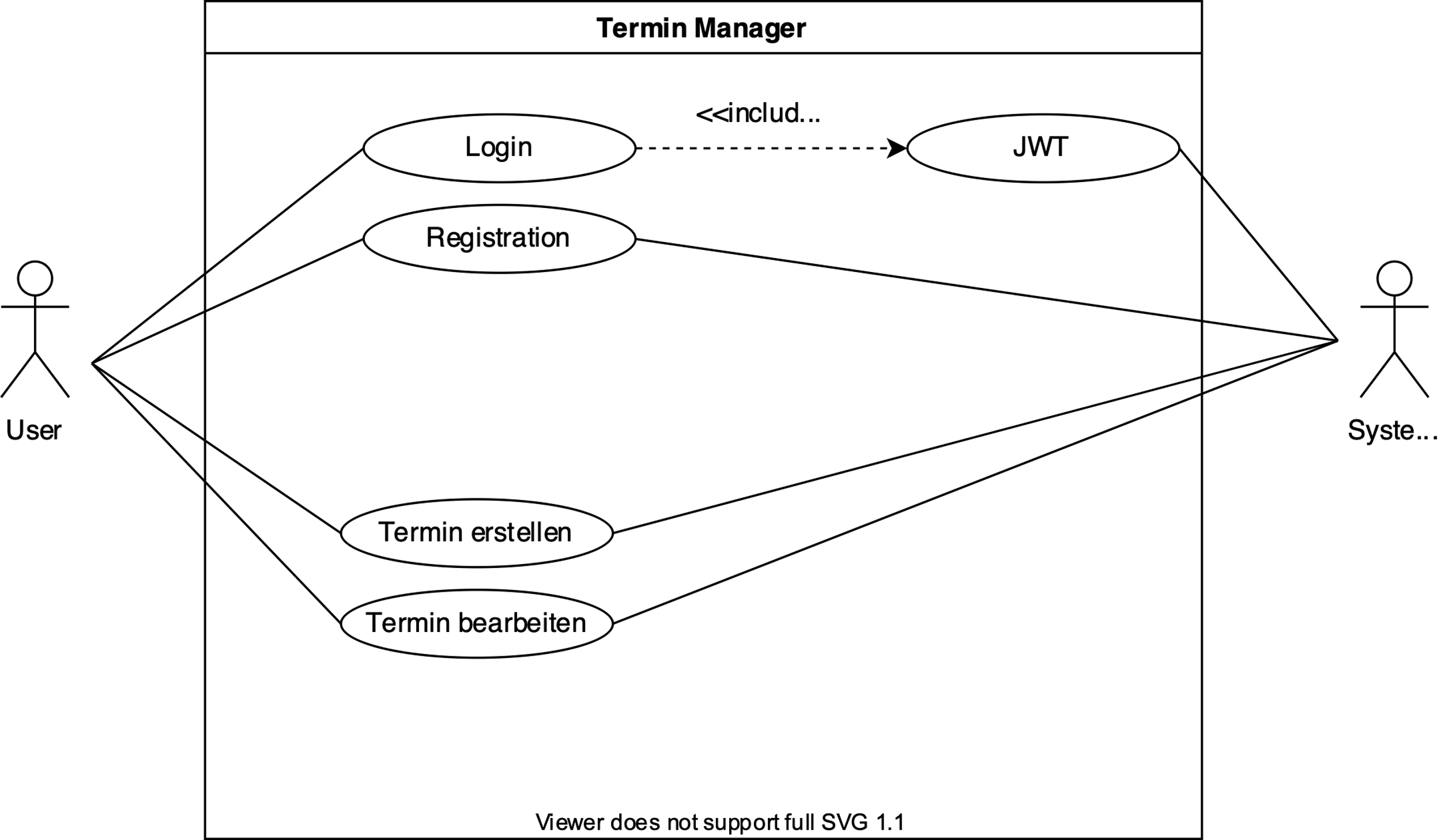
M223 Bank Manager

In diesem Projekt kann der User sich registrieren und einloggen per JWT. Der User kann danach selbst Termine erstellen und diese bearbeiten.

