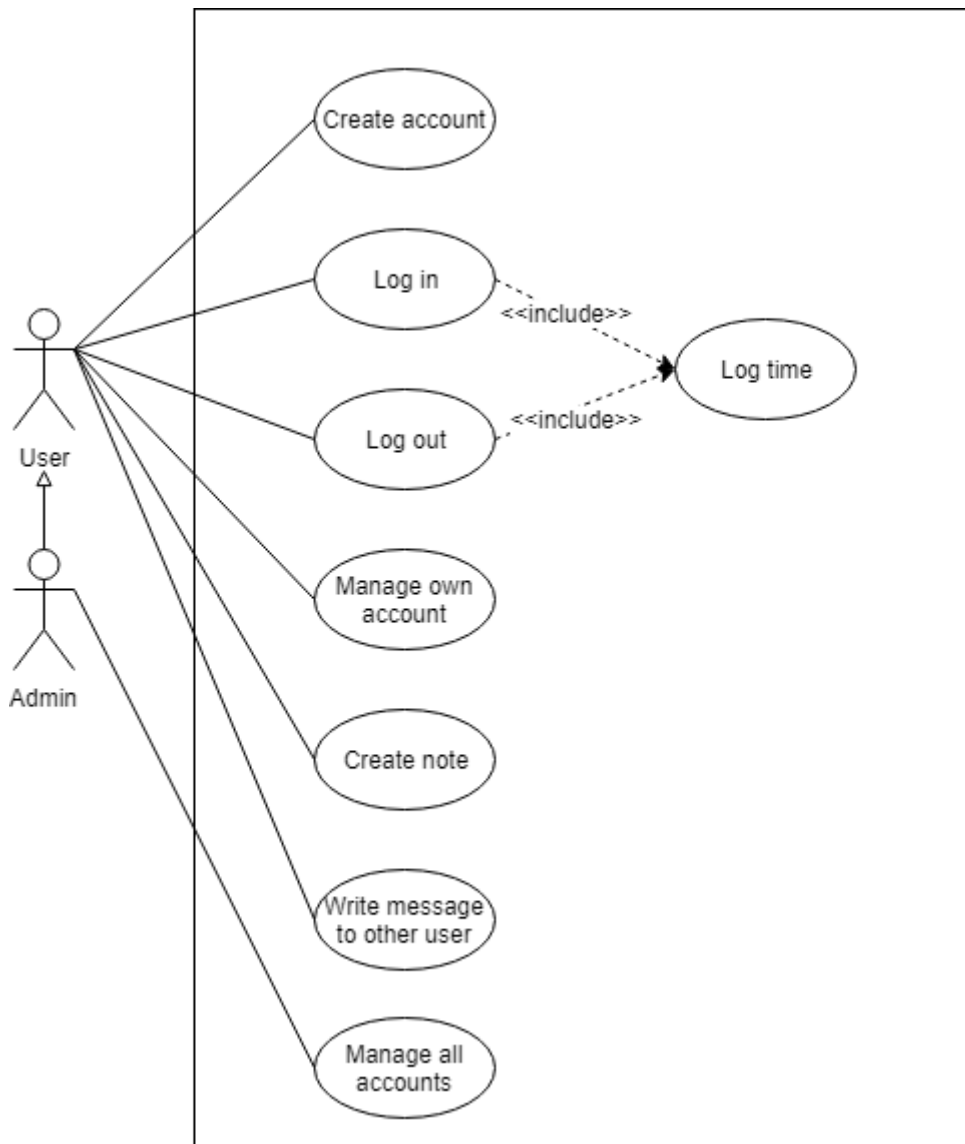


Praktische Prüfung (LB-B)

Anforderungen Analysieren

System abbilden

Anwendungsdiagramm:



