

Abschlussprojekt “Cronus” ÜK 223

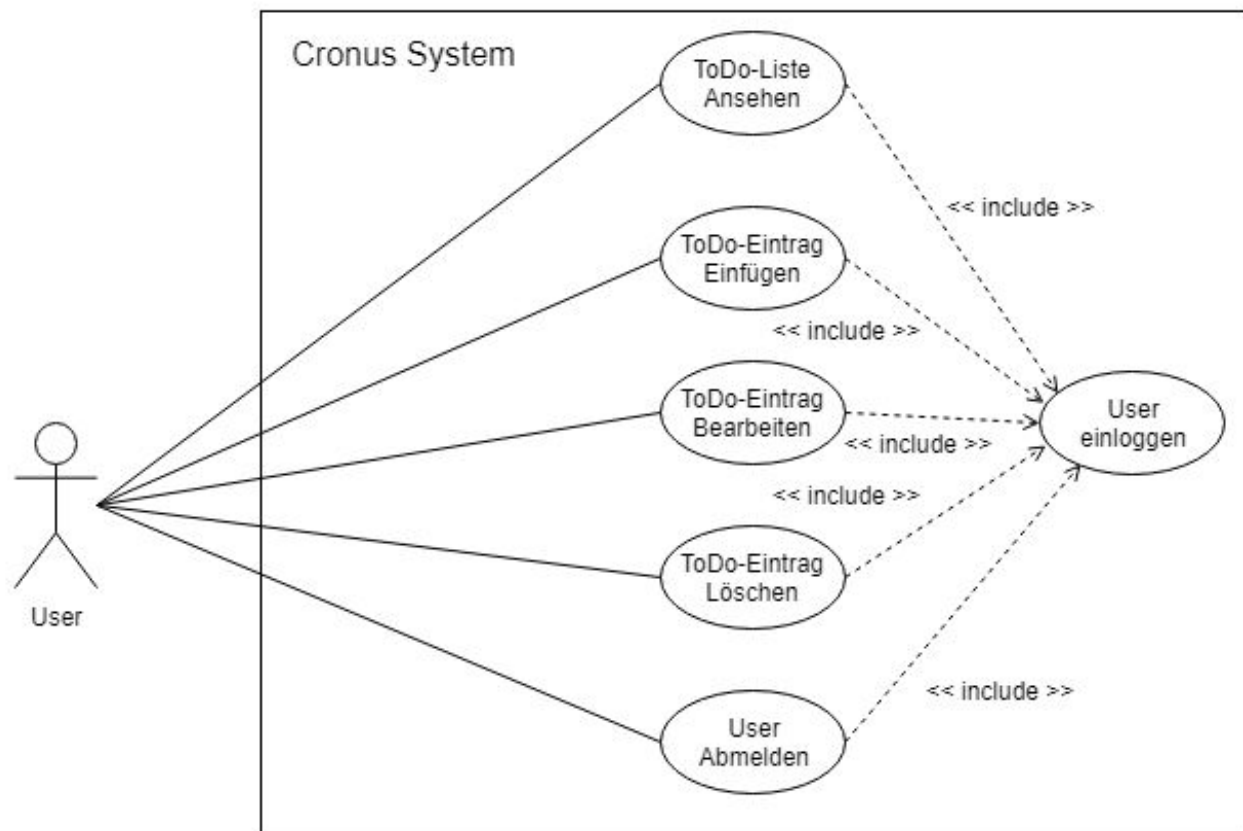
Projektbeschreibung Cronus	2
Systemanforderungen	3
Systemkontext	3
Akteure	3
Anforderungen	4
User Einloggen Aktivitätsdiagramm:	5
Nicht-Funktionale Anforderungen	6
Datenhaltung	7
Fachklassen	7
Architektur	8
Paketdiagramm	8
Klassendiagramm	9
Deployment Diagramm	10
Testfälle	11
Anforderungstests	11
Nicht-Funktionale Tests	12

Projektbeschreibung Cronus

Mein Projekt ist eine Digitale ToDo-Liste. Jeder Benutzer kann sich anmelden und dann seine eigene ToDo-Liste verwalten, sprich Einträge hinzufügen, löschen oder bearbeiten. Optional falls noch Zeit vorhanden kann mittels einem Admin-Konto kann man alle Listen ansehen und bearbeiten sowie Benutzer löschen, bearbeiten oder hinzufügen.

