



PUNCHCLOCK PLANUNG

Planung und Dokumentation vom Punchclock
System

Jakob Danilo

Inhalt

Analyse der Anforderungen	1
Akteure	1
Anwendungsdiagramm	2
Anforderungen	2
Funktionale Anforderungen	2
Nicht-funktionale Anforderungen	3
Abbildung einer Anforderung.....	3
Datenhaltung.....	4
Fachklassendiagramm	4
Architektur.....	5
Packagediagramm	5
Klassendiagramm Beispiel Entry	6
Deploymentdiagramm	7
Testfälle	7

Analyse der Anforderungen

Akteure

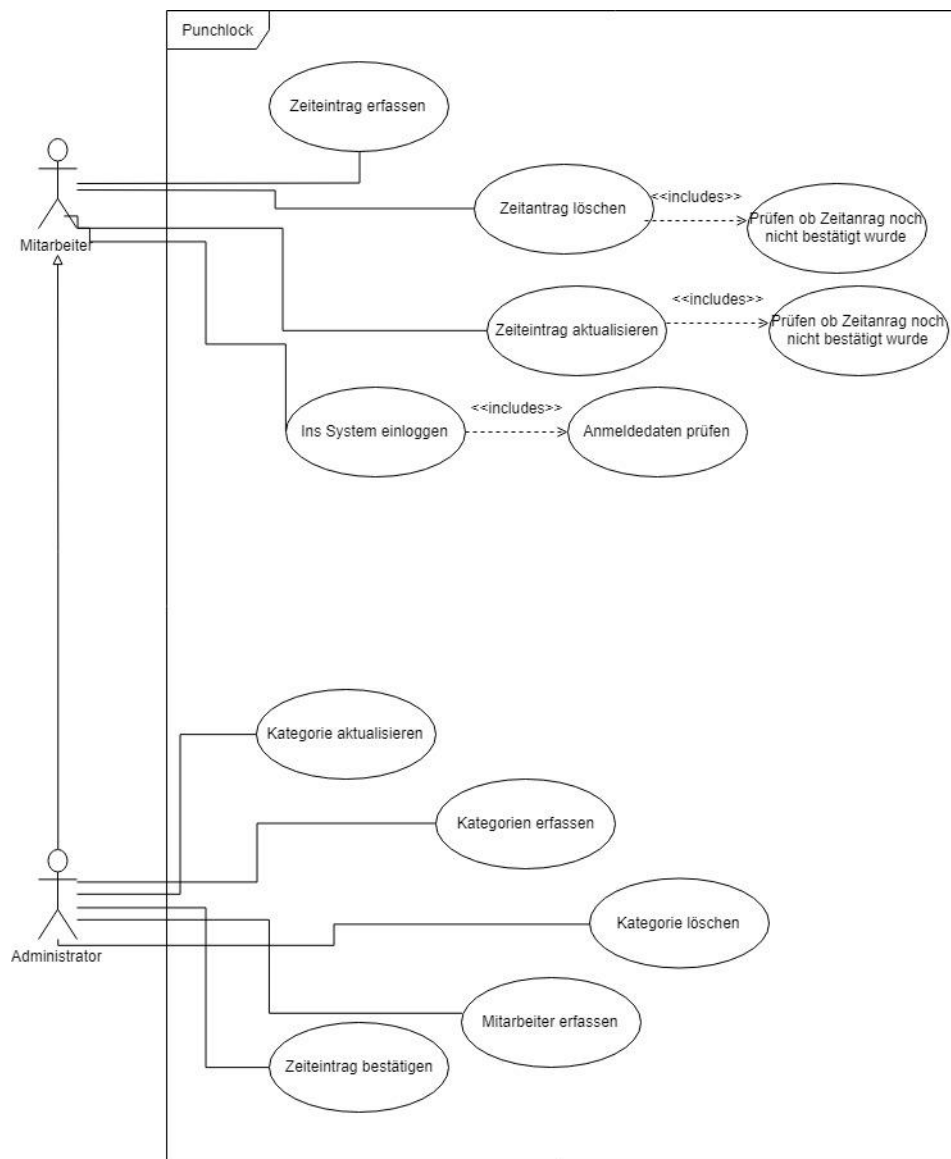
Im Punchclock Programm existieren zwei Akteure, dabei handelt es sich um:

- Mitarbeiter
- Administrator

Der Mitarbeiter soll in der Lage sein, seine Zeiten zu erfassen. Der Administrator hingegen soll in der Lage sein die eingetragenen Zeiten zu bestätigen, bzw. diese als bestätigt zu markieren.