Use Case Diagramm:



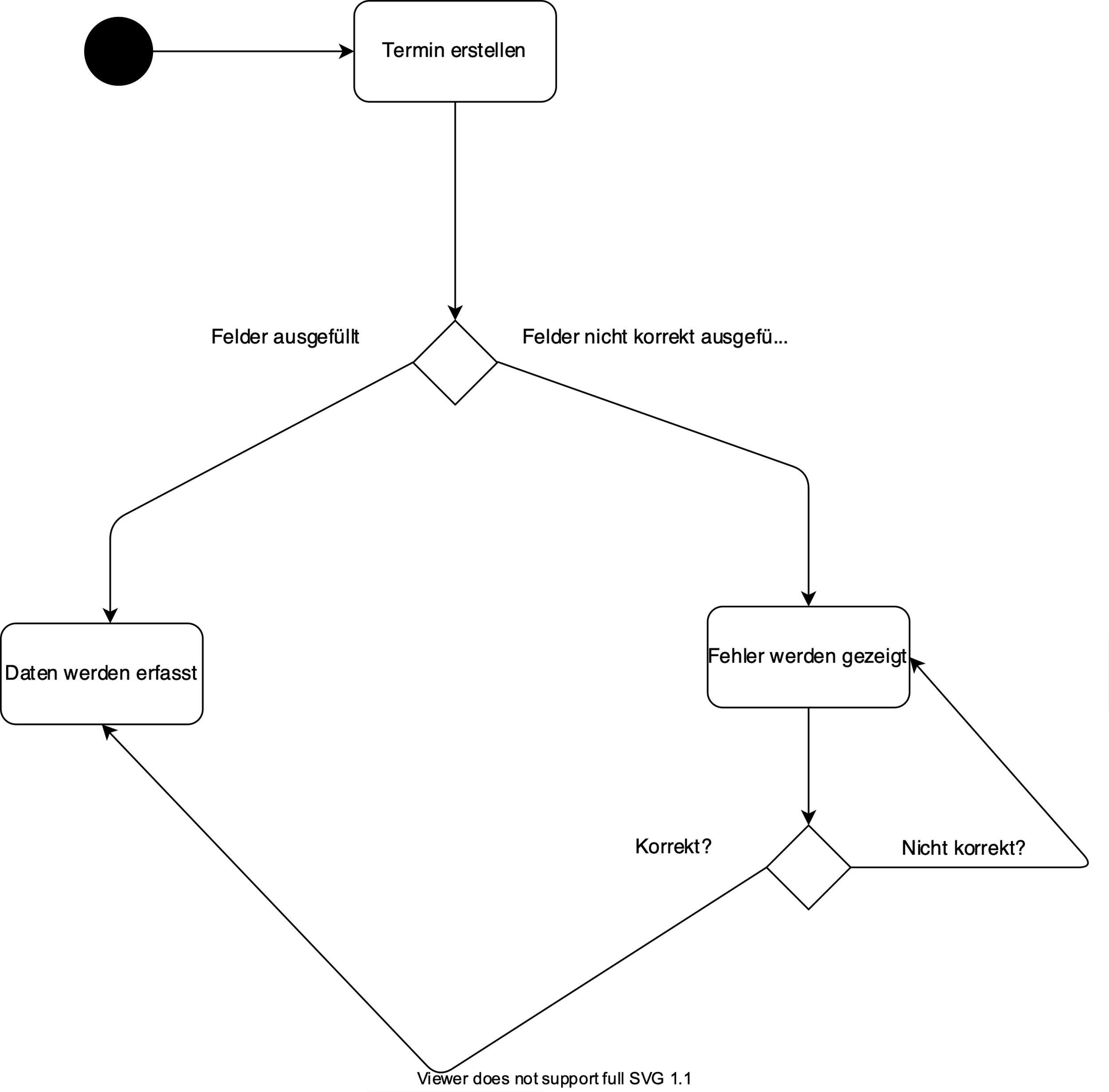
Registration: Der User kann sich registrieren mit Benutzername, E-Mail und Passwort.