Systemanforderungen

Systemkontext



Akteure

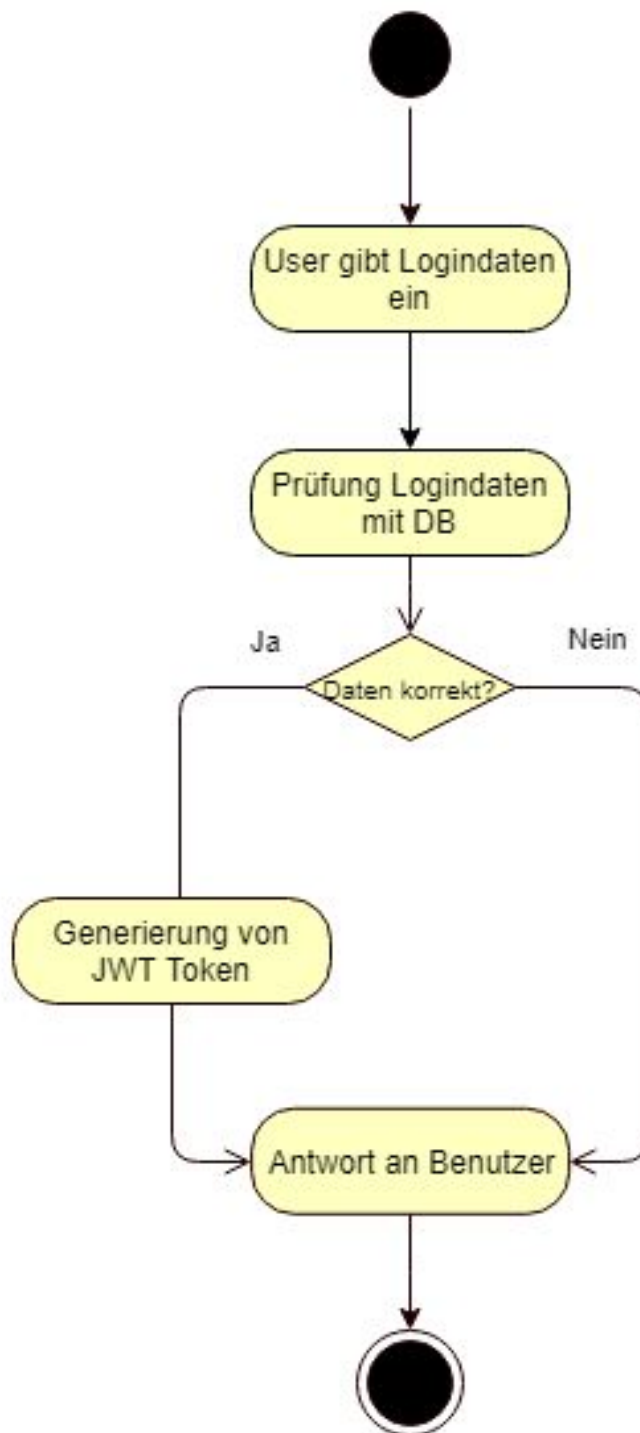
User :

Dies ist der "normale" Benutzer, sprich ein Benutzer der lediglich die basis Berechtigungen um seine Liste zu verwalten hat.

Anforderungen

Anforderung	Beschreibung	Vorbedingungen
ToDo-Liste Ansehen	Gibt dem User alle seine ToDo-Einträge zurück.	Der Benutzer hat ein Konto und ist eingeloggt.
ToDo-Eintrag Einfügen	Erstellt einen neuen Eintrag	Der Benutzer hat ein Konto und ist eingeloggt.
ToDo-Eintrag Bearbeiten	Bearbeitet den selektierten Eintrag.	Der Benutzer hat ein Konto, ist eingeloggt und hat einen Eintrag selektiert.
ToDo-Eintrag Löschen	Löscht den selektierten Eintrag.	Der Benutzer hat ein Konto und ist eingeloggt und hat einen Eintrag selektiert.
User Abmelden	Meldet den Benutzer ab (Sprich löscht beim Client den Token).	Der Benutzer hat ein Konto und ist eingeloggt.
User Einloggen	Überprüft ob die Daten korrekt sind und generiert dann einen Token um den Benutzer anzumelden.	Der Benutzer hat ein Konto und hat die richtigen Daten eingegeben.