Der Mitarbeiter steht mit dem Administrator im gleichen Verhältnis wie ein Personalchef zum Mitarbeiter.

Anwendungsdiagramm



Anforderungen

Funktionale Anforderungen

- **Zeiteintrag erfassen:** Der Mitarbeiter kann einen neuen Zeiteintrag erfassen
- **Zeiteintrag löschen:** Der Mitarbeiter kann sein Zeiteintrag wieder löschen falls dieser nicht bestätigt wurde
- **Zeiteintrag aktualisieren:** Der Mitarbeiter kann ein Zeitantrag aktualisieren falls dieser nicht bestätigt wurde
- **Ins System einloggen:** Der Mitarbeiter kann sich im System anmelden
- **Kategorie erfassen:** Der Administrator kann eine neue Kategorie erfassen

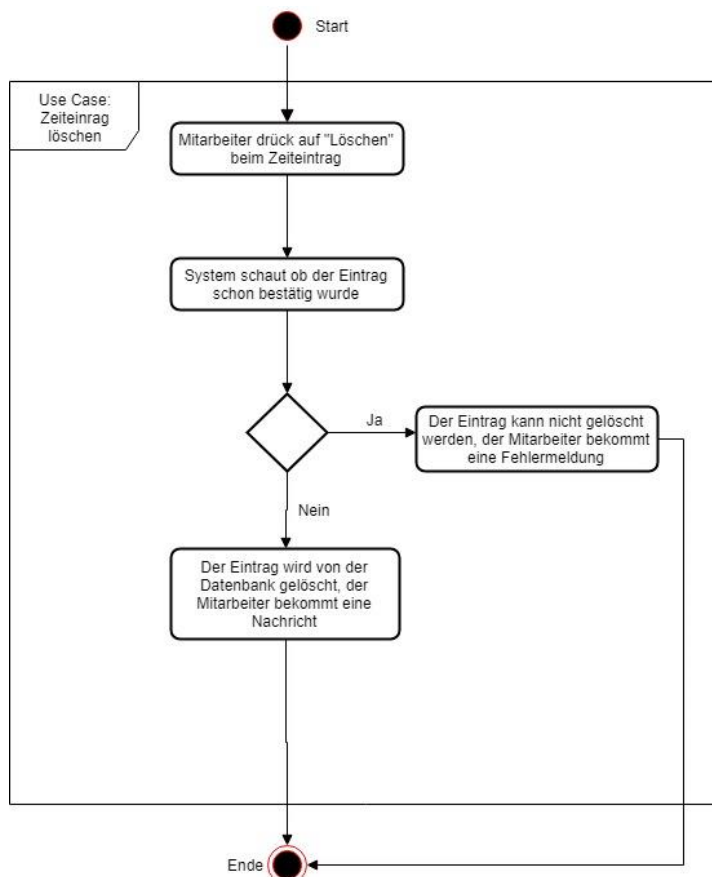
- **Kategorie löschen:** Der Administrator kann eine Kategorie löschen
- **Kategorie aktualisieren:** Der Administrator kann eine Kategorie aktualisieren
- **Zeiteintrag bestätigen:** Der Administrator kann ein Zeiteintrag bestätigen
- **Mitarbeiter erfassen:** Der Administrator kann neue Mitarbeiter erfassen im System

Nicht-funktionale Anforderungen

- **Applikation ist sicher:** Es können keine nicht Autorisierte Personen auf das System zugreifen, ausserdem kann ein Mitarbeiter keine Administrator Funktionen ausführen auf dem System
- **Applikation soll einfach zum deployen sein:** Das Programm soll ohne grossen Aufwand gestartet werden können, das heisst: Keine externe Datenbank soll verwendet werden, das System soll über ein Befehl gestartet werden können.
- **Das UI soll im Darkmode sein:** Für das Auge soll das System ansprechend sein, somit wird auf «White-Theme» verzichtet werden
- **Performance:** Das System wird unter einer Minute gestartet sein
- **Fehlerbehandlung:** Der Benutzer bekommt bei einem Fehler eine verständliche Fehlermeldung, bei welcher keine Informatikkenntnisse vorausgesetzt sind

