

Der Pate

Michel Blank, Lukas Feller,
Mohammad-Sharif-Moradi, Lewis Stoll

Herr Eick
Werner-Von-Siemens-Schule
60327 Frankfurt am Main

Inhaltsverzeichnis

1 Einleitung	3
1.1 Was ist „Der Pate“?	3
1.2 Weitere relevante Informationsquellen	3
2 Voraussetzungen	3
2.1 Entwicklung	3
3 Komponenten Zusammenhänge	4
3.1 Enum	4
3.2 Interfaces	5
3.3 Exceptions	6
3.4 Schnittstellen an das Backend	6
4 Interne Struktur	7
4.1 Grundaufbau	7
4.2 Admin-Web-Oberfläche	8
4.3 Godfather-Web-Oberfläche	9
4.4 Trainee-Web-Oberfläche	10
5 Technische Aspekte	13
5.1 Konfiguration von MariaDB	13
5.2 Konfiguration vom Git-Zugang	14
5.3 Konfiguration Entwicklungsumgebung	14
5.4 Technische Aspekte der Webseite	16
6 Implementation von Subkomponenten	17
6.1 Hibernate	17

6.2 Bootstrap	17
6.3 JQuery	18
7 Vergleich mit dem Pflichtenheft	18
7.1 Musskriterien	18
7.2 Wunschkriterien	19
8 Weitere Informationen	19
8.1 Verantwortlichkeiten	19
8.2 Besondere Probleme und Lösungen	20
8.3 Steuerung des Prozesses	20

1 Einleitung

Dieser Teil beschreibt das Zielpublikum des Dokuments, den Zweck des Dokuments und eine Liste von weiterführenden Dokumenten.

1.1 Was ist „Der Pate“?

Der Pate ist ein Online-Tool für zukünftige Nachwuchskräfte der DB Systel GmbH, das es diesen ermöglicht, einen symphytischen „Paten“ zu finden.

Bereits eingestellte Nachwuchskräfte können sich dann als Pate auf der Seite anmelden und einstellen, welche Informationen die neuen Nachwuchskräfte sehen können sollen.

Ein Administrator erstellt die dazugehörigen Accounts. Eine zukünftige Nachwuchskraft meldet sich mit einem Token an, der zufällig gewählt wurde, eine variable Länge hat und ihm per Post gesendet wurde. Paten und Administratoren melden sich mit ihrer „@deutschebahn.com“ E-Mail-Adresse und einem Passwort an.

1.2 Weitere relevante Informationsquellen

- Das aus dem Projekt zu generierende JavaDoc

2 Voraussetzungen

2.1 Entwicklung

Folgende Software wird für die Entwicklung verwendet:

- [JDK 11](#) (Java Development Kit)
- MariaDB Server (SQL Server)
- Tomcat 9 (Web Server)
- Eclipse JEE IDE (Version 2018-09 (4.9.0))
- GitHub
 - <https://github.com/michelblk/derPate>
- [Fork](#)
 - Ein Git-Client, der Git-Banches übersichtlicher als TortoiseGit darstellt

3 Komponenten Zusammenhänge

Die Anwendung wird mittels einer Java EE Umgebung umgesetzt. Es wird angestrebt, dass das Frontend nicht direkt vom Backend abhängt (JSP Anweisung an das Backend), sondern dies mittels REST-Anfragen zu gestalten.

Da die Implementierung von REST zu einigen unerwarteten Ergebnissen führte, haben wir uns zunächst entschieden, REST nicht zu verwenden, bis wir dann eine Lösung für die Authentifizierung gefunden haben. Solange werden Servlets und JSP verwendet.

Hinweis: Weitere Informationen zu den Zusammenhängen finden Sie im JavaDoc.

3.1 Enum

3.1.1 ServletParameter

In diesem Servlet werden die Keys für die Übergabeparameter aus dem Frontend an das Backend festgehalten. Sie haben folgenden Grundaufbau <Benutzergruppe>_<Feld>, da in den Formularen meist Daten eines Nutzers bearbeitet werden und dies Felder für eine ganze Benutzergruppe gelten.

Sollten Werte übertragen werden müssen, die nicht auf Benutzergruppen zurückzuführen sind, sollte überlegt werden, dennoch den Parameter in diesem Enum zu definieren. Ausnahmefälle können sein: Ein Util übernimmt die Definition des Keys und Values oder der Wert wird nur in einem Servlet benötigt (dann kann der Parameter in dem Servlet selbst als public static final String definiert werden).

