

## ZIEL: ERWEITERTES USECASE DIAGRAMM UND SICHERHEITS-MATRIX

### USECASE DIAGRAMM

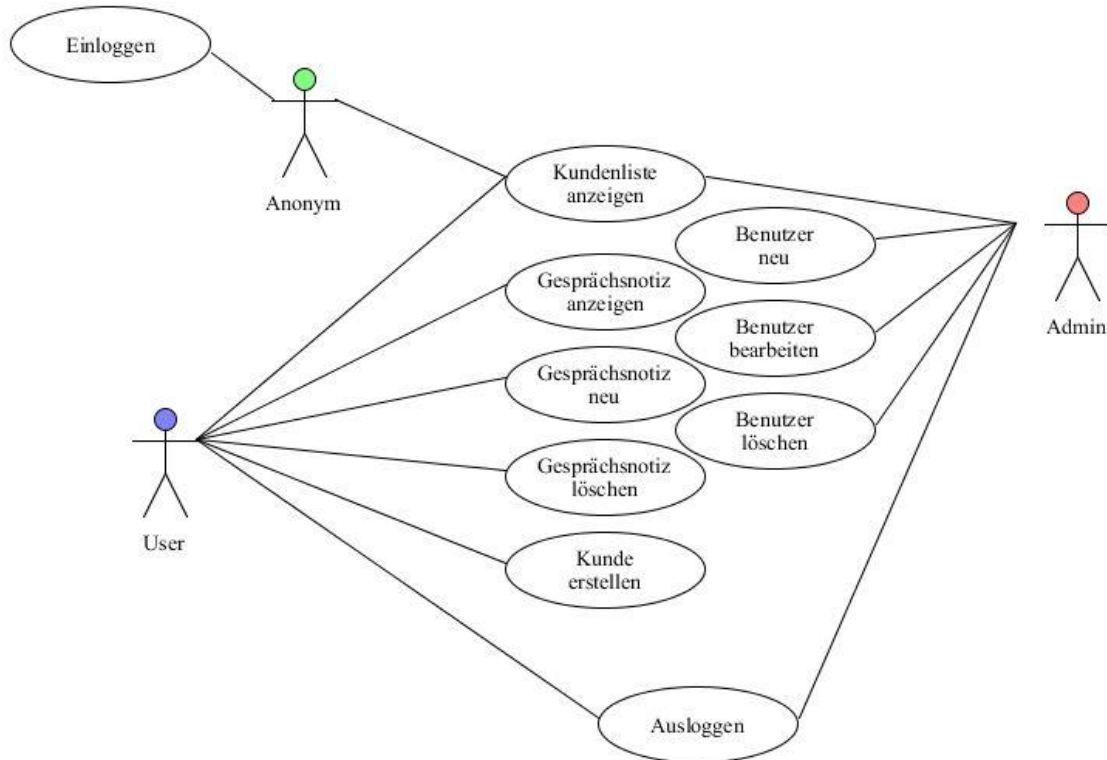


Abbildung 1 UseCase Diagramm Roles

### SICHERHEITSMATRIX

	Anonym	User	Admin
Einloggen	X		
Kundenliste anzeigen	X	X	X
Gesprächsnotiz erfassen		X	
Gesprächsnotiz anzeigen		X	
Gesprächsnotiz löschen		X	
Kunde erstellen		X	
Ausloggen		X	X
Benutzer neu			X
Benutzer bearbeiten			X
Benutzer löschen			X

Abbildung 2 Sicherheitsmatrix

## SECURITY

### AUTHENTICATION

Eine Authentifizierung ist ein Nachweis, dass man der ist, den man angibt zu sein.

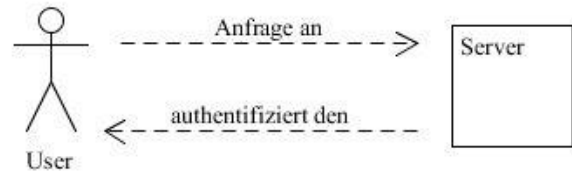


Abbildung 3 Authentifizierung

Der Ablauf einer Authentifizierung könnte bei einer Anmeldung via ssh so aussehen:

- Benutzer sendet eine Anfrage an den Server mit der Information zum «user»
- Der Server verlangt vom Benutzer ein Passwort zum «user»
- Der Server prüft ob der «user» mit dem eingegebenen Passwort übereinstimmen
- Stimmen die Daten überein, erteilt der Server dem Benutzer das Recht sich einzuloggen

### AUTHORIZATION

Die Autorisierung ist eine Zustimmung, die dem User bestimmte Rechte einräumt, nachdem dieser sich zuerst über eine Authentifizierung bekannt gemacht hat.