Abbildung einer Anforderung

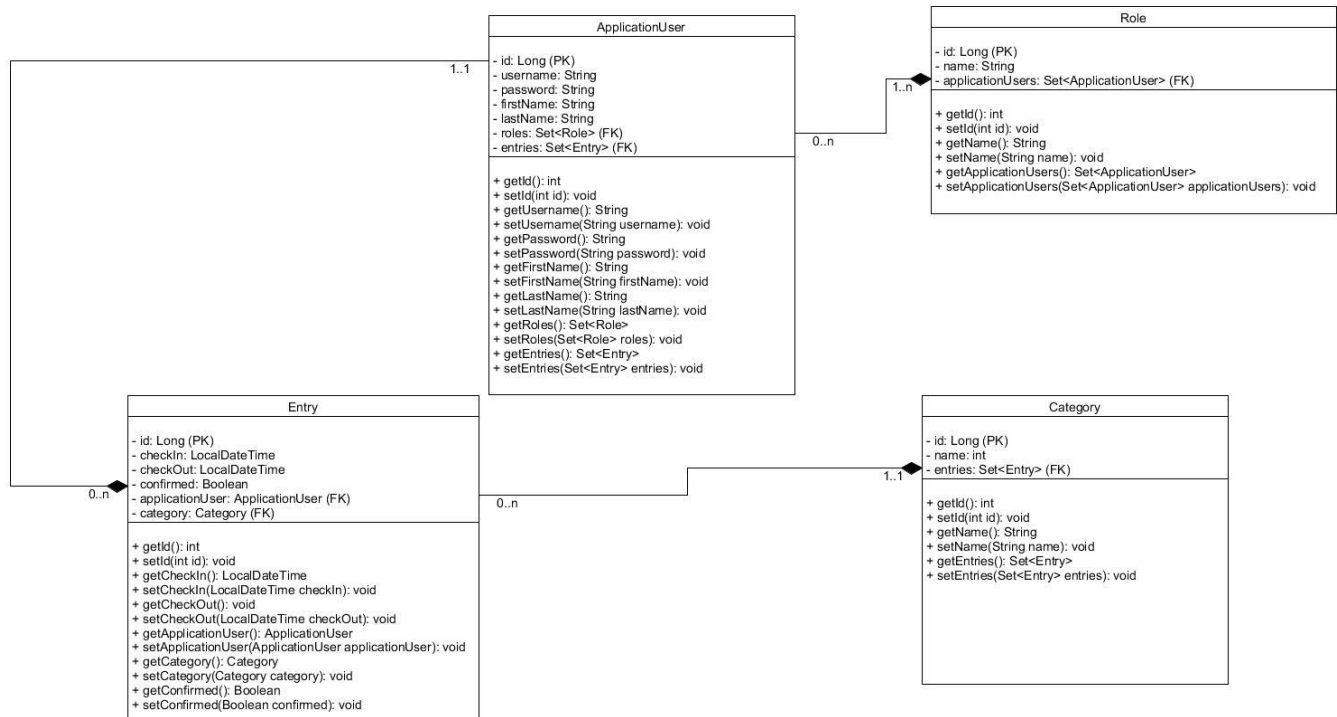
In diesem Abschnitt wird das löschen eines Zeiteintrages mithilfe eines Aktivitätsdiagrammes genauer beschrieben:



Datenhaltung

Fachklassendiagramm

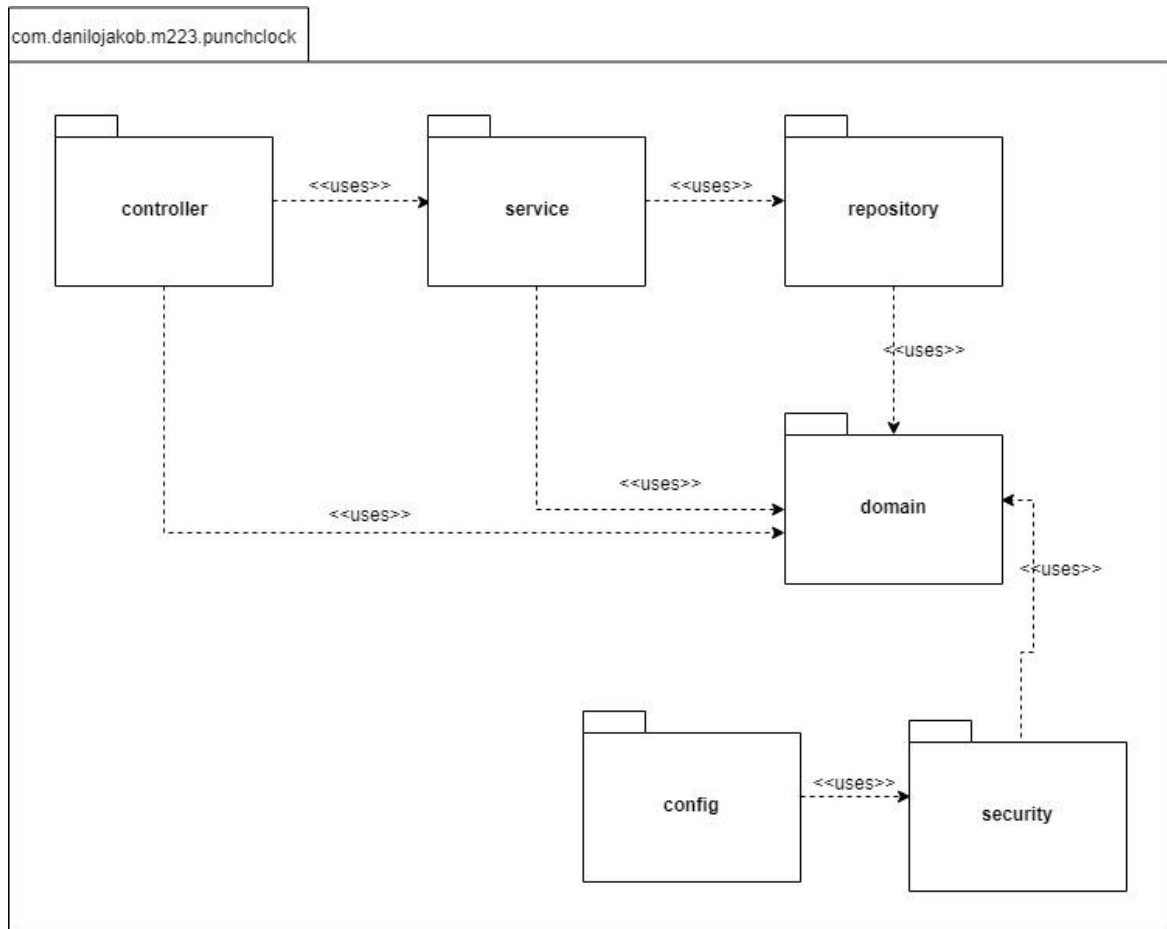
In diesem Abschnitt werden die Datenklassen mithilfe von einem Klassendiagramm dargestellt und ihre Relationen dokumentiert.



