

Erstklausur

Softwaretechnik II

Wintersemester 2022/23

Prof. Dr. Ralf H. Reussner

20.02.2023

Name: _____

Matrikelnummer: _____

Zur Klausur sind keine Hilfsmittel und kein eigenes Papier zugelassen. Die Bearbeitungszeit beträgt 90 Minuten. Die Klausur ist vollständig und geheftet abzugeben. Verwenden Sie ausschließlich dokumentenechte Stifte. Mit Bleistift oder roter Farbe geschriebene Angaben werden nicht bewertet. Lösen Sie die Aufgaben in leserlicher Schrift. Unlesbare Lösungen können naturgemäß nicht positiv bewertet werden. Wenn das generische Maskulinum gewählt wurde, geschieht dies zur besseren Lesbarkeit und zum einfachen Verständnis der Aufgabenstellung. Sofern nicht anders angegeben, beziehen sich Angaben im Sinne der Gleichbehandlung auf Vertretende aller Geschlechter.

Aufgabe	1	2	3	4	5	6	Σ
Maximal	10	16	18	14	20	12	90
K1							
K2							
K3							

Name:

Matrikelnummer:

Aufgabe 1: Wissensfragen (10 Punkte)

- a) Nennen Sie die drei Architekturmuster für die Geschäftslogik (*Domain Logic Patterns*) von Unternehmensanwendungen aus der Vorlesung. (1,5 Punkte)

- b) Nennen Sie stichpunktartig zwei Hauptcharakteristiken von *Infrastructure as a Service* Cloud-Anwendungen aus der Vorlesung. (2 Punkte)

- c) Nennen Sie vier der sechs Interaktionen zwischen abgeschlossenen Kontexten (*Bounded Contexts*) aus der Vorlesung zum Domänen getriebenen Entwurf. (2 Punkte)

Name:

Matrikelnummer:

- d) Nennen Sie stichpunktartig zwei zentrale Probleme des Wasserfallmodells aus der Vorlesung. (2 Punkte)

- e) Handelt es sich bei dem im folgenden beschriebenen Echtzeitsystem um ein Monitoring- (*monitoring*) oder Kontrollsystem (*control system*)? Begründen Sie Ihre Antwort kurz. (1 Punkt)

Das Dreiachsenstabilisierungssystem eines geostationären Kommunikationssatelliten überwacht die Ausrichtung des Satelliten mit einem Gyroskope und sendet kontinuierlich Steuersignale an 3 orthogonale Schwungräder, um jeder Ausrichtungsabweichung des Satelliten sofort entgegen zu wirken.

- f) Nennen sie drei Ziele von Software-Sicherheit aus der Vorlesung. (1,5 Punkte)

Aufgabe 2: Anforderungserhebung (16 Punkte)

Im Folgenden betrachten wir die Entwicklung eines Tracking-Systems für Bisons im Namen der Tierschutzorganisation *Phi's Sanctuary*.

Das Bison-Tracking-System (BTS) soll für Wissenschaftler und Mitglieder online eine länderübergreifende Plattform zur Beobachtung von Bisonpopulationen in Europa bereitstellen. Erfasst werden sollen die Lebensräume wilder Populationen, ihr Zugverhalten und Gruppengröße. Sichtungen sollen von allen Personen (auch Nichtmitgliedern) im Tracking-System erfasst werden können. Das Zugverhalten soll basierend auf Sichtungen und Gruppengröße berechnet werden. Um zu vermeiden, dass Bison-Gruppen böswillig aufgescheucht werden, sollen bestimmte Funktionen, wie Zugverhalten, genaue Lokalisierung und weitere Analysen ausschließlich verifizierten Wissenschaftlern bereitgestellt werden.

