit. Dieses System ist zwar nicht wie MySQL mit fast allen Geräten kompatibel, aber dies spielt kaum eine Rolle, da die Datenbank nur mit der Website kommunizieren muss.

Genauere Wahl der Systeme ist aus der Nutzwertanalyse zu entnehmen.

5.2)   WIRTSCHAFTLICHE MACHBARKEIT

5.2.1 Personalaufwand:

Der Umfang der Software ist relativ groß, denn es müssen eine einfache Website und eine Datenbank erstellt werden. Außerdem muss ein Server konfiguriert und mit zahlreichen Verschlüsselungssystemen gesichert werden. Dies wird einiges an Durchführungszeit benötigen.

Die Gesamtstundenanzahl beträgt ungefähr 200 Personenstunden. Diese werden gleichmäßig auf alle vier Projektmitglieder aufgeteilt, um möglichst effizientes Arbeiten zu gewährleisten.

5.2.1 Investitionsaufwand:

Die Investitionsaufwände sind relativ gering, da der Großteil der verwendeten Softwarepakete opensource und damit kostenlos ist. Die einzigen Anschaffungskosten sind der Kauf des Adobe – Dreamweavers, der ungefähr 100 Euro kostet. Das Projekt wird in Wien in Räumlichkeiten der Flitzer AG mit Firmencomputern durchgeführt. Daher fallen hier keine zusätzlichen Kosten an.

5.2.3 Nutzen:

Das System wird ausschließlich der Firma Flitzer AG gehören. Die Website ist für alle Internetbenutzer zugänglich und das Bestellsystem kann von allen Personen mit Führerschein, die das Fahrzeug in Österreich abholen, in Anspruch genommen werden.

5.2.4 Risikoanalyse:

Bei diesem Projekt sind Risiken nicht auszuschließen. Ein Risikofaktor ist der krankheitsbedingte Ausfall eines Teammitglieds. In diesem Fall werden die anderen Teamitglieder den Teil des Abwesenden übernehmen. Dabei würde es höchstwahrscheinlich zu einer Verzögerung des Projektabschlusses kommen. Krankheiten sind absolut zufällig und können nicht wirklich verhindert werden, aber da das Projekt im Winter stattfindet wird natürlich für einen beheizten Arbeitsraum gesorgt.

Ein weiteres nicht vorhersehbares Problem ist ein Virus auf den Arbeitsgeräten. Dies kann verheerende Folgen für das Projekt haben. Deshalb werden die Daten täglich nach Arbeitsschluss extern und zusätzlich auf einer Cloud gesichert, um die Auswirkungen eines Virus möglichst gering zu halten. Sollten die aktuellen Daten verloren gehen, werden die zuletzt gesicherten wiederhergestellt und weiterverwendet.

Eine andere Projekterschwerung könnten Netzwerkprobleme verursachen. Es wird daher hauptsächlich (in den Bereichen, in denen es möglich ist) mit LAN anstatt WLAN gearbeitet. Sollten trotzdem Probleme auftreten werden, die LAN – Kabel ausgetauscht. Bei einem Netzwerkproblem, bei dem LAN keine Alternative darstellt, muss der Netzwerkadministrator das Problem beheben und das Projekt muss bis zu diesem Zeitpunkt angehalten werden.

Eine weitere Fehlerquelle können die Server darstellen. Deshalb werden zusätzliche Server zur Sicherheit reserviert, sodass im Eintrittsfall auf einen alternativen Server gewechselt werden kann. Sollten auch diese ausfallen, muss das Projekt bis zu Reparatur dieser durch den Serverinhaber pausiert werden. Dieser Fall ist aber sehr unwahrscheinlich

5.3) Persönliche Machbarkeit

Der schwierigste Teil, das Sichern des Systems durch kryptographische Algorithmen, stellt kein Problem dar, denn dafür sind in unserem Team drei absolute Experten zuständig. Das Wissen über die Webseiten- und Datenbankkreation ist auch vorhanden. Das Team hat zwar nicht sehr viele Erfahrungen in diesen zwei Bereichen, aber die Aufgabe ist nicht zu komplex, daher wird dies kein Hindernis darstellen.