Neben dem Namen wird hier außerdem auch definiert, wie diese Parameter auf ihre Gültigkeit überprüft werden können (siehe [InputValidator](#)).

3.1.2 CSRFForm

Hier werden alle Formulare definiert, die Daten verändern können. Als Schutz vor [Cross-Site-Request-Forgery](#) werden hier als zusätzliche Maßnahme die zufällig generierten, dem Client und Server gemeinsam bekannten, Token für nur ein Formular gültig gemacht.

In der Zukunft ist zu überprüfen, ob dies sinnvoll ist oder es reicht diesen Token mit der Session zu speichern. [Tomcat bringt hier gegebenenfalls schon Features mit.](#)

3.1.3 Usermode

Hier werden die drei Benutzergruppen definiert, damit darüber später zum Beispiel Berechtigungen für eine Seite festgelegt werden können. Momentan enthält sie je einen Eintrag für die Benutzergruppen Trainee (Nachwuchskraft), Godfather (Pate), Admin (Administrator).

3.2 Interfaces

3.2.1 HasName

HasName wird von den Model-Klassen verwendet um anzuzeigen, dass das Model einen Namen zurückgeben kann. Dies wird für Entitäten verwendet, die eine id mit einem Namen zusammenfügen (beispielsweise einen Städtenamen).

3.2.2 Dao

Dao gibt vor, dass ein Data Access Object die Methoden zum Laden, Ändern, Hinzufügen und Löschen von Datensätzen implementieren muss.

3.2.3 InputValidator

InputValidator werden verwendet, um HTTP Parameter in Servlets auf ihre Gültigkeit zu prüfen. InputValidator'en müssen die Methode „isValid(String)“ implementieren und soll dann diesen String auf die Gültigkeit für den Parameter überprüfen (Beispiel IdValidator: Existiert diese Id in der Datenbank?).

3.2.4 ServletFilter

ServletFilter sorgen dafür, dass ein Servlet oder andere erreichbare Seiten nur von bestimmten Benutzergruppen oder unter bestimmten Voraussetzungen erreicht werden können (Beispiel: Nur angemeldete Nutzer).

3.2.5 EncryptionUtil

Ein EncryptionUtil (Util zur Verschlüsselung von Daten) soll eine Methode „encrypt(String)“ und „decrypt(String)“ enthalten.

3.2.6 HashUtil

Ein HashUtil (Util zum Hashen von Daten) soll die Methoden „hash(String, byte[], byte[], String)“, „hash(String, int, byte[], String)“ und „isEqual(String, String, byte[])“ enthalten.

3.3 Exceptions

Es gibt keine bestimmten Komponenten-Exceptions.

3.4 Schnittstellen an das Backend

Die Schnittstellen wurden mittels Java Servlets realisiert. Es existieren folgende Schnittstellen.

Schnittstelle	Pfad	Für Benutzergruppe
Nachwuchskraft Token generieren	/adminGenerateTraineeToken	Admin
Nachwuchskraft Token entfernen	/adminTraineeUpdate	Admin
Pate hinzufügen	/adminGodfatherAdd	Admin
Pate aktualisieren	/adminGodfatherUpdate	Admin
Pate Bild hochladen	/uploadGodfatherImage	Pate
Pate aktualisieren	/godfatherUpdate	Pate
Paten auswählen	/godfatherSelect	Nachwuchskraft
Passwort ändern	/changepassword	Pate, Admin
Pate suchen	/findGodfather	Nachwuchskraft, Admin
Pate Bild abrufen	/godfatherImage	Angemeldete Nutzer
Anmelden	/login	Nicht angemeldete Nutzer
Abmelden	/logout	Angemeldete Nutzer
Weiterleitung zur Startseite für die Benutzergruppe. Abgemeldete Nutzer werden zum Login weitergeleitet	/redirect	Alle, auch abgemeldete Nutzer

Hat ein Nutzer nicht die Berechtigung auf eine Seite zuzugreifen oder hat sie ungültige Werte mitgegeben, so erscheint eine Fehlerseite.

Beispiel für einen Fehler bei fehlender Berechtigung:



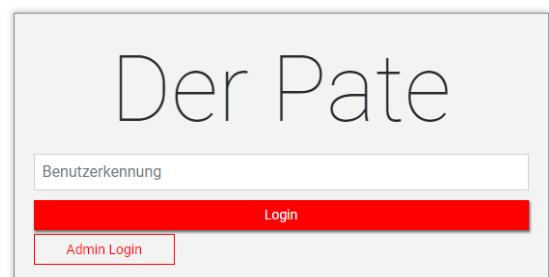
4 Interne Struktur

Die Namen der Ordner sind grundsätzlich Englisch und sollten auch weiterhin in Englisch gehalten werden.