Login: Nach dem Registrieren, darf sich der User einloggen mit seinen Daten und erhält einen temporären JWT-Token.

Termin erstellen: Der User darf einen Termin erstellen mit Titel, Ort, Datum.

Termin bearbeiten: Der User kann den Termin bearbeiten.

Aktivitätsdiagramm zu einer Anforderung:

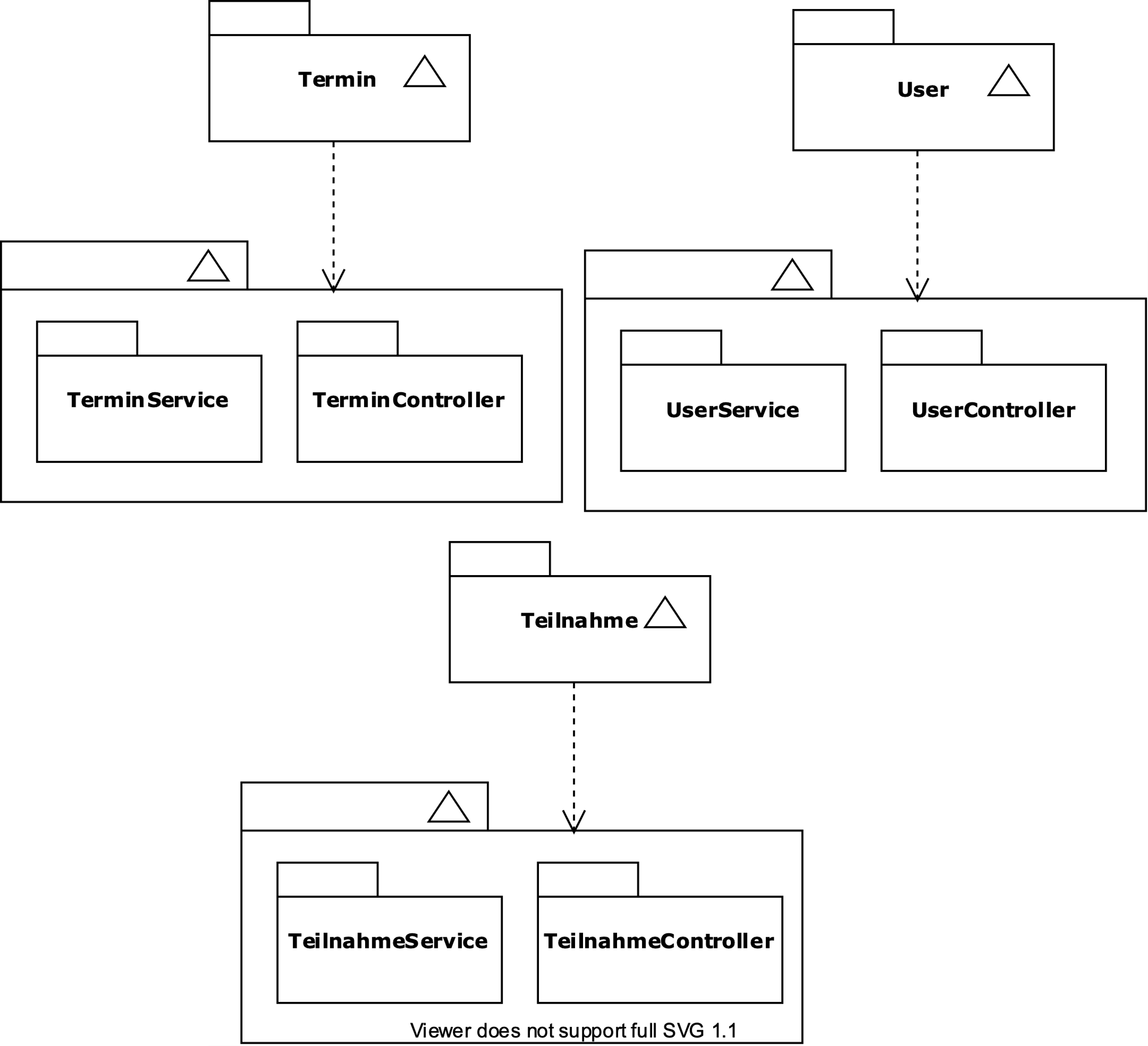


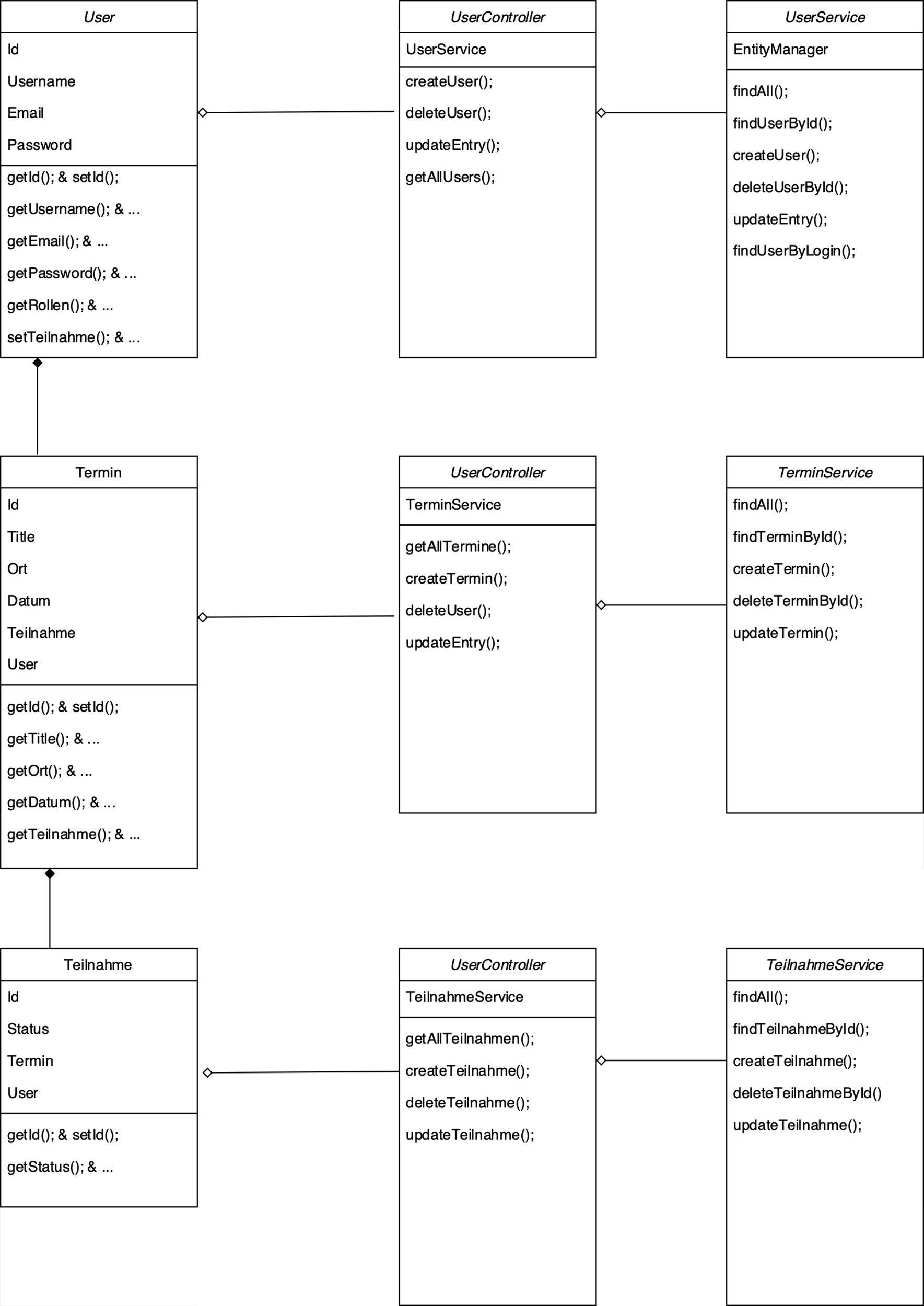
Im folgenden Diagramm wird aufgezeigt, wie man ein Termin erstellt. Falls die Daten fürs Erstellen nicht richtig abgefüllt wurden, werden Fehler angezeigt. Falls die Daten korrekt abgefüllt wurden, können diese in die Datenbank abgefüllt werden.