5.4) Nutzwertanalyse

5.4.1 Schlüsselaustauschverfahren:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Schlüsselaustauschverfahren** | | RSA | | Diffie-Hellman | | ECC | |
| Kriterium | % | Rang | G \* R | Rang | G \* R | Rang | G \* R |
| Sicherheit | 20 | 1 | 20 | 1 | 20 | 1 | 20 |
| Performance | 10 | 2 | 20 | 2 | 20 | 1 | 10 |
| Schlüssellänge | 10 | 3 | 30 | 2 | 20 | 1 | 10 |
| Bewährtheit | 40 | 1 | 40 | 2 | 80 | 3 | 120 |
| Schlüsselmanagement | 20 | 1 | 20 | 3 | 60 | 1 | 20 |
| Gesamt | 100 | **1** | 130 | 3 | 200 | 2 | 180 |

5.4.2 Symmetrisches Verschlüsselungsverfahren:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Symmetrisches Kryptoverfahren** | | AES | | Twofish | | RC6 | |
| Kriterium | % | Rang | G \* R | Rang | G \* R | Rang | G \* R |
| Sicherheit | 30 | 1 | 30 | 1 | 30 | 1 | 30 |
| Performance | 30 | 1 | 30 | 3 | 90 | 2 | 60 |
| Bewährtheit | 40 | 1 | 40 | 2 | 80 | 2 | 80 |
| Gesamt | 100 | **1** | 70 | 3 | 200 | 2 | 170 |

5.4.3 Hashfunktion:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Hashfunktion** | | SHA-3 | | SHA-512 | | SHA-2 | |
| Kriterium | % | Rang | G \* R | Rang | G \* R | Rang | G \* R |
| Sicherheit | 20 | 1 | 20 | 1 | 20 | 1 | 20 |
| Performance | 30 | 1 | 30 | 3 | 90 | 2 | 60 |
| Userakzeptanz | 25 | 1 | 25 | 2 | 50 | 2 | 50 |
| Bewährtheit | 25 | 3 | 75 | 1 | 25 | 1 | 25 |
| Gesamt | 100 | **1** | 150 | 3 | 185 | 2 | 155 |

6)    Projektorganisation

Die Projektauftraggeber und Ansprechpartner diese Projekts sind Walter Rafeiner-Magor und Erhard List.

**Gary Ye:**

|  |  |
| --- | --- |
| *Position:* | Projektleiter |
| *Spezialgebiet:* | Algorithmen |
| *Aufgabenbereich:* | Programmieren und Konfigurieren der Kommunikation über das Internet, Implementieren der Datenspeicherung, Projektleitung |

**Elias Frantar**:

|  |  |
| --- | --- |
| *Position:* | Teammitglied |
| *Spezialgebiet:* | Kryptographie |
| *Aufgabenbereich:* | Planung und Implementierung der kryptografischen Systeme, Android Entwicklung |

**Wolfram Soyka**:

|  |  |
| --- | --- |
| *Position:* | Teammitglied |
| *Spezialgebiet:* | Android |
| *Aufgabenbereich:* | Android Entwicklung, Unterstützung in allen Aufgabenbereichen |

Erhard List

Walter Rafeiner-Magor

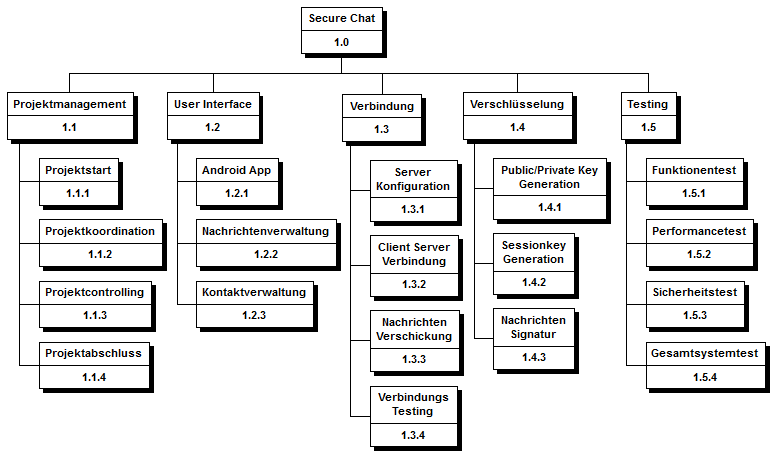
Gary Ye

Elias Frantar

Wolfram Sokya

7)    Projektplanung:

7.1) Projektstrukturplan



7.2) Meilensteinplan

|  |  |  |
| --- | --- | --- |
| **Meilenstein** | **Deliverable** | **Datum** |
| Start | Alle Hefte mit enthaltenen Informationen | 1.9.2013 |
| Android Skelett fertiggestellt | Eine App, die alle grundlegenden GUI-Elemente enthält, aber weder ein gutes Design noch Funktionen bereitstellt | Voraussichtlich am 7.10.2013 |
| Versenden von Nachrichten über das Internet funktioniert | Es können Textnachrichten zwischen zwei Geräten über das Internet (unverschlüsselt versendet werden) | Voraussichtlich am 21.10.2013 |
| Implementierung der Verschlüsselungssysteme abgeschlossen | Die Nachrichten können nun verschlüsselt versendet werden | Voraussichtlich am 15.11.2013 |
| Implementierung der Datenspeicherung abgeschlossen | Die ausgetauschten Nachrichten können nun abgespeichert und wider geladen werden. | Voraussichtlich am 6.12.2012 |
| Gesamtsystemtest abgeschlossen | Fertige App | Voraussichtlich am 15.12.2013 |

8)    Management Summary:

Die Firma Flitzer AG möchte ein Online – Bestellsystem, welches dem Kunden erlaubt 24 Stunden täglich Fahrzeuge zu mieten, einführen. Außerdem soll dieses System die Organisation automatisieren und die Papierorganisation komplett wegoptimieren.