Architektur

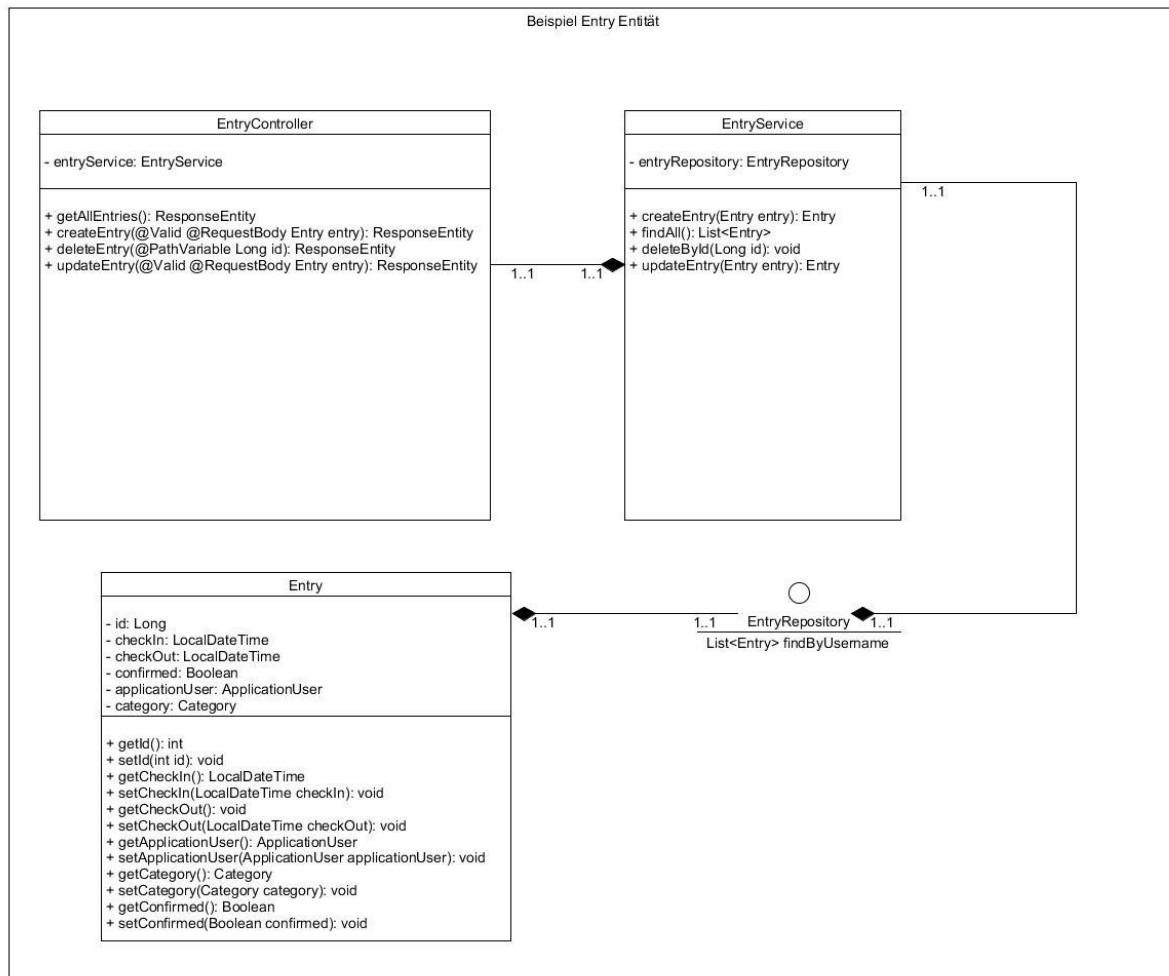
Packagediagramm

In diesem Abschnitt werden alle Pakete im Projekt dargestellt und dessen Beziehungen dokumentiert.



Klassendiagramm Beispiel Entry

In diesem Abschnitt wird dokumentiert wie die Struktur vom Controller bis zum Model aussieht, hierbei handelt es sich um die Entry Entität.