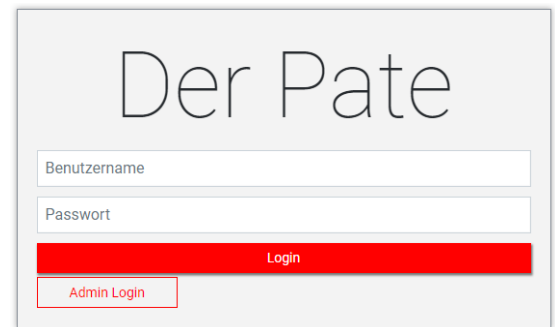
4.1 Grundaufbau

4.1.1 Loginseite

Auf dem Anmeldebildschirm kann sich die zukünftige Nachwuchskraft mit dem ihm zuvor mitgeteiltem Token anmelden.



Außerdem hat der Administrator und der Pate die Möglichkeit, sich mit der E-Mail-Adresse und dem Passwort anzumelden.



Der Pate

Benutzername

Passwort

Login

Admin Login

4.1.2 Header

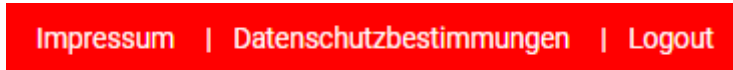
Ein Header soll den Nachwuchskräften, Paten und Administratoren zeigen, dass sie sich noch auf der Seite des Projektes befinden.

Dieser Header wird auf jeder Seite angezeigt und enthält das DB-Logo auf der rechten Seite. Zusätzlich kann er Menüpunkte enthalten, falls eine Navigation stattfinden kann.



4.1.3 Footer

Ein auf jeder Seite eingebundener Footer gibt dem Nutzer via Links die Möglichkeit, auf das Impressum, Datenschutzbestimmungen und den Logout, sofern der Benutzer angemeldet ist, zuzugreifen.




4.2 Admin-Web-Oberfläche

Der Admin soll die Möglichkeit haben, die Patenaccounts sowie die Token für die Nachwuchskräfte zu bearbeiten.

4.2.1 Patenverwaltung

Der Administrator hat die Möglichkeit, die bestehenden Accounts der Paten zu bearbeiten, zu entfernen sowie einen solchen anzulegen.

Patenverwaltung Nachwuchskräfteverwaltung
DB



Vorname:


Nachname:

Geburtstag:

Email:

Standort:

Einstellung
sdatum:



Vorname:


Nachname:

Geburtstag:

Email:

Standort:

Einstellung
sdatum:



Vorname:

Nachname:

Geburtstag:

Email:

Standort:

Einstellung
sdatum:

4.2.2 Nachwuchskräfteverwaltung

Patenverwaltung Nachwuchskräfteverwaltung
DB

Token generieren

Token	Pate	Aktionen
Trainee1	Erika Mustermann	↺ ✖
Trainee2	Erik Mustermann	↺ ✖
Trainee3	Max Mustermann	↺ ✖
Trainee4	- nicht gewählt -	↺ ✖
Trainee5	- nicht gewählt -	↺ ✖

Impressum | Datenschutzbestimmungen | Logout
H

hier können Token für die Nachwuchskräfte erstellt werden, bei bestehenden kann die Verknüpfung zum Pate gelöscht werden und die Token können gelöscht werden. Außerdem sieht man, welche Nachwuchskraft welchen Pate gewählt hat.

4.3 Godfather-Web-Oberfläche

Meldet man sich als Pate an, hat man die Möglichkeit, seine Eigenen Daten zu ändern.

4.3.1 Daten ändern

Profil
DB

Vorname	Max	Nachname	Mustermann
Geburtstag	1998-01-20	Email	<input type="text" value="max.mustermann@deutschebahn.com"/>
Standort	<input type="text" value="Frankfurt am Main"/>	Einstellungsdatum	2016-08-15
Job	Fachinformatiker Anwendungsentwicklung (Ausbildung)	Ausbildungsjahr	3
Max. Anzahl Nachwuchskräfte	<input type="text" value="1"/>	Akt. Anzahl Nachwuchskräfte	0
Kurze Beschreibung	<input type="text" value="Hallo, ich bin Max und beantworte gern Deine Fragen rund um das Thema Duales Studium bei der Deutschen Bahn."/>		
Neues Passwort	<input type="text"/>	Password wiederholen	<input type="text"/>

Als Pate kann man die Einstellungen „E-Mail“, „Standort“, die Anzahl maximaler Nachwuchskräfte sowie die Beschreibungen verändern. Die Einstellungen „Vorname“, „Nachname“, „Geburtstag“, „Einstellungsdatum“ (daraus ergebend das „Ausbildungsjahr“) sowie die aktuelle Anzahl an Nachwuchskräften kann nur der Administrator verändern.