Akteure definieren

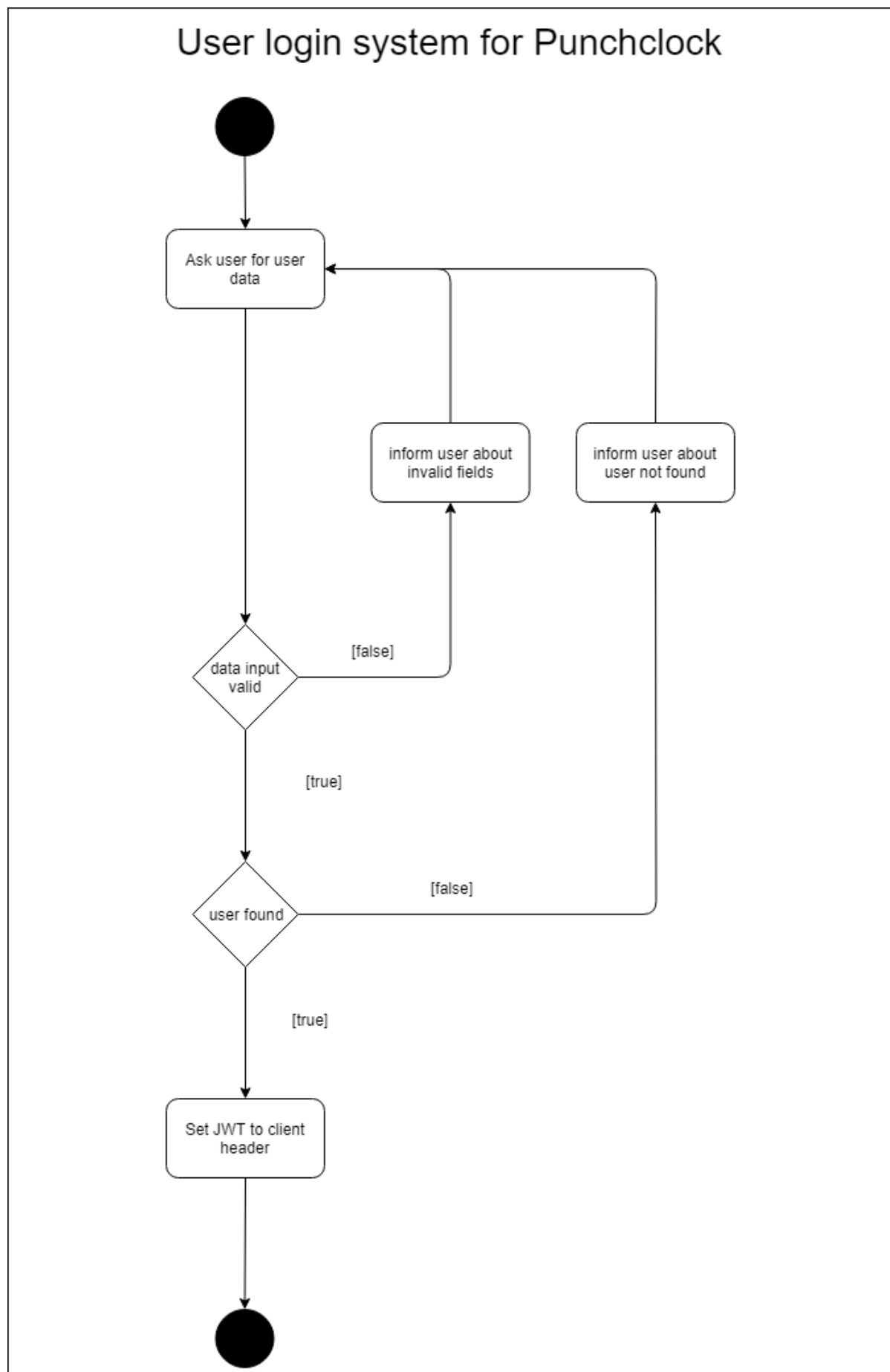
Während ein Benutzer nur Zugriff zu seinem Konto hat, hat der Administrator zusätzlich die Optionen, dass er jedes Konto zu seinem beliebigen anschauen, verändern und löschen kann.