Während der Anforderungserhebung wurden für das Bison-Tracking-System unter anderem die folgenden Anforderungen identifiziert:

Nr. Anforderung

- /R1/ *Das BTS muss alle Datenschutzrichtlinien der teilnehmenden Länder einhalten.*
 - /R2/ *Das BTS stellt aufbereitete Infografiken für Museen zur Aufklärung bereit.*
 - /R3/ *Das BTS speichert das Datum, die Uhrzeit und Ort der Sichtung.*
 - /R4/ *Sobald eine Sichtung eingetragen wird, überprüft das BTS die Sichtung auf Plausibilität.*
 - /R5/ *Wenn ein Mitglied sich anmeldet wird sein Mitgliedsstatus geprüft.*
 - /R6/ *Zu jedem Account wird ein Vorname, Nachname, Passwort und Status, genauer Mitglied, Wissenschaftler oder Standard, gespeichert.*
 - /R7/ *Das BTS sollte sich intuitiv bedienen lassen.*
 - /R8/ *Das BTS muss sich durch Plugins erweitern lassen.*
-

- a) Nennen Sie zwei weitere Stakeholder, die neben Wissenschaftlern und Mitgliedern, für die Entwicklung des Tracking-Systems relevant sind. (1 Punkte)

Name:

Matrikelnummer:

- b) Klassifizieren Sie die oben stehenden Anforderungen nach der in der Vorlesung behandelten Klassifikation von Glinz. Ordnen Sie hierzu die acht Anforderungen des Bison-Tracking-Systems den in der Vorlesung genannten Anforderungsarten (*kind: functional, quality, constraint*), deren Unterkategorie (*subkind: functions, data, performance, security, availability, legal, ...*) sowie deren Repräsentation (*representation: operational, quantitative, qualitative, declarative*) zu. (12 Punkte)

Nr.	Anforderung	Art	Unterkategorie	Repräsentation
/R1/	<i>Einhalten von Datenschutz</i>			
/R2/	<i>Bereitstellen von Infografiken</i>			
/R3/	<i>Speicherung von Sichtungsdaten</i>			
/R4/	<i>Plausibilität von Sichtungen</i>			
/R5/	<i>Anmeldung von Mitgliedern</i>			
/R6/	<i>Speicherung von Nutzerdaten</i>			
/R7/	<i>Intuitive Bedienung</i>			
/R8/	<i>Erweiterung durch Plugins</i>			

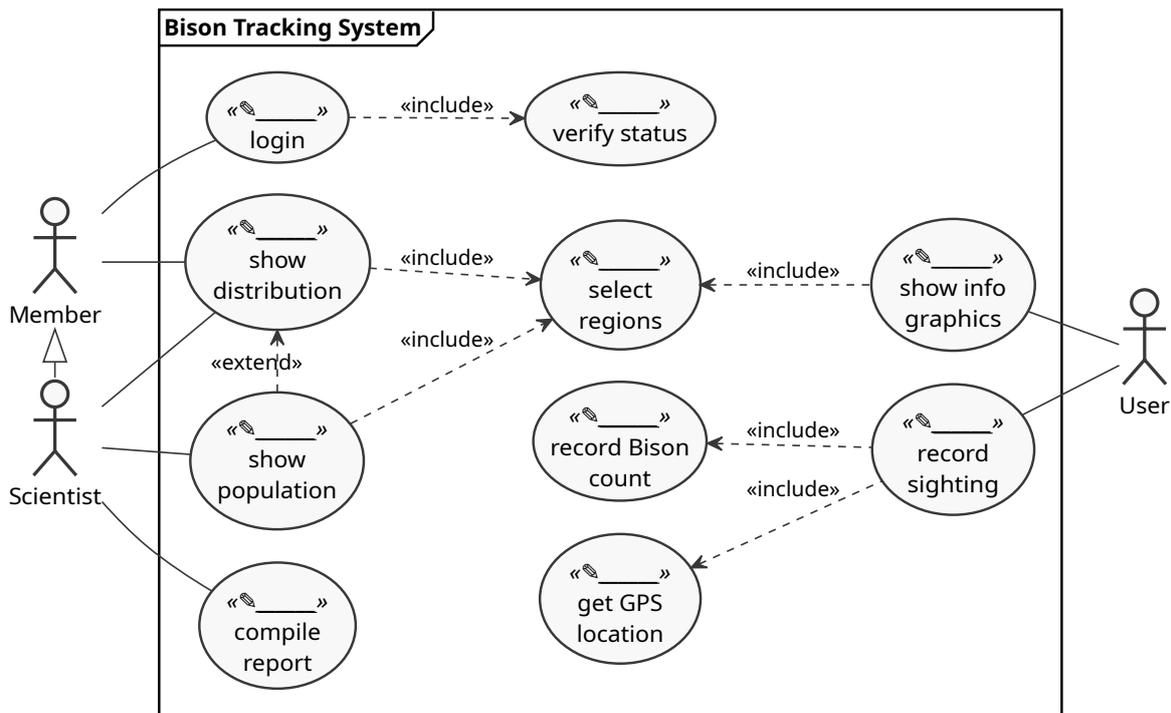
Name:

Matrikelnummer:

- c) Einige der oben stehenden Anforderungen verstoßen gegen die Empfehlungen für gut geschriebene Anforderungen (*basic writing recommendations*) aus der Vorlesung. Finden Sie **drei** Anforderungen die gegen **unterschiedliche** Richtlinien verstoßen. Nennen Sie jeweils die Nummer der Anforderung und die Richtlinie, gegen die verstoßen wurde. (3 Punkte)

Aufgabe 3: Anwendungsfallbeschreibung (18 Punkte)

Diese Aufgabe betrachtet die Anwendungsfälle der Nutzer (*user*), Mitglieder (*members*) und Forscher (*scientist*) des Bison-Tracking-Systems. Diese sind im folgenden Anwendungsfalldiagramm zusammengefasst und in der nachfolgenden Tabelle kurz erläutert.



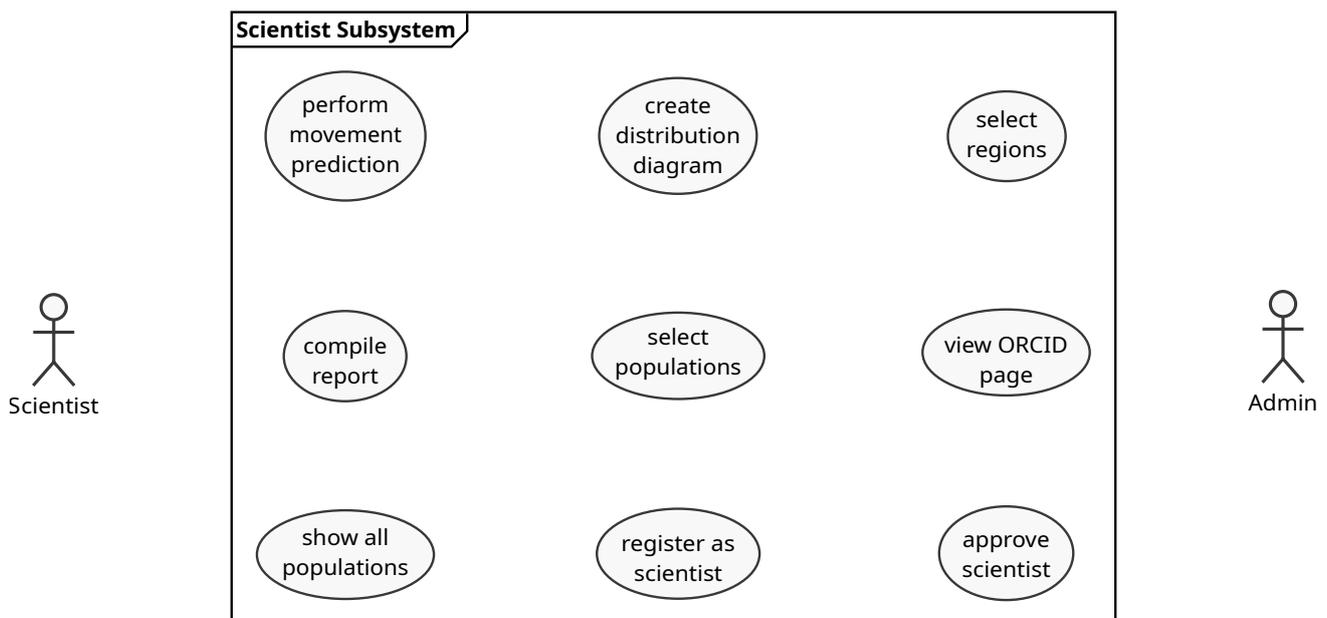
Anwendungsfall	Beschreibung
<i>login</i>	Anmelden des Nutzers im Trackingsystem
<i>show distribution</i>	Anzeigen der Verbreitung aller Bisonpopulationen ausgewählter Regionen
<i>show population</i>	Anzeigen einer Population in ausgewählten Regionen
<i>compile report</i>	Erstellen eines wissenschaftlichen Berichts
<i>verify status</i>	Automatisches Verifizieren des Status eines eingeloggten Nutzers
<i>select regions</i>	Auswählen der Regionen durch den Nutzer
<i>record Bison count</i>	Eintragung der Anzahl beobachteter Bisons durch den Nutzer
<i>get GPS location</i>	Beziehen der aktuellen GPS-Position vom Smartphone des Nutzers
<i>show info graphics</i>	Darstellen der gewählten Informationen in einer Infografik
<i>record sighting</i>	Registrierung einer Sichtung durch den Nutzer