4.4 Trainee-Web-Oberfläche

Meldet man sich als Nachwuchskraft an, kann man sich einen Paten aussuchen, um auftretende Fragen zu klären.

4.4.1 Willkommensseite

Paten
DB

Willkommen!

Der Pate ist eine interne Applikation um dich bei anfänglichen Fragen und Schwierigkeiten zu unterstützen. Hier kannst du selbst einen sogenannten Paten auswählen mit welchem du über mehrere Möglichkeiten kommunizieren kannst. Dein erster Schritt ist es einen Paten auszuwählen. Das erfolgt auf der nächsten Seite. Hier ist oben am Bildschirmrand eine Menüleiste aufzufinden. Über dieses Menü kannst du auswählen aus welchen Kategorien du einen Paten brauchst. Hier sind Dinge wie Ausbildungsjahr, Wohnort und Ausbildungsberuf auswählbar. Nach dem drücken auf "Suchen", werden dir deine passenden Paten angezeigt. Über den Knopf "Mehr anzeigen" gelangst du auf die ausführlichere Ansicht der Person. Falls dir diese Person als Paten zusagt kannst du ihn über den Knopf "Als Paten auswählen" ihn oder Sie auswählen. Falls nicht kommst du über den Knopf schließen wieder zu der Auswahl zurück.

Weiter


Impressum
Datenschutzbestimmungen
Logout

So bald man sich mit dem Token angemeldet hat, kommt man auf diese Willkommens-Seite. Hier ist kurz beschrieben, was die Nachwuchskraft hier machen kann. Sowohl über den Button „Weiter“ als auch über den Button Paten, wird man auf die nächste Seite weitergeleitet.