Anforderungen beschreiben

Anforderungstabelle:

Titel	Satz	Vorbedingungen
Create account	Auf der Webseite hat der Benutzer die Möglichkeit ein Konto zu erstellen, solange alle Angaben den Anforderungen entsprechen (= valide sind).	-
Log in	Auf der Webseite hat der Benutzer die Möglichkeit sich mit einem bestehenden Konto anzumelden, indem er einen existierenden Benutzernamen und das dazugehörige Passwort eingibt.	Man muss zuvor ein Konto erstellt haben.
Log out	Man hat die Möglichkeit sich jeder Zeit abzumelden und falls dies für eine zu lange Zeit nach der Anmeldung nicht eintritt wird dafür gesorgt, dass der Benutzer aus Sicherheitsgründen automatisch abgemeldet ist.	Man muss zuvor angemeldet gewesen sein.
Log time	Bei An- oder Abmeldung eines Benutzers wird die exakte Zeit der Aktion geloggt.	Eine Log in oder Log out Aktion muss durchgeführt werden.
Manage own account	Man hat die Möglichkeit auf dem Konto, auf welchem man gerade eingeloggt ist, seine Daten zu sehen, ändern oder sein Konto zu löschen.	Man muss angemeldet sein.
Create note	Man hat die Möglichkeit eine oder mehrere Notizen für sich zu erstellen.	Man muss angemeldet sein.
Write message to other user	Man hat die Möglichkeit eine Nachricht an einen anderen Benutzer zu schicken.	Man muss angemeldet sein.
Manage all accounts	Als Administrator, hat man die Möglichkeit alle Konten so anzupassen, wie man will.	Man muss mit Administratorrechten angemeldet sein.

Aktivitätsdiagramm für Loginanforderung:

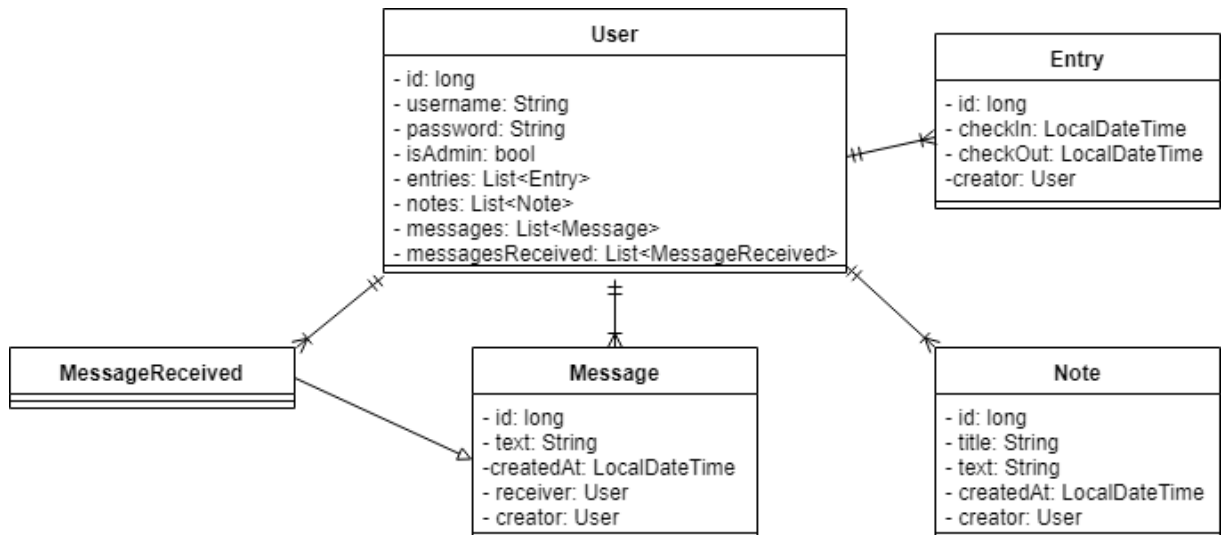


Nicht-funktionale Anforderungen beschreiben

- Der JWT wird nach 20 Minuten ungültig.
- Alle Benutzerinfos müssen per REST API innerhalb 100ms nach Request, zurückgegeben werden.
- Bei 3 Fehlversuchen im Log-in wird eine ReCAPTCHA Verifikation verlangt.
- Bei 10 Fehlversuchen wird die IP gesperrt.