- a) Klassifizieren Sie jeden Anwendungsfall im obigen Anwendungsfalldiagramm (use case diagram) entweder als Nutzerziel (*user goal*) mit **U**, als Teilfunktion (*subfunction*) mit **S** oder als Operation (*operation*) mit **O**. Tragen Sie hierzu die entsprechende Klassifikation in die mit “☞” markierten Bereiche der Anwendungsfälle ein. (5 Punkte)

Da diese Anwendungsfälle nicht detailliert genug sind, ist es Ihre Aufgabe ein weiteres Anwendungsfalldiagramm zu erstellen. Hierzu charakterisiert die folgende Beschreibung wie das Bison-Tracking-System Forschern bei der Erstellung von wissenschaftlichen Berichten hilft:

Das BTS erlaubt es Forschern sich zu registrieren (register as scientist). Die Registrierung wird von einem angemeldeten Administrator bestätigt (approve scientist), der hierzu, sofern er das möchte, die ORCID-Seite des Forschers aufrufen kann (view ORCID page). Angemeldete Forscher können mit dem BTS einen wissenschaftlichen Bericht erstellen (compile report). Hierzu wählt er zuerst die gewünschten Regionen aus (select regions). Danach wählt der Forscher die zu betrachtenden Bisonpopulationen aus (select populations). Falls eine der gesuchten Populationen unbekannt ist, so kann er sich hier, über eine Schaltfläche, eine Gesamtübersicht der gesichteten Bisonpopulationen anzeigen lassen (show all populations). Falls die Verbreitungsinformationen im Bericht nicht ausreichen, bietet das BTS dem Forscher bei Erstellung des Berichts die zusätzliche Möglichkeit eine Bewegungsvorhersage zu berechnen (perform movement prediction). Nach Auswahl der Regionen und Populationen während der Berichterstellung wird dem Forscher das Verbreitungsdiagramm erstellt (create distribution diagram). Am Ende erzeugt das BTS den wissenschaftlichen Bericht. Um über die aktuelle Bisonentwicklung zu informieren, werden die für den Bericht erstellten Infografiken allen Interessierten öffentlich bereitgestellt.

- b) Vervollständigen Sie nun das Anwendungsfalldiagramm (*use case diagram*) genau so, dass es die Anforderung im obigen Text beschreibt. Zeichnen Sie ausschließlich die Beziehungen von Aktoren (*actors*) zu Anwendungsfällen (*use cases*) und zwischen Anwendungsfällen (*use cases*) ein, die im Text beschrieben werden. Zeichnen Sie keine weiteren Anwendungsfälle (*use cases*) oder Aktoren (*actors*) ein. (8 Punkte)



Name:

Matrikelnummer:

- c) Verwenden Sie das textuelle Anwendungsfallbeschreibungsschema *Fully-Dressed* zur vollständigen Spezifikation des Erfolgsfalls des Erstellens eines wissenschaftlichen Berichts (*compile report*). Nutzen Sie hierfür die unten gegebene Schablone. Nennen Sie einen weiteren Stakeholder (außer Forscher) und geben Sie stichpunktartig dessen Interesse am Anwendungsfall an. Geben Sie zusätzlich mindestens eine Erweiterung an. (5 Punkte)

Name d. Anwendungsfalls

(Use Case Name)

Erstellen eines wiss. Berichts

(compile report)

Zielebene

(Goal Level)

Nutzerziel (User Goal)

Primäraktor(en)

(Primary Actor(s))

Forscher

Stakeholder(s)

Vorbedingungen

(Preconditions)

Nachbedingungen

(Postconditions)

Primäres Erfolgsszenario

(Main Success Scenario)