4.4.2 Patenauswahl

Willkommen
Paten wählen
DB

Standort ▾
Ausbildungsform ▾
Beruf ▾
Lehrjahr ▾
Suchen



☒ Ausbildung
☐ Duales Studium

Erika


Frankfurt am Main

Ausbildung

Fachinformatiker

Anwendungsentwicklung

Mehr anzeigen



Erik

Berlin

Ausbildung

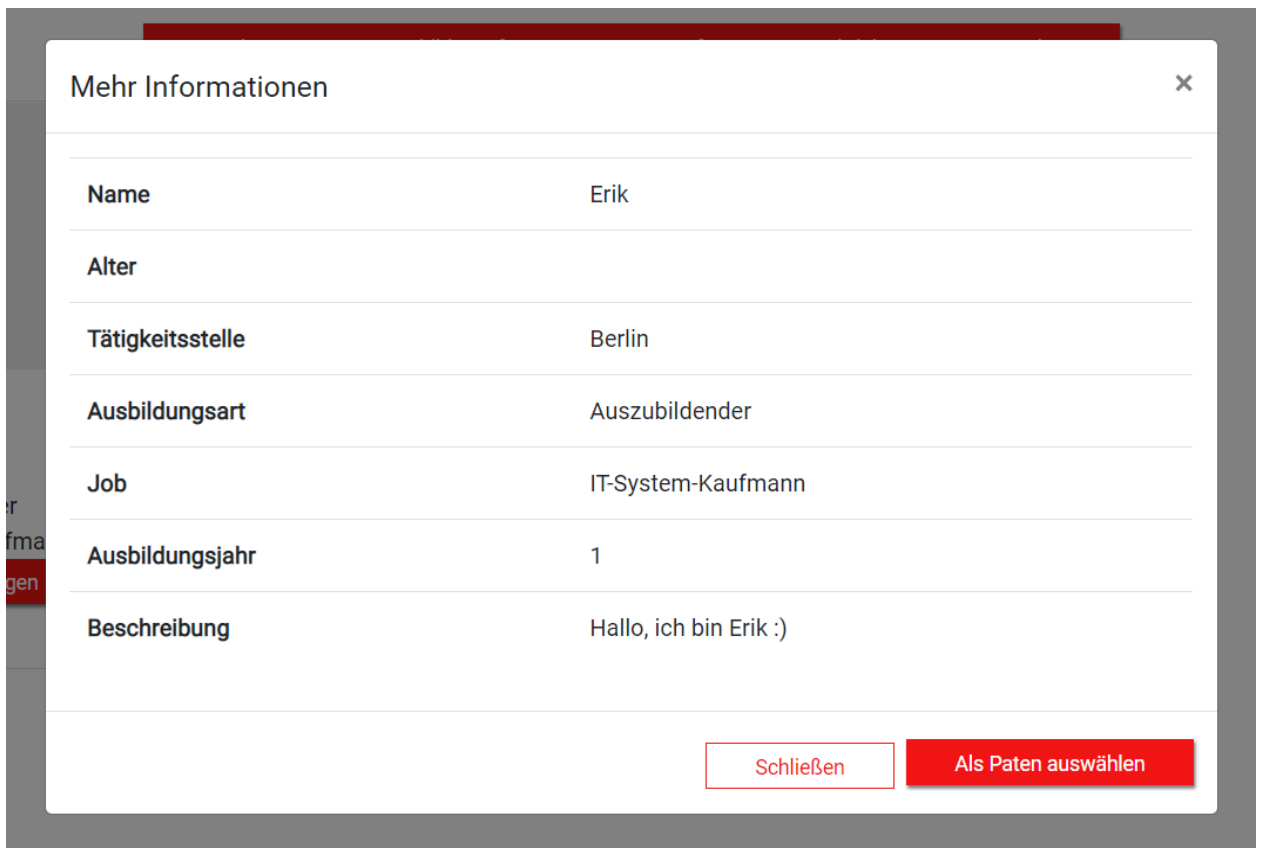
IT-System-Kaufmann

Mehr anzeigen

[Impressum](#) | [Datenschutzbestimmungen](#) | [Logout](#)

Hier kann nun über den Filter nach Standort, Ausbildungsform, Beruf und Lehrjahr gefiltert werden. Es werden nur die Paten angezeigt, die verfügbar sind, d.h. die noch nicht die maximale Anzahl an Nachwuchskräften haben. Über den Button „Mehr anzeigen“ lässt sich ein Modal öffnen, in dem mehr Informationen über den Paten stehen.

4.4.3 Mehr Informationen



The screenshot shows a modal window titled 'Mehr Informationen' with a close button (X) in the top right corner. The modal contains a table with the following data:


Name	Erik
Alter	
Tätigkeitsstelle	Berlin
Ausbildungsart	Auszubildender
Job	IT-System-Kaufmann
Ausbildungsjahr	1
Beschreibung	Hallo, ich bin Erik :)

At the bottom right of the modal, there are two buttons: 'Schließen' (outlined) and 'Als Paten auswählen' (solid red).

Hier stehen nun alle Informationen die der Pate von sich angegeben hat. Die verschiedenen Informationen sind: Name, Alter, Tätigkeitsstelle, Ausbildungsart, Job, Ausbildungsjahr und eine kurze Beschreibung. Nun habe ich die Möglichkeit den Paten zu wählen, oder das Modal wieder zu schließen. Wenn man den Button „Als Paten auswählen“ drückt, wird man auf eine neue Seite weiter geleitet und erhält dort noch einmal mehr Informationen.

4.4.4 Der gewählte Pate

DB



Erik Mustermann
Ausbildung
IT-System-Kaufmann
Berlin
1. Lehrjahr

Hallo, ich bin Erik :) Ich freue mich, dass Du Dich für mich entschieden hast. Klicke unten auf den Button, um mich zu kontaktieren :)

Klicke hier, um mir eine E-Mail zu schreiben

[Impressum](#) | [Datenschutzbestimmungen](#) | [Logout](#)

Wenn man einen Paten ausgewählt hat, gelangt man auf diese Seite. Hier erhält man zusätzliche Informationen. Wenn man auf „Klicke hier, um mir eine E-Mail zu schreiben“ klickt, öffnet sich Outlook und man kann eine E-Mail an den Paten schreiben. Sobald man sich den Paten ausgesucht hat, kann die Nachwuchskraft nicht mehr zurück zur Übersicht und sich nach anderen Paten umschauen.

5 Technische Aspekte

5.1 Konfiguration von MariaDB

Damit die Datenbank mit der Anwendung „Der Pate“ verwendet werden kann, ist es notwendig die Datenbankstruktur zu importieren. Dazu erstellt man eine Datenbank im Datenbank-Management-System und importiert die Datei [Datenbank.sql](#). Um sich mit Beispieldaten anmelden zu können, müssen außerdem die Beispieldaten eingefügt werden. Am einfachsten geht das, indem man die Datei [Beispieldaten.sql](#) importiert. Wichtig dabei ist, die Fremdschlüsselüberprüfung zu deaktivieren.