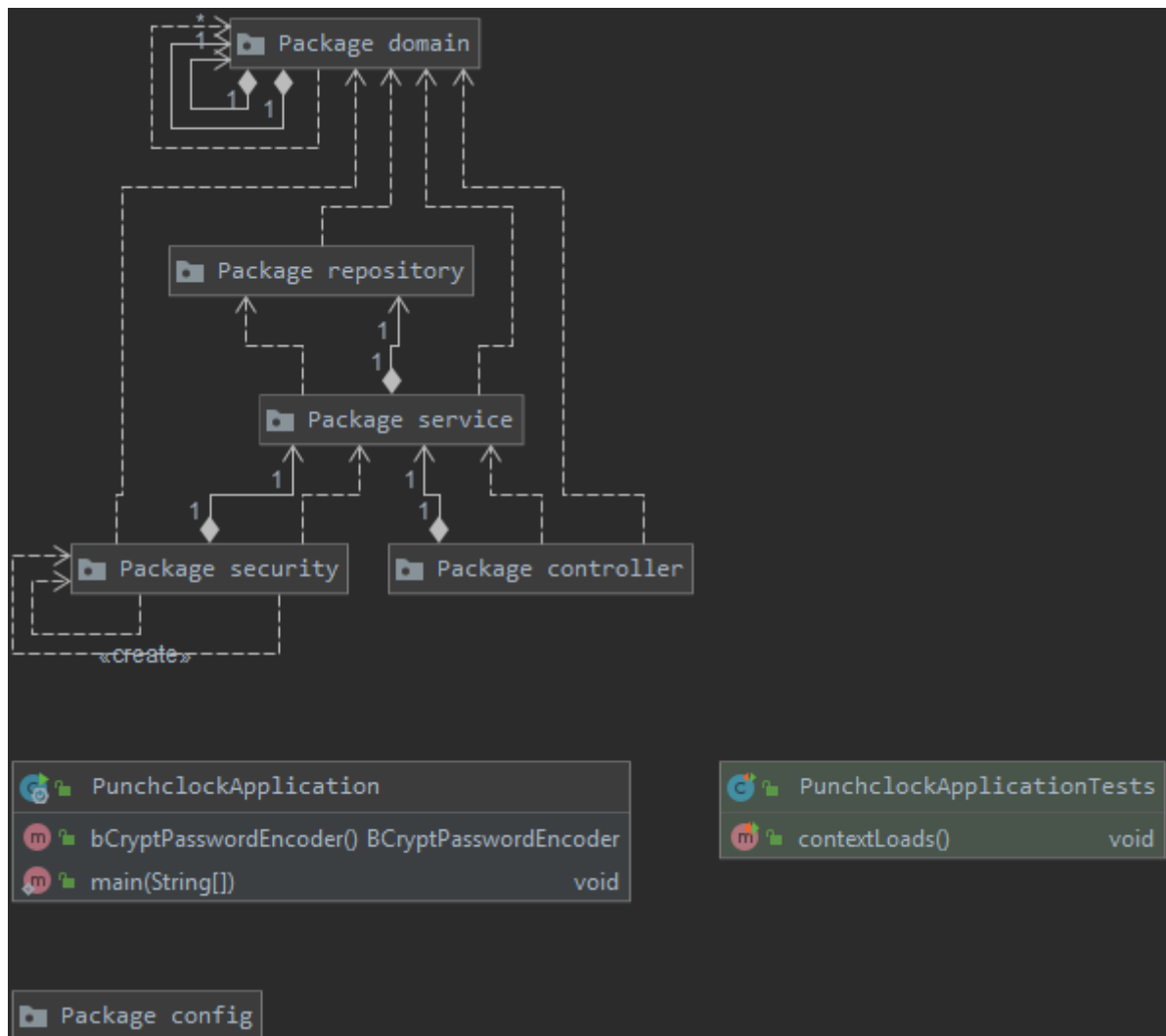
Datenhaltung gestalten

Fachklassendiagramm:



Architektur planen

Packagediagramm:

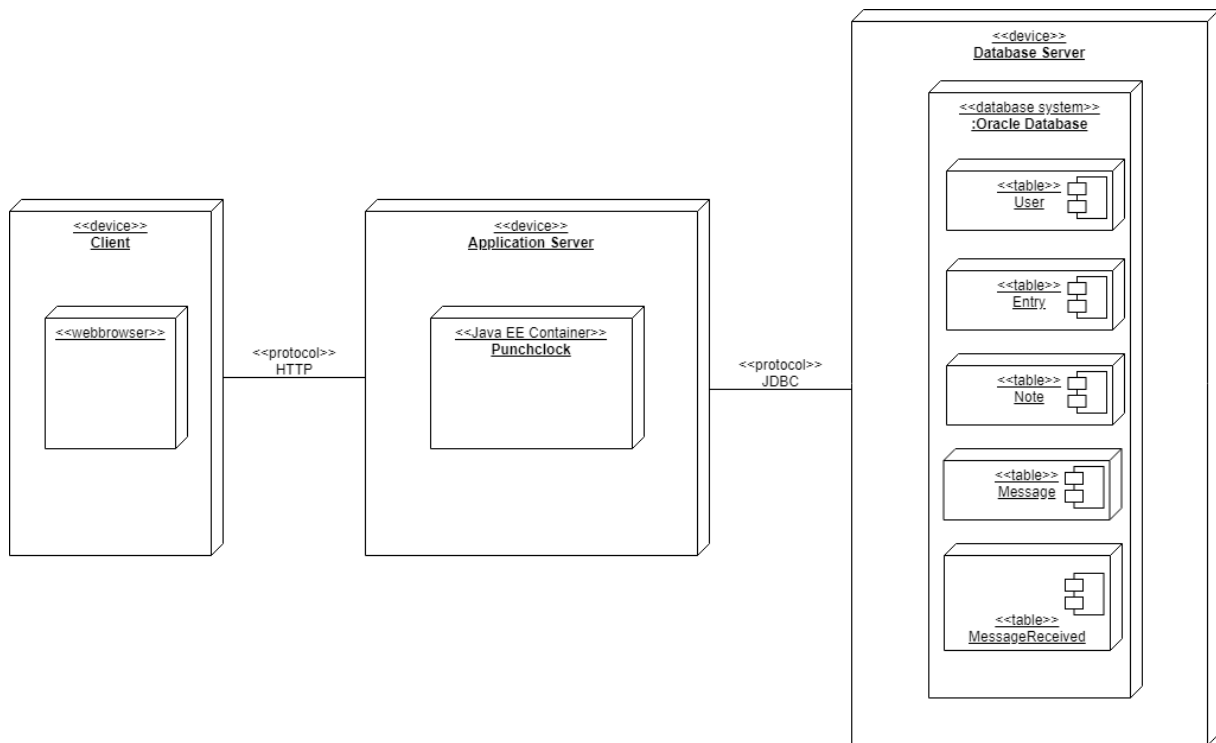


The UML class diagram illustrates the structure of a Spring Boot application. It includes the following classes and their attributes:

- SecurityConstants**:
 - SECRET: String
 - EXPIRATION_TIME: long
 - TOKEN_PREFIX: String
 - HEADER_STRING: String
 - SIGN_UP_URL: String
- WebSecurity**:
 - userService: UserServiceImpl
 - bCryptPasswordEncoder: BCryptPasswordEncoder
 - configure(HttpSecurity): void
 - configure(AuthenticationManagerBuilder): void
 - configure(ConfigurationSource): ConfigurationSource
- AuthenticationManager**:
 - attemptAuthentication(HttpServletRequest, HttpServletResponse): Authentication
 - successfulAuthentication(HttpServletRequest, HttpServletResponse, FilterChain, Authentication): void
- JWTAuthenticationFilter**:
 - doFilterInternal(HttpServletRequest, HttpServletResponse, FilterChain): void
 - getAuthentication(HttpServletRequest, UsernamePasswordAuthenticationToken): Authentication
- UserService**:
 - userRepository: UserRepository
 - bCryptPasswordEncoder: BCryptPasswordEncoder
 - createUser(User): User
 - deleteUser(User): void
 - findAll(): List<User>
- UserRepository**:
 - findByUsername(String): User
- MessageReceived**:
 - createdAt: LocalDateTime
 - creator: User
 - text: String
 - id: long
- Message**:
 - createdAt: LocalDateTime
 - creator: User
 - text: String
 - id: long
- Entry**:
 - checkedOut: LocalDateTime
 - creator: User
 - checkedIn: LocalDateTime
 - id: long
- Note**:
 - createdAt: LocalDateTime
 - creator: User
 - text: String
 - id: long
- User**:
 - password: String
 - id: long
 - notes: List<Note>
 - username: String
 - entries: List<Entry>
 - messages: List<Message>
 - admin: boolean