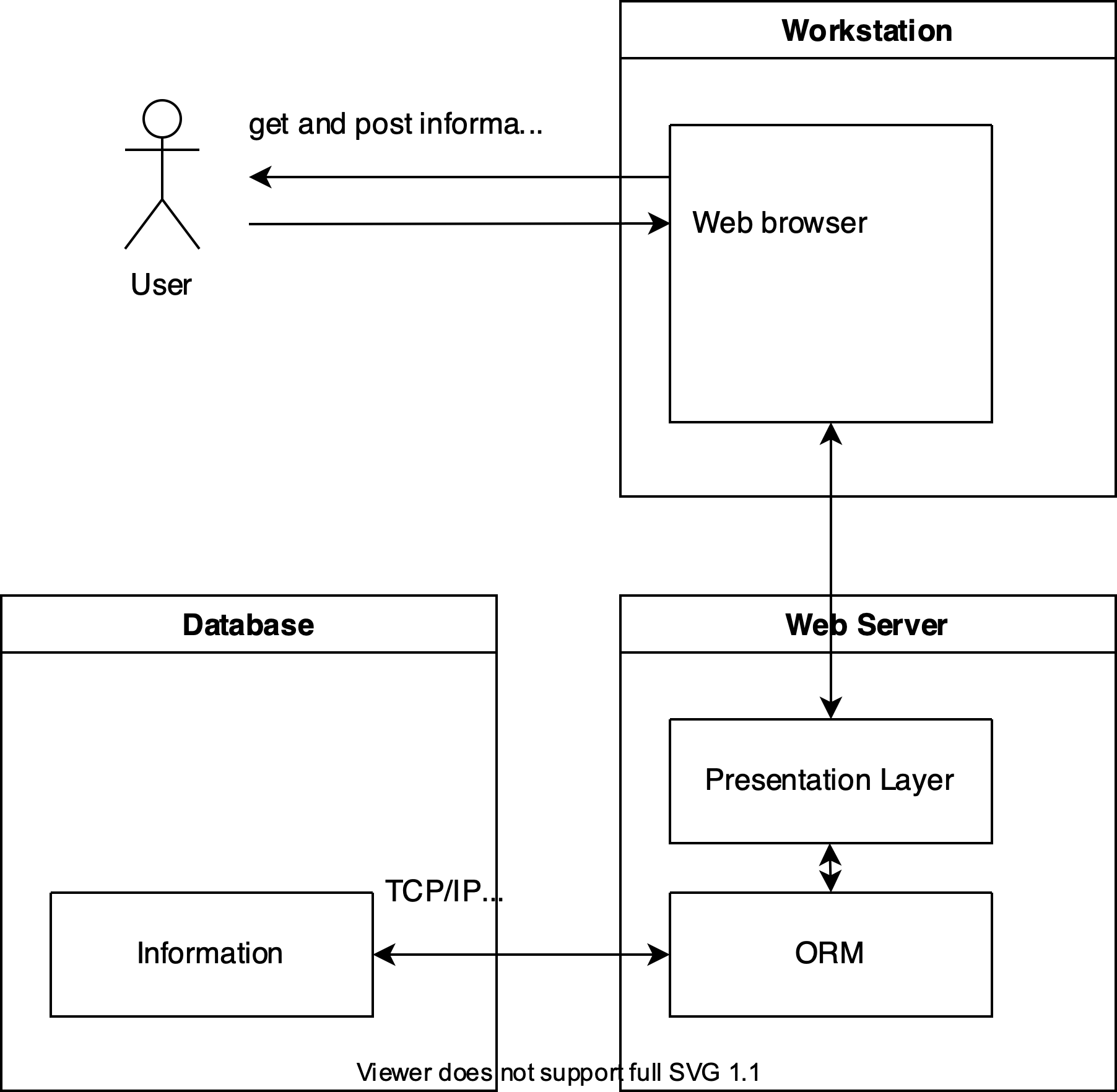
Nicht-funktionale Anforderungen:

* Zeitverhalten: Das System lädt nie unter 3s das Ergebnis. Das bedeutet, dass das System sehr performant programmiert wurde.
* Robustheit: Das System kann im Dauerbetrieb ohne Neustart betrieben werden. Ausser das System muss Wartungsarbeiten durchführen, dann kann es nicht laufen.
* IT-Sicherheit: Das Systemkann nur mit gültigem JWA-Token gebraucht werden. Es gibt keine Möglichkeit das System zu bedienen ohne JWA-Token.
* Die Datenbank kann nicht mit fehlerhaften Daten abgefüllt werden. Das System verfügt über Transaktionen

Package Diagramm:



Klassendiagramm: 

Deployment Diagram:

Tests zu den funktionalen Anforderungen:

1. Wrong Input for User Email Test:

Das System wird geprüft, indem man eine ungültige E-Mail eingibt. Es wird geschaut, ob ein falscher E-Mail-Syntax akzeptiert wird.

1. SignUp Button betätigen
2. Die Felder gewünscht abfüllen
3. Im E-Mail-Feld komischer Syntax testen. (s\_»+»\*@a.as.d..com)
4. Auf Fehler Ausschau halten
5. Input Test for User Username:

Das System wird geprüft, welche Chars für den Username akzeptiert werden. Beispiele sind: «+\*ç%&/()=»

1. SignUp Button betätigen
2. Die Felder gewünscht abfüllen
3. Im Username-Feld komischer Syntax testen. ($$\_£\_!èààé£é£àé»+»\*)
4. Auf Fehler Ausschau halten
5. Auth Test:

Das System wird geprüft, ob man auf Unterseiten kommt, ohne sich angemeldet zu haben. Beispiel: Ohne Login auf «/Teilnahmen»

1. Statt auf den SignIn/SignUp Button zu klicken in der URL auf /Termine switchen.
2. Auf Fehler Ausschau halten
3. Input Test for User Password

Das System wird geprüft, ob das Password verschiedene Anforderungen hat. Damit kann sichergestellt werden, ob der User nicht ein zu einfaches Passwort wählen darf.

1. SignUp Button betätigen
2. Die Felder gewünscht abfüllen
3. Im Password-Feld komischer Syntax testen. (a)
4. Überprüfen, ob Hinweise erscheinen, dass man ein längeres Passwort wählen muss.
5. View Test for Termine

Das System wird geprüft, ob man nur seine eigenen Termine sieht und nicht allenfalls noch Termine von anderen Usern.

1. SignIn Button betätigen
2. Unter /Termine/new einen Termin erstellen
3. Unter /Termine überprüfen, ob nur selbst erstellte Termine angezeigt werden.

Tests zu den nicht-funktionalen Anforderungen:

1. Termin Flooder Test

Das System wird geprüft, ob das System überlastet, wenn sehr viele Termine erstellt werden.

1. Unter /Termine/new den Erstellen Knopf spammen und schauen ob alle Einträge auch erstellt werden.
2. Live-Time Test

Das System wird geprüft, ob das System nach längerer Nutzung den Geist aufgibt. Dies sollte nicht auftreten. (Sollte selbst erklärend sein)

1. JWA Test

Das System wird geprüft, wie lange der JWA-Token gültig ist. Die Zeit sollte nicht über 10 Minuten liegen. (Sollte selbst erklärend sein)