User Einloggen Aktivitätsdiagramm:



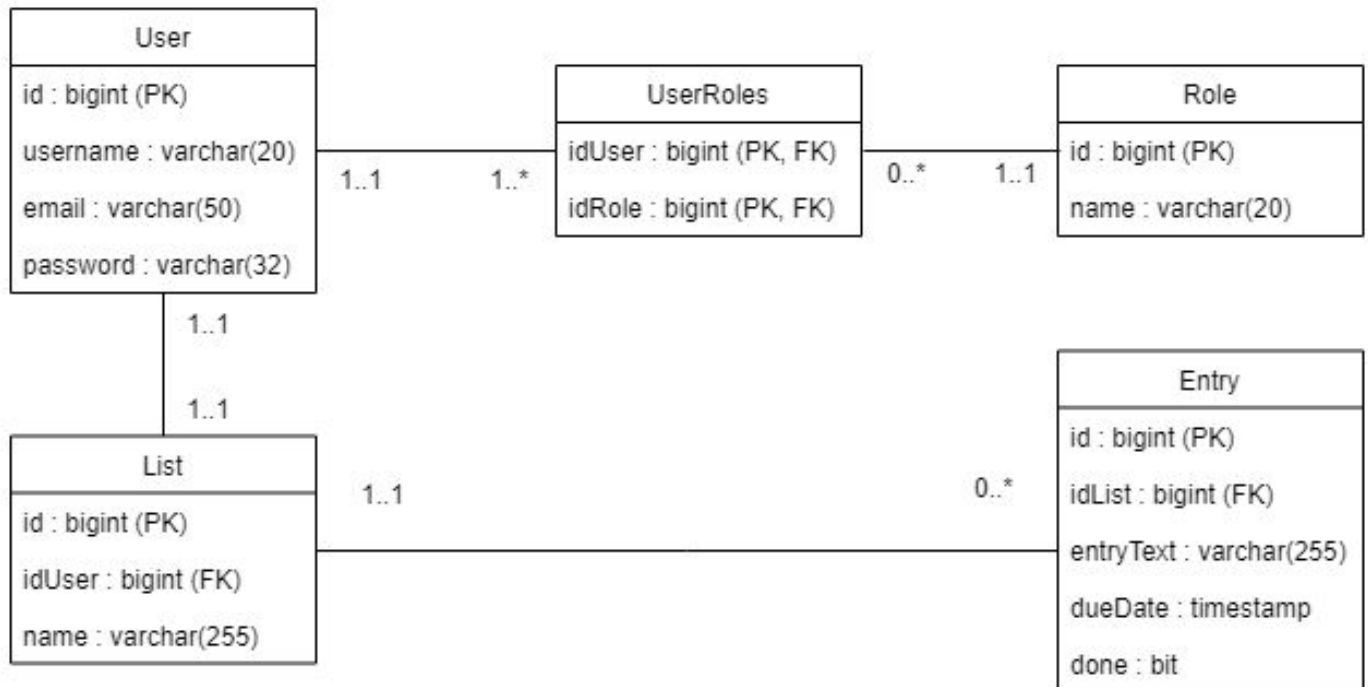
Nicht-Funktionale Anforderungen

Folgende nicht-funktionale Anforderungen sollen gegeben sein:

- Die Applikation muss unter Windows laufen.
- Das Programm soll nicht länger als 10 Sekunden zum antworten brauchen.
- Das GUI soll auf Chrome, Firefox und Opera gleich aussehen.
- Der Server soll in Java umgesetzt werden.