Es soll also eine Website mit Online – Bestellsystem entwickelt werden, die eine Datenbank bei Bestellung automatisch aktualisiert. Es gibt zwar bereits sehr viele ähnliche Systeme, aber keines entspricht allen gewünschten Kriterien oder enthält alle Funktionen. Deshalb wird ein speziell abgestimmtes Produkt entwickelt.

Als Serverbetriebssystem wird Linux verwendet, da es erstens kostenlos ist und zweiten Adi Shamir, der die Serverkonfiguration übernehmen wird schon einige Erfahrungen damit hat.

Die Website wird mit Adobe Dreamweaver und PHP erstellt, da die Alternative Javascript in einigen Webbrowsern versehentlich deaktiviert werden kann. Außerdem werden alle Berechnungen bereits am Webserver durchgeführt, was schnelle Antwortzeiten auf Smartphones und weniger leistungsstarken PCs gewährleistet.

Die Umsetzung ist sowohl technisch, wirtschaftlich als auch persönlich definitiv machbar. Das Team und die Arbeitsmittel sind ausreichend. Etwaige Wissenslücken können sehr schnell und ohne großen Zeitverlust geschlossen werden. Das Know – How ist vor allem im komplexesten und schwierigsten Part, nämlich der Sicherheit des Systems, ausreichend vorhanden. Zur Verwirklichung des Projekts muss nur noch ein Programm um ungefähr 100 Euro angeschafft werden.

Das Bestellsystem wird (ausgenommen der Mitarbeiterkosten) ungefähr 100 Euro kosten. Außerdem muss noch die monatliche Miete der Website – Domäne von ungefähr 10 Euro beachtet werden.

Das komplette Projekt wird ca. 3, 5 Monate dauern. Genaue Termine sind in der Meilensteinplanung ersichtlich.

Alles in allem ist das Projekt definitiv machbar.

9)    Glossar:

* **Asymmetrische Verschlüsselung:** Ver- und Entschlüsselungen geschehen durch unterschiedliche Schlüssel (bei symmetrischen Verfahren durch denselben Schlüssel)
* **Public/Private Key-pair:** Ein asymmetrisches Verschlüsselungsverfahren benötigt immer einen öffentlichen Schlüssel zur Verschlüsselung und einen privaten Schlüssel zur Entschlüsselung von Nachrichten. Der öffentliche Schlüssel ist allen Sendern bekannt, der private aber nur dem Empfänger.
* **Session Key:** Jede Nachricht wird aus Performancegründen nur mit einem symmetrischen Sessionkey verschlüsselt, der anschließend asymmetrisch verschlüsselt wird.
* **Digitale Signatur:** Mithilfe von asymmetrischen Verschlüsselungsverfahren können Nachrichten digital signiert werden
* **Stromchiffre:** Eine Chiffrenart, die beliebig große Daten verschlüsseln kann. Dabei wird jedes Bit der Daten einzeln verschlüsselt. Ein Schlüssel kann nur einmal verwendet werden
* **Blockchiffre:** Eine Chiffrenart, die einen Datenblock bestimmter Größe verschlüsselt. Ein Schlüssel kann mehrmals verwendet werden.
* **Diffie-Hellman-Schlüsselaustausch:** Ein auf dem diskreten Logarithmus basierendes Verfahren zur geheimen Generierung eines gemeinsamen Schlüssel
* **RSA (Rivest Shamir Adelmann):** Ein asymmetrisches Verschlüsselungsverfahren dessen Sicherheit auf dem Faktorisierungsproblem, beruht mit einem privaten und einem öffentlichen Schlüssel
* **ECC (Elliptic Curve Cryptography):** Ein public-key-Kryptographieverfahren auf Basis elliptischer Kurven
* **AES (oder Rijndael):** Der Sieger des Blockchiffrenstandardisierungswettbewerbs für den *Advanced Encryption Standard*
* **SHA3 (Keccak):** Der Sieger des Einweghashfunktionsstandardisierungswettbewerbs für den *Secure Hash Algorithm 3*
* **SHA2, SHA512:** andere sichere kryptografische Einweghashfunktionen