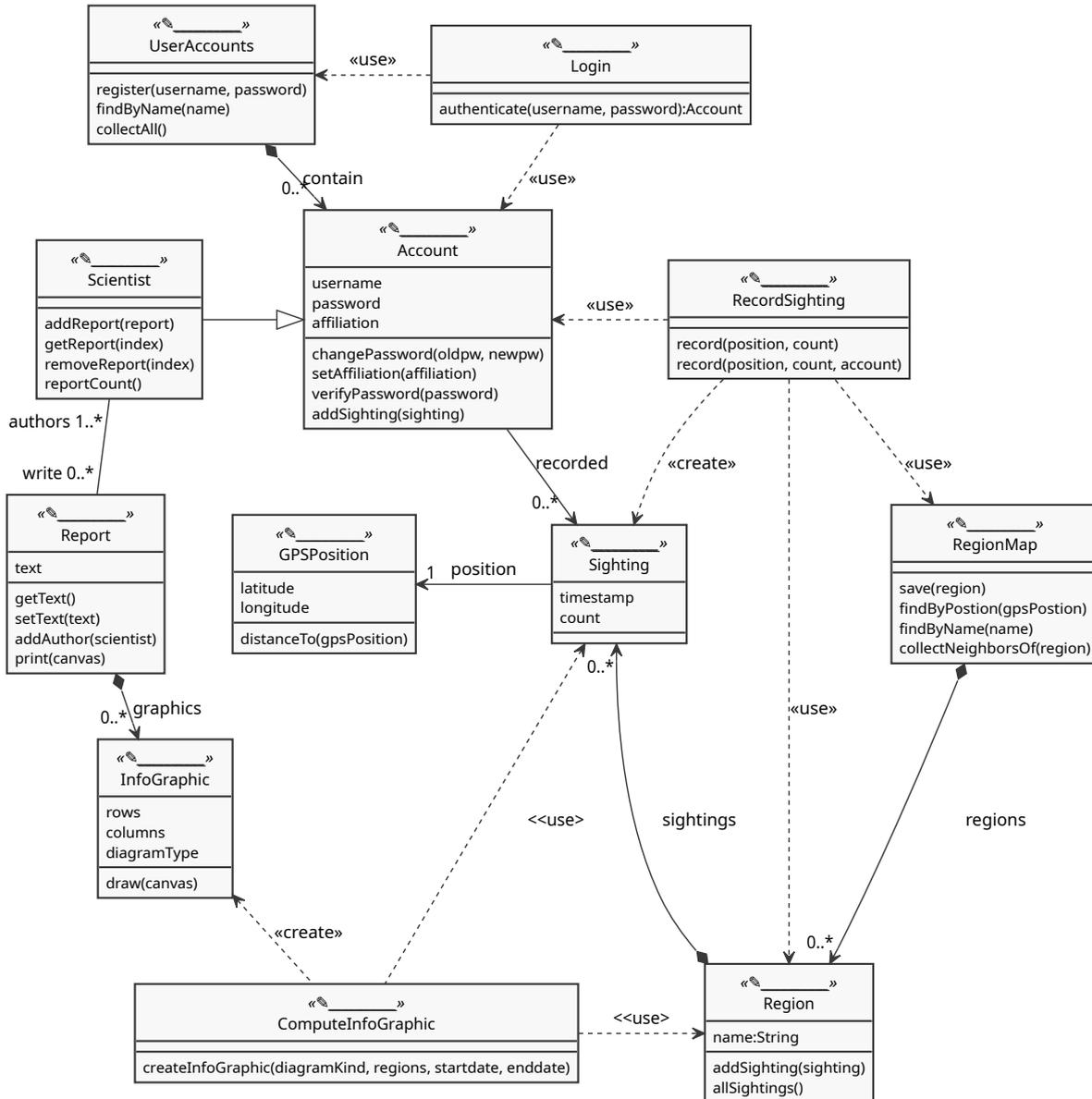
Erweiterung (Weiterer Ablauf)

(Extension; Additional Flow)

Aufgabe 4: Domänen-getriebener Entwurf (14 Punkte)

Der folgende Aufgabenkomplex widmet sich der Anwendung des Domänen-getriebenen Entwurfs für das Bison-Tracking-Systems (BTS). Dieses erlaubt es Nutzern und Mitgliedern Sichtungen von Bisons zu melden (*RecordSighting*) sowie Informationsgraphiken (*ComputeInfoGraphic*) zu erzeugen. Zusätzlich wird es Mitgliedern ermöglicht sich im BTS anzumelden.