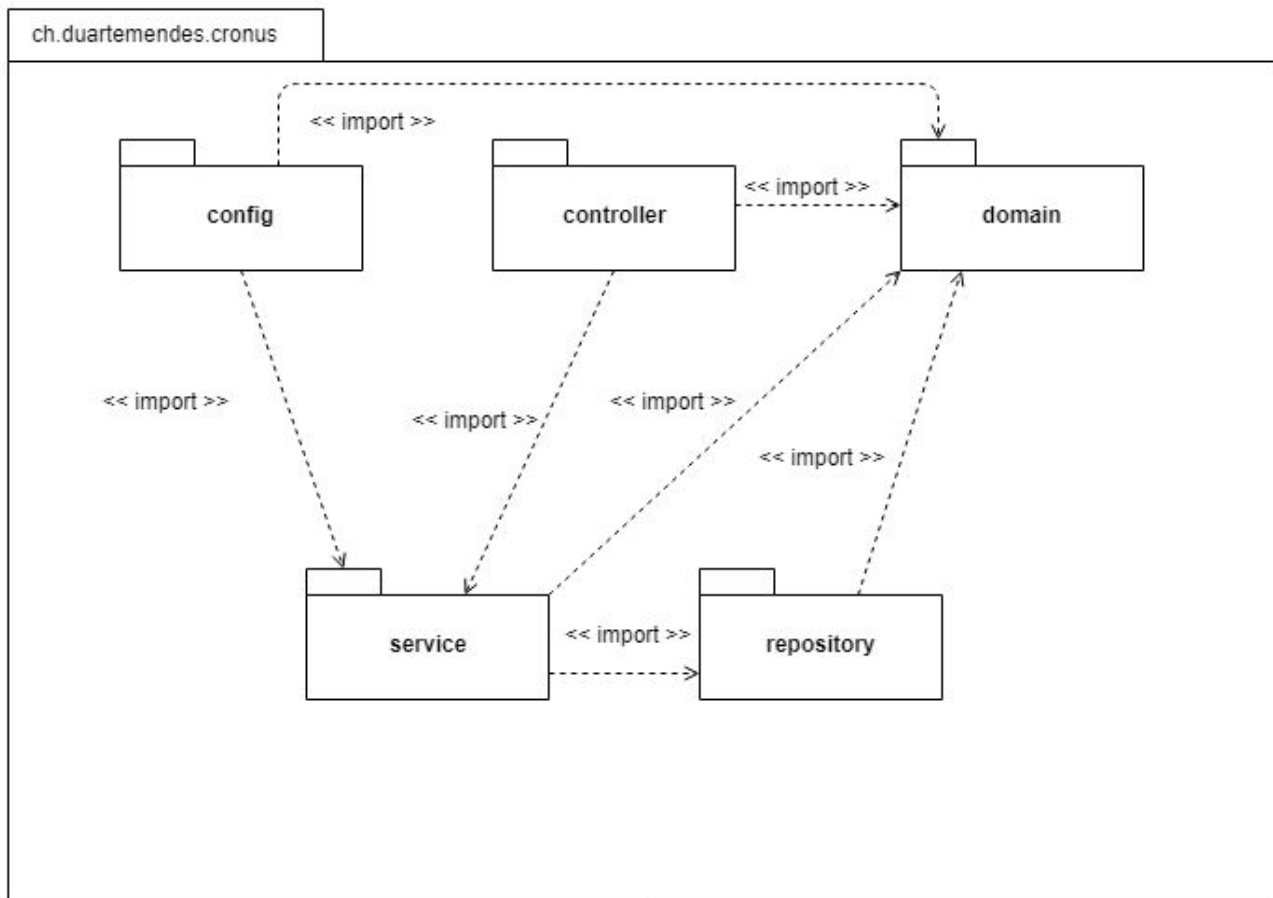
Datenhaltung

Fachklassen

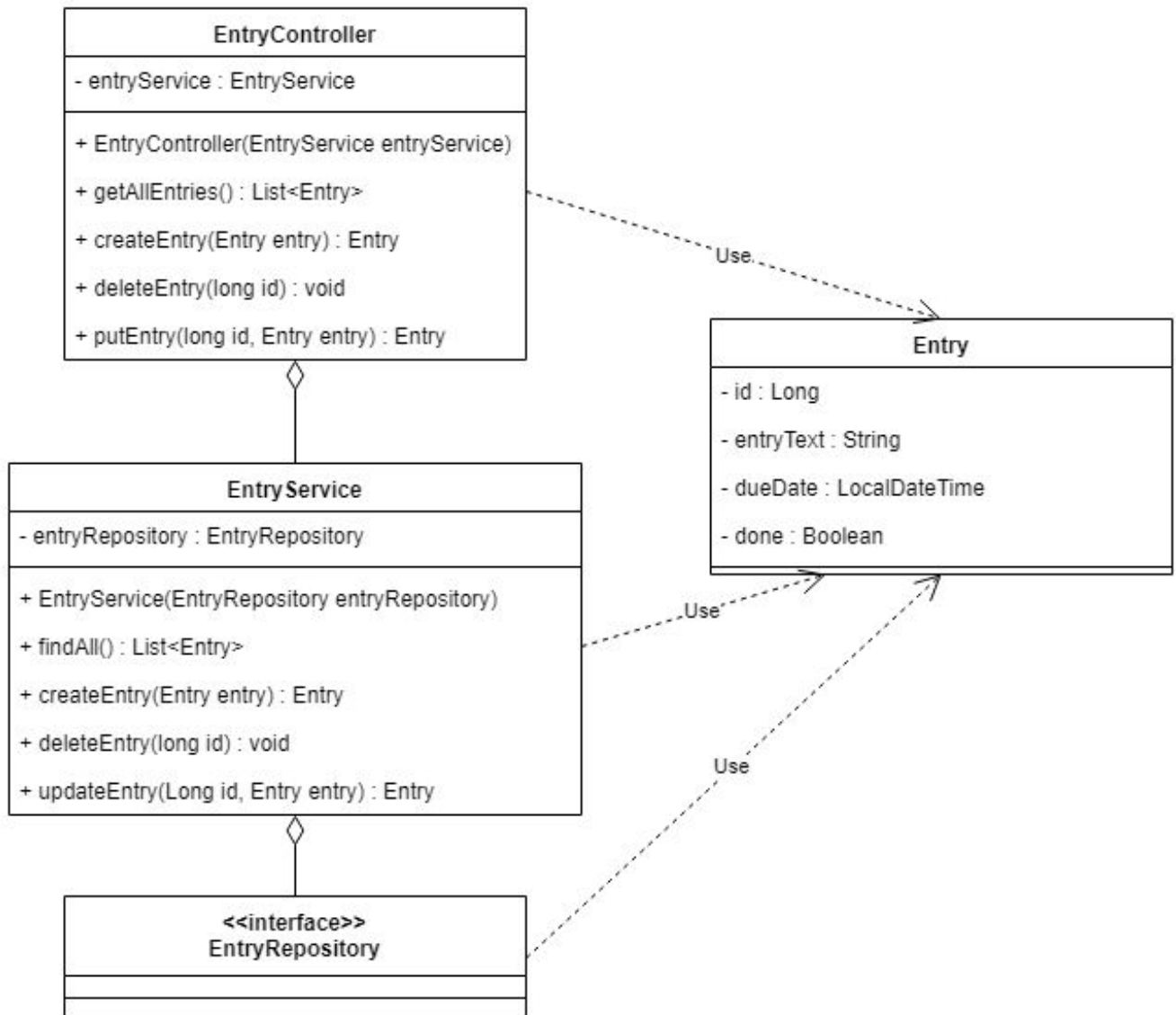


Architektur

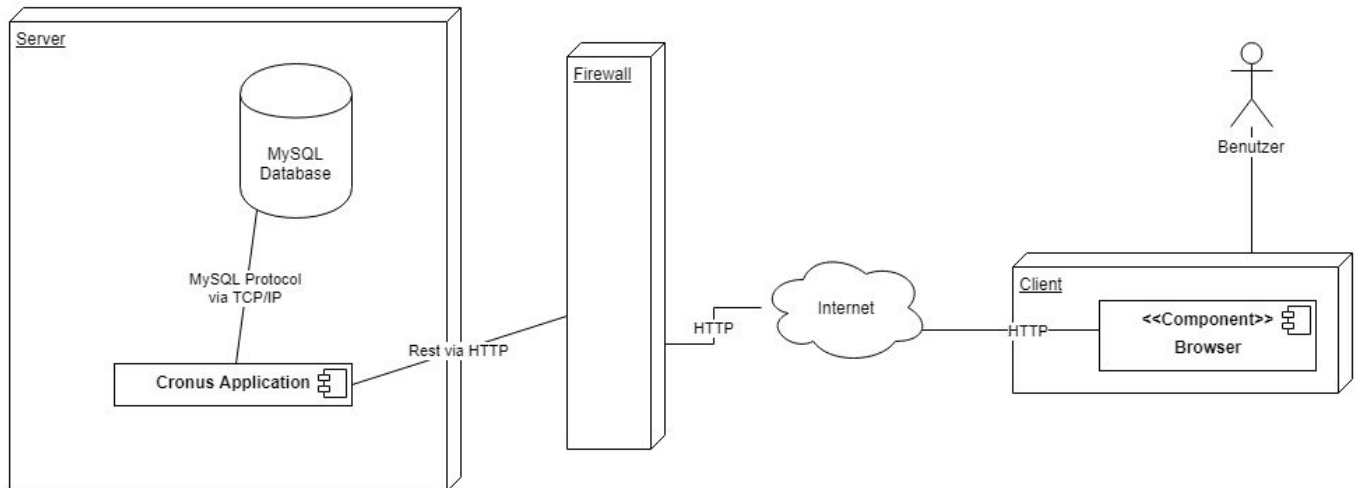
Paketdiagramm



Klassendiagramm



Deployment Diagramm



Testfälle

Anforderungstests

1. Testfall: Ich kann mit einem gültigen Login meine ToDo-Liste ansehen.

Ablauf: Gültige Logindaten abschicken und einen Token erhalten. Danach die dazugehörige Liste aufrufen.

Erwartetes Resultat: Die ToDo-Liste des Benutzers und einen Token soll erhalten werden.

2. Testfall: Ich komme mit einem ungültigen Login an keine Daten.

Ablauf: Ungültige Logindaten abschicken und danach eine Passende Meldung erhalten.