Danach kann man sich mit folgenden Daten anmelden:

Benutzergruppe	Anmeldename	Passwort
Nachwuchskraft	Trainee1	

Nachwuchskraft	Trainee2	
Nachwuchskraft	Trainee3	
Nachwuchskraft	Trainee4	
Nachwuchskraft	Trainee5	
Pate	Max.Mustermann@deutschebahn.com	test
Pate	Erika.Mustermann@deutschebahn.com	test
Pate	Erik.Mustermann@deutschebahn.com	test
Admin	Admin@deutschebahn.com	test

5.2 Konfiguration vom Git-Zugang

Um lesenden Zugriff auf das Git-Repository <https://github.com/michelblk/derPate/> zu erhalten, ist keine Freigabe nötig. Um allerdings Änderungen hochzuladen, muss durch ein bestehendes Teammitglied die Freigabe unter <https://github.com/michelblk/derPate/settings/collaboration> erteilt werden.

5.3 Konfiguration Entwicklungsumgebung

Um mit der Entwicklungsumgebung Eclipse an dem Projekt weiterentwickeln zu können, muss das Projekt in Eclipse importiert werden. Dazu wählt man unter „Files“ den Punkt „Import...“ aus und wählt schließlich „Import existing Maven Project“. Nun muss lediglich noch der Projektpfad angegeben werden (/Workspace/DerPate).

5.3.1 secret.properties

Um das Projekt nun erstmalig einzurichten, muss der MariaDB Server (inklusive Beispieldaten) gestartet und eine Internetverbindung muss vorhanden sein. Die Datei **secret.properties, die nicht in Git versioniert ist**, müssen die Zugangsdaten für die Datenbank festgelegt werden. Diese muss in dem Ordner „Workspace/DerPate/src/main/resources“ abgelegt werden.

Beispiel:

```

1 # DO NOT UPLOAD THIS FILE TO A PUBLIC INSTANCE, e.g. github!
2
3 # Database connection
4 database.url           = jdbc:mysql://localhost/derpate
5 database.username      = root
6 database.password
7
8 # Pepper used for hashing
9 encryption.pepper      = tzcgvsd783wer!hzasdnkjvahjbd*'asdseaiwerjfewjk4:
10
11 #AES256 encryption
12 encryption.aes256.password = ^CvW+t^6Abmn8LQJvs=p+Q3BHmRa5e8Dm7dTY3EbMEbu%nZ{
13 encryption.aes256.salt    = deab233028c12aea09590aa8b7ba7128
14

```

In dem genannten Ordner befinden sich außerdem zwei weitere relevante Dateien, die gegebenenfalls Änderungen bedürfen: `app.properties` und `security.properties`

5.3.2 app.properties

Hier wird der Zeichensatz, der Ort für die Speicherung von Bildern und der maximale Wert für Paten festgelegt, wie viele Nachwuchskräfte sie maximal betreuen dürfen.

```

1 app.charset           = UTF-8
2
3 # data folder used for file storage
4 app.data              = D:/derpate/data
5
6 # godfather settings
7 godfather.max_trainees = 10
8

```

5.3.3 security.properties

Hier werden sicherheitsrelevante Einstellungen vorgenommen: Die Länge des zu generierenden Salts für die Verschlüsselung/Hashbildung, der Trenner zwischen Salt und Hash sowie die Anzahl in Sekunden, die ein Nutzer maximal inaktiv sein darf.

Die ersten beiden Einstellungen betreffen vor allem die Speicherung in der Datenbank.

```

1 # Number of bytes used for generating a random salt
2 encryption.salt_length = 64
3 #Character used to separate salt from encryption
4 encryption.separator   = .
5
6 # Time in seconds, that the user can be inactive, without getting his session logged out
7 login.timeout          = 600
8

```


5.4 Technische Aspekte der Webseite

5.4.1 Definition der Servletpfade

Die Pfade für die oben genannten Servlets werden im Servlet selbst mittels der „@WebServlet“-Annotation definiert:

```

37  *
38  */
39  @WebServlet("/uploadGodfatherImage")
40  @MultipartConfig
41  public class GodfatherUpdateImageServlet extends FilterServlet {
42  /**

```

5.4.2 Definition der Filter

5.4.2.1 Servlet

Filter für Servlets werden im zu schützenden Servlet selbst definiert. Das Grundgerüst dafür sieht wie folgt aus:

```

4
5  public class ExampleServlet extends FilterServlet {
6  public ExampleServlet() {
7      super();
8  }
9  }