a) Durch die Domänenanalyse des Trackingsystems ist das folgende Klassendiagramm entstanden. Klassifizieren Sie jedes Domänenkonzept entsprechend der Bausteine (building blocks) des Domänen-getriebenen Entwurfs entweder als ValueObject mit V, Entity mit E, Service mit S, Factory mit F oder Repository mit R. Tragen Sie hierzu die Klassifikation in die mit “☞” markierten Bereiche der Domänenklassen ein. (6 Punkte)



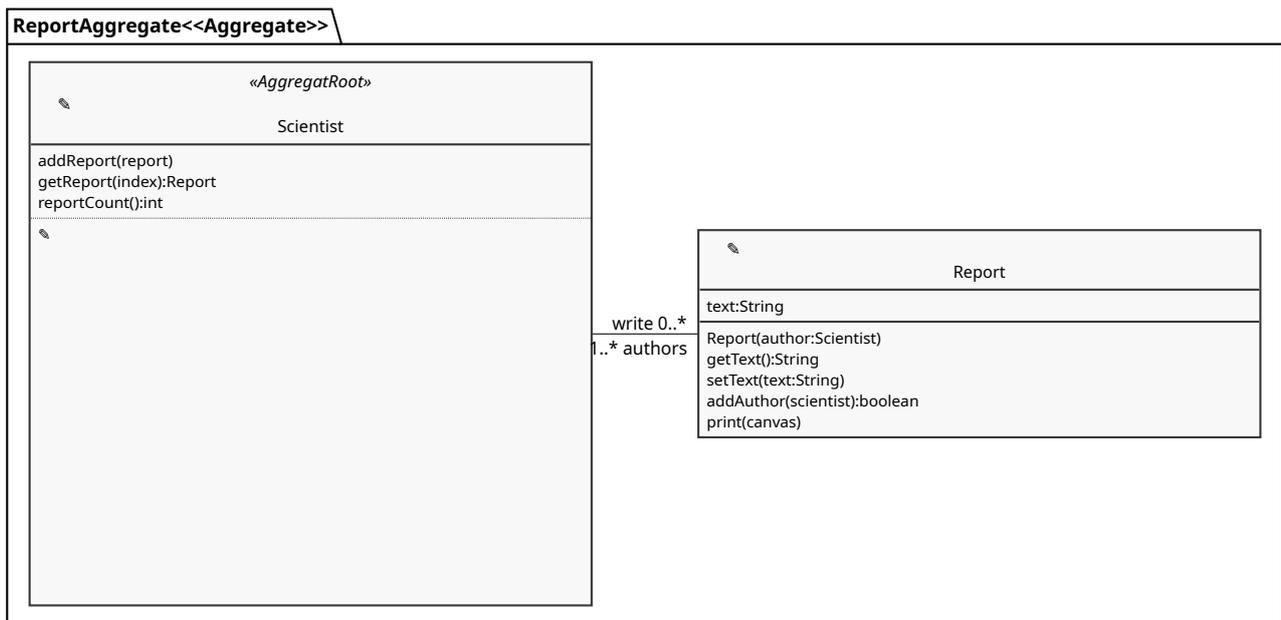
Name:

Matrikelnummer:

- b) Nennen Sie in zwei Stichpunkten Gründe für Ihre Klassifikation der Domänenklasse InfoGraphic. (1 Punkt)

- c) Entwerfen Sie ein *Aggregat* für die Forscher (Scientist), welches sicherstellt, dass Autoren nur die eigenen wiss. Berichte (Report) einsehen und ändern können. Hierzu wurden die beiden Klassen bereits in das ReportAggregate-Paket hinzugefügt. Streichen Sie alle Methoden der Scientist-Klasse durch, die einen direkten Zugriff auf die Report-Klasse erlauben und ersetzen Sie diese durch equivalenten Methoden. Tragen Sie jede zusätzliche Methode in die Scientist-Klasse ein, geben Sie jeweils Namen, die Parameter und den Rückgabotyp an. Tragen Sie weiterhin die Sichtbarkeit der beiden Klassen in die Kopfzeile der Klasse ein: public (+), package private (~), protected (#). (7 Punkte)

Hinweis: Datentypen der Parameter (wie zum Beispiel int, long, Date, String, ...) müssen nicht angegeben werden. Datentypen der Rückgabetypen müssen angegeben werden.