Relationships (Associations) are shown with solid lines and hollow diamonds at the end of the association line, indicating aggregation or composition. Dashed lines with hollow diamonds at the end of the association line indicate a weak association. The diagram shows a complex network of relationships between these classes, including associations between SecurityConstants and WebSecurity, WebSecurity and AuthenticationService, AuthenticationService and JWTAuthenticationFilter, JWTAuthenticationFilter and UserRepository, UserRepository and UserService, UserService and MessageReceived, MessageReceived and Message, Message and Entry, Entry and Note, Note and User, and User and MessageReceived.

Deployment Diagramm:



Testfälle planen

Funktionale Anforderungen:

Schritt #	Beschreibung	Erwartetes Resultat	Eigentliches Resultat
1	Registrieren	Auf der Webseite hat der Benutzer die Möglichkeit ein Konto zu erstellen, solange alle Angaben den Anforderungen entsprechen (= valide sind).	Auf der Webseite hat der Benutzer die Möglichkeit ein Konto zu erstellen, solange alle Angaben den Anforderungen entsprechen (= valide sind).
2	Einloggen	Auf der Webseite hat der Benutzer die Möglichkeit sich mit einem bestehenden Konto anzumelden, indem er einen existierenden Benutzernamen und das dazugehörige Passwort eingibt.	Auf der Webseite hat der Benutzer die Möglichkeit sich mit einem bestehenden Konto anzumelden, indem er einen existierenden Benutzernamen und das dazugehörige Passwort eingibt.
3	Ausloggen	Nachdem sich der Benutzer erfolgreich authentifizieren konnte und die Anmeldung in sein Konto gelang, hat er nun die Möglichkeit sich jeder Zeit wieder abzumelden und falls dies für eine zu lange Zeit nach der Anmeldung nicht eintritt wird dafür gesorgt, dass der Benutzer aus Sicherheitsgründen automatisch abgemeldet ist.	Nachdem sich der Benutzer erfolgreich authentifizieren konnte und die Anmeldung in sein Konto gelang, hat er nun die Möglichkeit sich jeder Zeit wieder abzumelden und falls dies für eine zu lange Zeit nach der Anmeldung nicht eintritt wird dafür gesorgt, dass der Benutzer aus Sicherheitsgründen automatisch abgemeldet ist.
4	Zeit erfassen	Bei An- oder Abmeldung eines Benutzers wird die exakte Zeit der Aktion geloggt.	Bei An- oder Abmeldung eines Benutzers wird die exakte Zeit der Aktion geloggt.
5	Eigenes Konto anpassen	Man hat die Möglichkeit auf dem Konto, auf welchem man gerade eingeloggt ist, seine Daten zu sehen, ändern oder sein Konto zu löschen.	Man hat die Möglichkeit auf dem Konto, auf welchem man gerade eingeloggt ist, seine Daten zu sehen, ändern oder sein Konto zu löschen.
6	Notiz erstellen	Wenn man eingeloggt ist, hat man die Möglichkeit eine oder mehrere Notizen zu erstellen.	Wenn man eingeloggt ist, hat man die Möglichkeit eine oder mehrere Notizen zu erstellen.
7	Nachricht an anderen Benutzer schreiben	Wenn man eingeloggt ist, hat man die Möglichkeit eine Nachricht an einen anderen Benutzer zu schicken.	Funktion im UI nicht verfügbar.
8	Alle Konten anpassen	Als Administrator, hat man die Möglichkeit alle Konten so anzupassen, wie man will.	Als Administrator, hat man die Möglichkeit alle Konten so anzupassen, wie man will.

Nicht funktionale Anforderungen:

9	JWT Frist	Der JWT wird nach 20 Minuten ungültig.	Der JWT wird nach 20 Minuten ungültig.
10	Ressourcen-effizienz	Alle Benutzerinfos werden per REST API innerhalb 100ms nach Request zurückgegeben.	Alle Benutzerinfos werden per REST API innerhalb einem Durchschnitt von 120ms nach Request zurückgegeben.
11	ReCAPTCHA	Bei 3 Fehlversuchen im Log-in wird eine ReCAPTCHA Verifikation verlangt.	Bei 3 Fehlversuchen im Log-in wird eine ReCAPTCHA Verifikation verlangt.
12	IP bannen	Bei 10 Fehlversuchen wird die IP gesperrt.	Bei 10 Fehlversuchen wird die IP gesperrt.