```

Um nun ein Servlet zu schützen, kann der Konstruktor der Superklasse mit den Filtern befüllt werden. Im folgenden Beispiel wird das Servlet nur für Nachwuchskräfte erreichbar gemacht:

```

3  import de.db.derpate.Usermode;
4  import de.db.derpate.servlet.FilterServlet;
5  import de.db.derpate.servlet.filter.LoginServletFilter;
6
7  public class ExampleServlet extends FilterServlet {
8  public ExampleServlet() {
9      super(new LoginServletFilter(Usermode.TRAINEE));
10 }
11 }

```

5.4.2.2 Ordner und Dateien

Um Filter auf ganze Ordner oder Dateien anzuwenden, müssen sie in der „/Workspace/DerPate/src/main/webapp/WEB-INF/web.xml“ definiert werden.

Im nebenstehenden Beispiel ist zu sehen, dass der Ordner „/admin/“ nur von Admins aufgerufen werden darf, der Ordner „/trainee/“ nur von zukünftigen Nachwuchskräften und „/godfather/“ nur von Paten.

Es ist zu beachten, dass es

```

21 <!-- Filter -->
22 <filter>
23   <filter-name>TraineeOnly</filter-name>
24   <filter-class>de.db.derpate.servlet.filter.LoginServletFilter</filter-class>
25   <init-param>
26     <param-name>usermode</param-name>
27     <param-value>TRAINEE</param-value>
28   </init-param>
29 </filter>
30 <filter>
31   <filter-name>GodfatherOnly</filter-name>
32   <filter-class>de.db.derpate.servlet.filter.LoginServletFilter</filter-class>
33   <init-param>
34     <param-name>usermode</param-name>
35     <param-value>GODFATHER</param-value>
36   </init-param>
37 </filter>
38 <filter>
39   <filter-name>AdminOnly</filter-name>
40   <filter-class>de.db.derpate.servlet.filter.LoginServletFilter</filter-class>
41   <init-param>
42     <param-name>usermode</param-name>
43     <param-value>ADMIN</param-value>
44   </init-param>
45 </filter>
46
47 <filter-mapping>
48   <filter-name>AdminOnly</filter-name>
49   <url-pattern>/admin/*</url-pattern>
50 </filter-mapping>
51 <filter-mapping>
52   <filter-name>TraineeOnly</filter-name>
53   <url-pattern>/trainee/*</url-pattern>
54 </filter-mapping>
55 <filter-mapping>
56   <filter-name>GodfatherOnly</filter-name>
57   <url-pattern>/godfather/*</url-pattern>
58 </filter-mapping>

```

aufgrund eines Bugs von Tomcat dazu kommen kann, dass nicht nur die genannten Ordner, sondern auch jede Datei im Stammverzeichnis, die mit dem Namen „admin“, „trainee“ beziehungsweise „godfather“ beginnt, geschützt wird – darunter zählen auch Servletpfade.

6 Implementation von Subkomponenten

6.1 Hibernate

Hibernate ist ein Persistenz-Framework. Dieses generiert aus Java Objekten entsprechende Datenbanktabellen und kommuniziert mit dieser über SQL Befehlen.

Es verwendet die Java Persistence API, um die Daten zwischenspeichern, um so die Anzahl der Datenbankabfragen zu minimieren.

6.2 Bootstrap

Bootstrap ist ein frontend CSS Framework, welches auf CSS und JavaScript basiert.

Es beinhaltet viele Gestaltungsvorlagen für Typografie, Formulare, Buttons, Tabellen, Grid-Systeme, etc.

Bootstrap wird unter der [MIT Lizenz](#) veröffentlicht, weshalb die Lizenz- und Copyright Abschnitte in Bootstrap's CSS und JavaScript Dateien erhalten bleiben müssen.

Auch wenn es erlaubt ist, diese zu bearbeiten, empfiehlt es sich, die zu verändernden Eigenschaften in einer neuen CSS Datei zu überschreiben, damit es einfacher ist, Bootstrap auf eine neue Version zu heben.

6.3 JQuery

jQuery ist eine JavaScript-Bibliothek, die Webentwicklern die Arbeit mit JavaScript erleichtern soll. Sie überbrückt technische Unterschiede zwischen verschiedenen Browsern und ermöglicht komfortablen und kompakten Code, der im direkten Vergleich zu JavaScript häufig sehr viel kürzer ist.

7 Vergleich mit dem Pflichtenheft

7.1 Musskriterien

- Es gibt zwei Rollen und einen Admin-Account

Dieses Musskriterium haben wir erfüllt, siehe 4.1.1.