Die beiden häufigsten Fälle für die Autorisierung sind:

- den Zugriff auf Verzeichnisse und Dateien in einem Netzwerk
- die Erlaubnis zur Installation und Verwendung von Computerprogrammen

Im Normalfall, wird eine Autorisierung nicht erfolgen, wenn zuvor keine Authentifizierung stattgefunden hat.

### ENCRYPTION

Eine Verschlüsselung ist eine Methode um «Klartext» in «Geheimtext» unter Verwendung eines Schlüssels zu tätigen. Die Nachricht kann nur mit dem korrekten Schlüssel wieder in den Klartext zurückübersetzt werden.

Es gibt mehrere Verschlüsselungsarten:

- Symmetrisch
  - AES, DES, Blowfish
- Asymmetrisch
  - RSA
- Hybrid
  - PGP

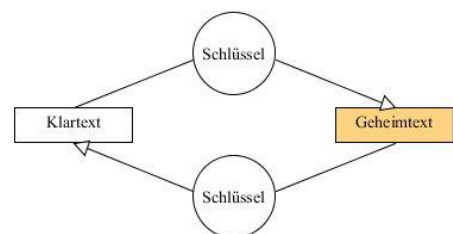


Abbildung 4 Symmetrische Verschlüsselung

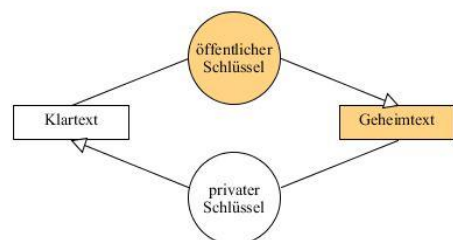


Abbildung 5 Asymmetrische Verschlüsselung

## SPRING SECURITY

Über Spring Security handeln wir die Authentifizierung und die Autorisierung der Sicherheit ab. Wir verwenden zusätzlich den BCryptPasswordEncoder um das Passwort zu verschlüsseln.

### «USERDETAILSSERVICEIMPL»

Die Klasse «UserDetailsServiceImpl» wird verwendet um die User zu authentifizieren und ihnen eine Rolle zuzuweisen.

Die Klasse enthält eine Unterklasse «UserDetailsImpl», welche mit der springframework.User klasse extended wird. Die Unterklasse kreiert im Konstruktor eine «AuthorithyList» für die Rollen, eine Abfrage für die ID und ein Set<> für die roleNames.

### «WEBSECURITYCONFIG»

In der Klasse «WebSecurityConfig» wird festgelegt, wer wo Zugriff erhält.

In unserer Klasse sind das für http:

- Alle ⇒ /, /customerList, /home, /public/\*, /rest/\*
- Admin ⇒ /admin/\*, /authenticatedUsers/\*
- User ⇒ /user/\*, /authenticatedUsers/\*
  
- Anonym ⇒ /login
  
- Admin, User ⇒ logout

In unserer Klasse wird folgendes das für web ignoriert:

- /resources/\*
- /static/\*
- /css/\*
- /js/\*
- /images/\*

### «ACCESSDENIEDHANDLERIMPL»

Der «AccessDeniedHandlerImpl» sorgt dafür, dass wenn sich einer mit falschen Informationen anzumelden versucht, auf eine «403»-Access Denied Seite weitergeleitet wird.

## **QUELLENVERZEICHNIS**

### **AUTHENTICATION**

<https://de.wikipedia.org/wiki/Authentifizierung>

### **AUTHORIZATION**

<https://de.wikipedia.org/wiki/Autorisierung>

### **ENCRYPTION**

<https://de.wikipedia.org/wiki/Verschlüsselung>