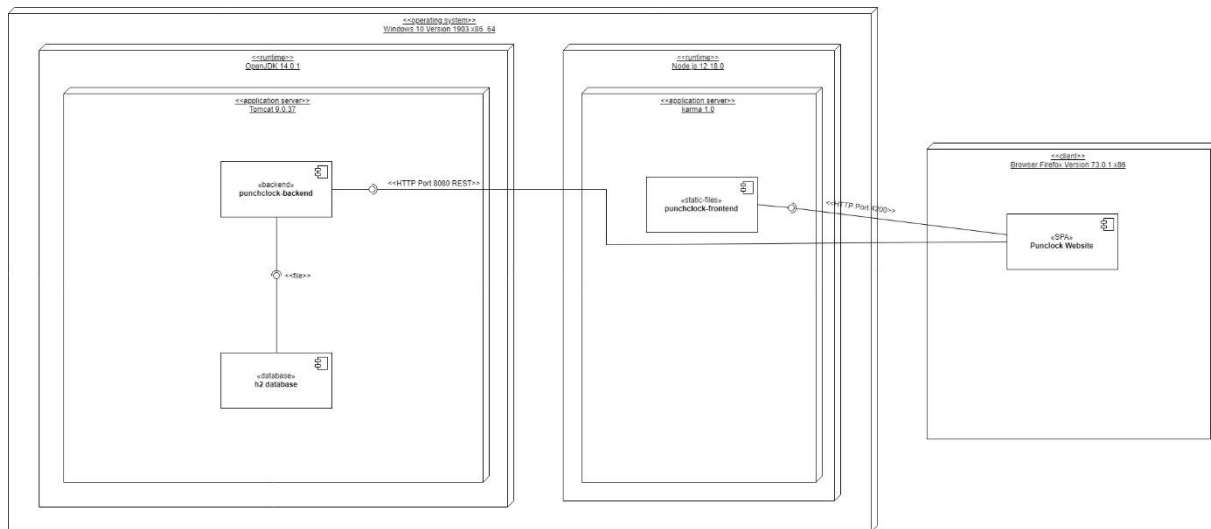


Mithilfe vom Diagram sieht man, das jede Requests welche auf dem Backend gemacht werden vom Controller entgegen genommen werden. Falls Logik für die Verarbeitung benötigt wird, wird zum Service delegiert welcher danach mithilfe vom Repository die Daten auf dem Backend verändert.

Ausserdem gibt es im EntryRepository eine eigene Methode welche auf einem **NamedQuery** gemapped ist welcher in der Entry Entität deklariert ist. Dieser wird aufgerufen sobald die Methode im Repository auferufen wird.

Deploymentdiagramm

In diesem Abschnitt wird die Deploymentlandschaft von Punchclock mithilfe eines Deploymentdiagramms dargestellt und dokumentiert.



Auf dem gleichen Server befindet sich der Backend Server welcher mit Spring auf einem Tomcat server läuft, parallel läuft das Frontend mit Angular auf einem karma Server.

Testfälle

In diesem Abschnitt werden sechs Funktionale und drei nicht-funktionale Tests beschrieben.

Test Name	Beschreibung	OK / NOK (OK, nicht OK)
Zeiteintrag erfassen	Mit einem autorisierten Mitarbeiter über das Frontend einen neuen Zeiteintrag erstellen. Dieser sollte danach in der Übersicht auftauchen und als «nicht bestätigt» erscheinen.	
Zeiteintrag aktualisieren	Mit einem autorisierten Mitarbeiter den nicht bestätigten Zeiteintrag mit einer neuen Kategorie aktualisieren. Der Eintrag sollte mit der neuen Kategorie in der Übersicht erscheinen	
Zeiteintrag bestätigen	Mit einem autorisierten Administrator den vorher aktualisierten Zeiteintrag bestätigen. Der Eintrag sollte	

	nun in der Übersicht als «bestätigt» erscheinen.	
Zeiteintrag löschen	Mit einem autorisierten Benutzer einen bestätigten Zeiteintrag löschen. Es sollte eine Fehlermeldung kommen da man bestätigte Zeiteinträge nicht löschen kann	
Mitarbeiter erfassen	Mit einem autorisierten Benutzer einen neuen Mitarbeiter erstellen, es sollte der neue Mitarbeiter in der Übersicht erscheinen.	
Authentifizierung als Mitarbeiter	Über den Login Screen isch versuchen anzumelden	
Als Mitarbeiter versuchen auf die Administrationsansicht zu kommen	Mit einem autorisierten Mitarbeiter versuchen die Administrationsansicht zu öffnen. Dies sollte nicht funktionieren, da diese Ansicht versteckt ist, und die REST Schnittstelle die CRUD Operationen nicht zulässt	
Server starten	Versuchen den Source Code mithilfe vom Befehl: <code>./gradlew.bat bootRun</code> Zu starten. Dies sollte unterhalb einer Minute geschehen	
Darkmode	Das UI sollte direkt im Darkmode verfügbar sein	