- Die Nachwuchskräfte können sich mittels Zugangscodes, die Ansprechpartner mit Benutzernamen und Passwort anmelden. Sie haben unterschiedliche Rollen und Benutzeroberflächen

Dieses Musskriterium haben wir erfüllt, siehe 4.3. – 4.4.

- Nachwuchskräfte sehen einige der von Ansprechpartnern zur Verfügung gestellten Informationen

Dieses Musskriterium haben wir mit einem Modal umgesetzt, siehe 4.4.3.

- Nachwuchskräfte können einen Ansprechpartner auswählen und sehen dann alle verfügbaren Daten zu dieser Person, Jedoch lässt sich dann nicht mehr zur Übersicht zurückkehren und die Nachwuchskraft ist an diesen Ansprechpartner gebunden

Dieses Musskriterium haben wir umgesetzt, siehe 4.4.4.

- Ansprechpartner können ihre persönlichen Daten ändern.

Dieses Musskriterium haben wir umgesetzt, siehe 4.3.1.

- Der Admin kann weiter Zugangscodes für Nachwuchskräfte erstellen, die für das jeweilige Ausbildungsjahr gültig sind

Dieses Musskriterium haben wir umgesetzt, siehe 4.2.2.

- Admin können weiter Ansprechpartner-Accounts erstellen und löschen

7.2 Wunschkriterien

- Ansprechpartner müssen bei Erstanmeldung ihr Passwort ändern können danach beliebig oft freiwillig ändern

Dieses Wunschkriterium wurde teilweise umgesetzt. Der Pate kann so oft sein Passwort ändern, allerdings nicht bei der ersten Anmeldung. Siehe 4.3.1.

- Ansprechpartner sollen einstellen können, wie viele Nachwuchskräfte sie aufnehmen wollen und können einsehen, wie viele Nachwuchskräfte sich bereits für sie entschieden haben.

Dieses Wunschkriterium wurde umgesetzt, siehe 4.3.2.

- Nachwuchskräfte können in der Übersicht der Ansprechpartner nach Ausbildungsjahr, Beruf und Standort gruppieren.

Dieses Wunschkriterium wurde umgesetzt, siehe 4.4.2.

8 Weitere Informationen

8.1 Verantwortlichkeiten

Wir haben uns dazu entschieden, die Verantwortlichkeiten nach Programmierschicht zu verteilen:

Schicht	Verantwortliche/-r
Frontend (HTML, CSS, JavaScript, JSP)	Lukas Feller, Lewis Stoll, Mohammad-Sharif Moradi
Backend (Java)	Michel Blank, Lukas Feller

Datenbank (Entwurf + SQL)	Lukas Feller
---------------------------	--------------

8.2 Besondere Probleme und Lösungen

8.2.1 Cross-Site-Request-Forgery

Wir haben uns dazu entschieden, unsere Anwendung vor Cross-Site-Request-Forgery zu schützen. Wie bereits [oben](#) beschrieben, wird für jedes Formular und Sitzung ein zufällig generierter Code erzeugt, der dem Client mitgeteilt wird. Wird dann ein Formular an das Backend abgesendet, zeigt dieser Code an, dass die Anfrage auch wirklich von unserer Anwendung ausgelöst wurde und nicht durch ein Formular auf einer anderen Seite, das auf unsere Seite weiterleitet.

Beispiel: Ein Link wird mit einer E-Mail verschickt. Dieser Link zeigt unter einem anderen Titel auf den Logout unserer Anwendung. Würde ein Anwender diesen Link anklicken und in unserer Anwendung angemeldet sein, würde er gegen seinen Willen abgemeldet werden. Mit CSRF-Schutz ist dies nicht mehr möglich, da der Token von derselben Sitzung generiert werden müsste, die der Absender der E-Mail nicht kennt.

8.3 Steuerung des Prozesses

Wir haben uns zu Projektbeginn dazu entschieden, unseren Fortschritt in einem Kanban-Board festzuhalten. Da wir Kosten sparen wollten, haben wir uns für die kostenfreie Software Trello entschieden. Dort dokumentieren wir die Aufgaben und zukünftige Verbesserungsmöglichkeiten und verfolgen so, wie wir in der Zeit liegen.

Wir haben unser Kanban-Board in sieben Abschnitte unterteilt: ToDo, Im Gange, Test, Ready for Development, Development, Ready for Review, Review und Fertig.

Features, die wir als wichtig erachtet haben, haben wir mit einem Fälligkeitsdatum versehen, solche, die später hinzugefügt werden können, haben wir mit der Markierung „Backlog“ versehen.