Erwartetes Resultat: Keine Daten und kein Token bekommen, sondern eine passende Meldung.

3. Testfall: Ich kann mich abmelden und bekomme dann keine Daten mehr.

Ablauf: Einen Token haben und die ToDo-Liste abfragen. Danach Abmelden und schauen ob man noch an Daten kommt.

Erwartetes Resultat: Token wird lokal gelöscht und man verliert zugang zu jeglichen Daten.

4. Testfall: Ich kann einen Eintrag in meiner Liste erstellen.

Ablauf: Einen Token haben, die Liste aufrufen und dann einen neuen Eintrag erstellen. Anschliessend Liste nochmals aufrufen um dies zu überprüfen.

Erwartetes Resultat: Der neue Eintrag ist nun in der Liste.

5. Testfall: Ich kann einen Eintrag in meiner Liste bearbeiten.

Ablauf: Einen Token haben, die Liste aufrufen und dann einen Eintrag selektieren und ändern. Anschliessend Liste nochmals aufrufen um dies zu überprüfen.

Erwartetes Resultat: Der selektierte Eintrag wurde verändert.

6. Testfall: Ich kann einen Eintrag in meiner Liste löschen.

Ablauf: Einen Token haben, die Liste aufrufen und dann einen Eintrag selektieren und löschen. Anschliessend Liste nochmals aufrufen um dies zu überprüfen.

Erwartetes Resultat: Der selektierte Eintrag wurde gelöscht.

Nicht-Funktionale Tests

1. Die Seite sieht bei Opera, Chrome und Firefox genau gleich aus.
2. Ich kann die Applikation auf Windows benutzen.
3. Ich erhalte eine Antwort in bis zu 10 Sekunden.