Status OK

Status NOK

Schnittstelle (UI)

Startseite Idee:

Welcome to Punchclock

Sign in

Sign up

Git History

File Edit View Repository Branch Help	
Current repository Punchclock	
Changes (43)	History
No branches to compare	
Removed node modules + added README.md to main directory. Danilo Furrer committed 9 minutes ago	
Added delete request function for user controller. Danilo Furrer committed 9 hours ago	
Added script in usercontroller constructor to add users on run of project. Danilo Furrer committed 9 hours ago	
Added missing doc. Danilo Furrer committed 10 hours ago	
Finished the java docs for the api interfaces. Danilo Furrer committed 11 hours ago	
Fixed bug where time adjustment for anyone other than the first user wouldn't work. Danilo Furrer committed 11 hours ago	
Added possibility to adjust entries. Danilo Furrer committed 12 hours ago	
Added entry + user support for client. codpro2005 committed 18 hours ago	
Created home page with user sign out + extended httpservice. Todo: check for delete... Danilo Furrer committed a day ago	
Added JWT authorization by cookie on client + fixed bug where headers weren't rece... Danilo Furrer committed 2 days ago	
Fixed small bug where sign-in api request was of invalid url. Danilo Furrer committed 2 days ago	
Created httpservice to handle api requests. Danilo Furrer committed 2 days ago	
Fixed bug where on empty form submit password confirm control would be valid. Danilo Furrer committed 3 days ago	
Changed protection level on passwordConfirm validator to private. Danilo Furrer committed 3 days ago	
Fixed bug where password validation error would be triggered on all formgroup erro... Danilo Furrer committed 3 days ago	
Added password confirm validation. Danilo Furrer committed 3 days ago	
Authentication process is now devided into 3 components: authentication parent, sig... Danilo Furrer committed 3 days ago	
Created authentication component (including sign-in and sign-up), added angular m... Danilo Furrer committed 4 days ago	
added getCurrentUser method which returns the user who's username is saved in the... Danilo Furrer committed 4 days ago	
implemented JWT + added new entities Danilo Furrer committed 5 days ago	
Merge pull request #1 from zlict/improve-project Robin committed Sep 3, 2019	
Fix small bugs, remove spring-boot-starter-data-rest dependency Manuel Allenspach committed Sep 3, 2019	
Add serializer und deserializer for dates Robin committed Sep 2, 2019	
Scaffold application Robin committed Sep 1, 2019	
Scaffold application Robin committed Sep 1, 2019	
Initial commit Robin committed Sep 1, 2019	

Git log:

commit 4489dc71659ea9c90756a933810641a92004a7e5 (HEAD -> master, origin/master, origin/HEAD)
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Sat Sep 21 11:42:46 2019 +0200
Removed node modules + added README.md to main directory.

commit a5108c2fc22eab4d9b10bb72c74fdc8cff835665
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Sat Sep 21 03:03:29 2019 +0200
Added delete request function for user controller.

commit f2de80f724a05a03bfbeee0dcaaf263c02c2d77a
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Sat Sep 21 02:40:49 2019 +0200
Added script in usercontroller constructor to add users on run of project.

commit 6065c5ce78386e6c9cb94a3efcb1666d48a8b882
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Sat Sep 21 01:26:49 2019 +0200
Added missing doc.

commit e12496fcf3563be2135171b8e65cf1efc89322ae
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Sat Sep 21 00:54:02 2019 +0200
Finished the java docs for the api interfaces.

commit c5ed58dfde61fbebdec511e55dfc94e8b19ae165
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Sat Sep 21 00:23:42 2019 +0200
Fixed bug where time adjustment for anyone other than the first user wouldn't work.

commit 4e4102d2f7c094af429f9adfbcb3152d796bbf2a3
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Sat Sep 21 00:06:47 2019 +0200
Added possibility to adjust entries.

commit 3cfac2f547bdd0569302aaaa1590816e0a188642
Author: codpro2005 <40277095+codpro2005@users.noreply.github.com>
Date: Fri Sep 20 17:56:23 2019 +0200
Added entry + user support for client.

commit 1936f1e1a02366ad63f9ecfbddc5b450a200fadd
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Fri Sep 20 02:04:51 2019 +0200
Created home page with user sign out + extended httpservice. Todo: check for delete and log time error.

commit 4c146fb8c50e93734a75f493870865047ae42404
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Thu Sep 19 19:04:47 2019 +0200
Added JWT authorization by cookie on client + fixed bug where headers weren't received on client by modern browsers

commit 4fb2fb27495d799c0742aa967f32d218f99998af
Author: Danilo Furrer <danilo.furrer@outlook.com>
Date: Wed Sep 18 23:50:51 2019 +0200
Fixed small bug where sign-in api request was of invalid url.

commit 631c0718937067e9f497ddb25f83ddb6d84ce4bf

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Wed Sep 18 23:44:40 2019 +0200

Created httpservice to handle api requests.

commit c59deb0a6848a0c0bbdb2a43ec88ccee37a092ad

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Wed Sep 18 20:27:29 2019 +0200

Fixed bug where on empty form submit password confirm control would be valid.

commit ad7bd0d86ea122130804f0ef30a50ef9cf41dad8

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Wed Sep 18 20:11:18 2019 +0200

Changed protection level on passwordConfirm validator to private.

commit ce535dd2df070b1d11cf941cbe4cef85fb73cbfb

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Wed Sep 18 20:08:14 2019 +0200

Fixed bug where password validation error would be triggered on all formgroup errors rather than just differentPassword error.

commit 3b3bf7158d5b5df993f896f7c8a0130c92641e55

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Wed Sep 18 20:03:29 2019 +0200

Added password confirm validation.

commit 62c09bde22e15e6314230c528a26e97ae24638d9

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Wed Sep 18 00:25:34 2019 +0200

Authentication process is now divided into 3 components: authentication parent, sign-in child & sign-up child and has been made more dynamic.

commit 9f8803ab565143d08c054fbf4d52eccddf4c33b6

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Tue Sep 17 22:25:04 2019 +0200

Created authentication component (including sign-in and sign-up), added angular material and added guard.

commit ba8c3a2254430051939b1e14f1633c07410a3fb7

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Tue Sep 17 00:44:42 2019 +0200

added getCurrentUser method which returns the user who's username is saved in the jwt payload, made username unique and created client project with Angular

commit 24f257e41d6a48a857a797ae415b5c6b434c2738

Author: Danilo Furrer <danilo.furrer@outlook.com>

Date: Mon Sep 16 19:23:25 2019 +0200

implemented JWT + added new entities

commit 7cbb42dc6527658ca10fd6bd2912141ed1831424

Merge: 54ef77eb acf06c4f

Author: Robin <1105080+openscript@users.noreply.github.com>

Date: Tue Sep 3 14:31:48 2019 +0200

Merge pull request #1 from zlict/improve-project

Fix small bugs, remove spring-boot-starter-data-rest dependency

commit acf06c4fcb0f69e7fd2eab73fc6e49dc37ff908f

Author: Manuel Allenspach <manuel@riok.ch>

Date: Tue Sep 3 13:39:51 2019 +0200

Fix small bugs, remove spring-boot-starter-data-rest dependency
commit 54ef77ebd52c0a0fe94eb647f587f505dd55e21b

Author: Robin <1105080+openscript@users.noreply.github.com>

Date: Mon Sep 2 11:23:50 2019 +0200

Add serializer und deserializer for dates

commit 6af25ec422a8580445867eb06cd942e9ddc83286

Author: Robin <1105080+openscript@users.noreply.github.com>

Date: Sun Sep 1 19:42:03 2019 +0200

Scaffold application

commit 8709cf461e7fe70e94c4cc8385307fb11189e609

Author: Robin <1105080+openscript@users.noreply.github.com>

Date: Sun Sep 1 19:41:39 2019 +0200

Scaffold application

commit de377d0065b5a311009f0500c68d3956fc84a478

Author: Robin <1105080+openscript@users.noreply.github.com>

Date: Sun Sep 1 18:16:36 2019 +0200

Initial commit

(END)