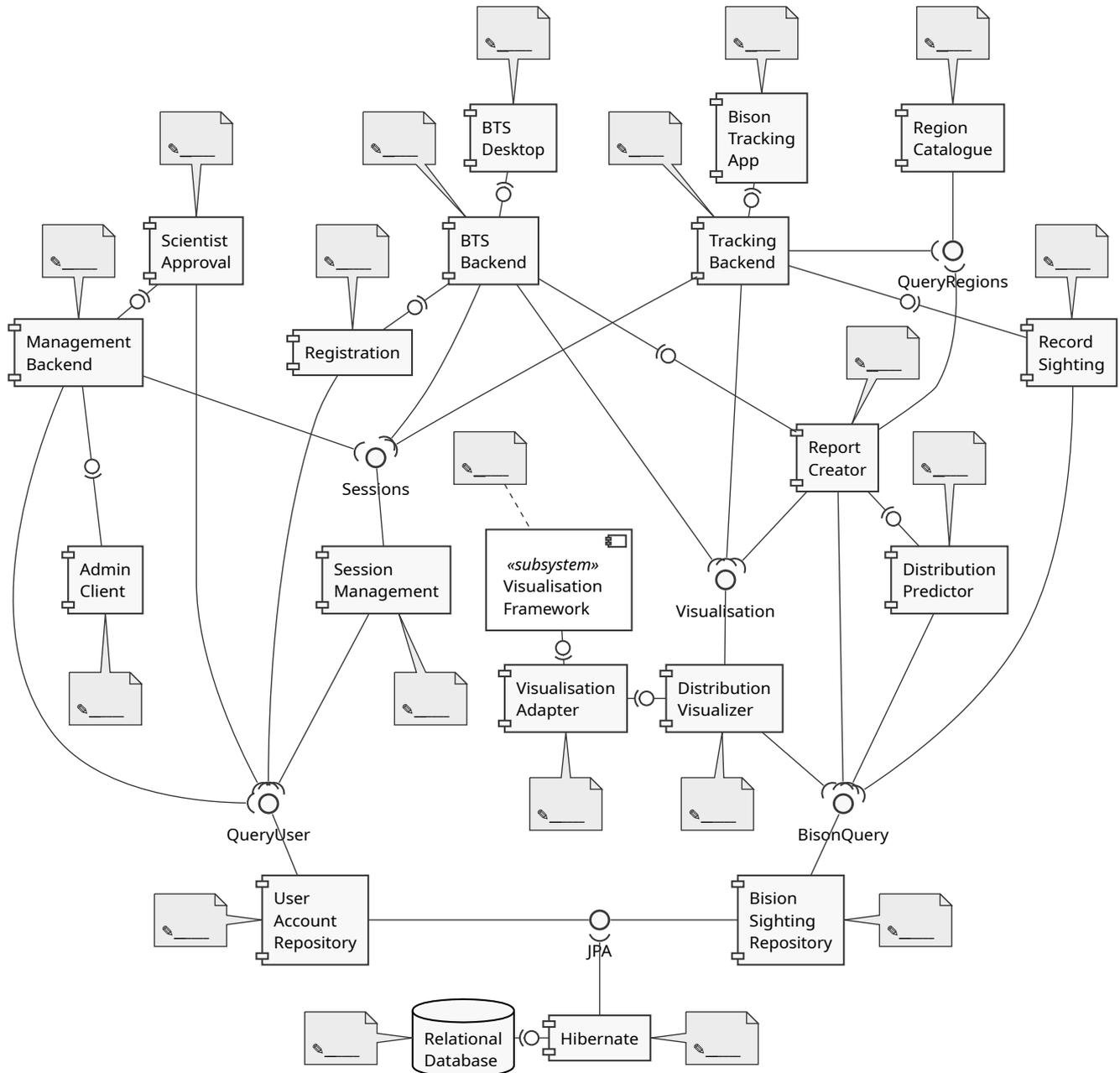


Aufgabe 5: Saubere Unternehmensarchitekturen (20 Punkte)

- a) Tragen Sie in die Tabelle die vier Schichten der *sauberen* Architektur (*clean architecture*) so ein, dass sie zur angegebenen Richtung der Abhängigkeiten passen (*dependency rule*). (2 Punkte)

Schicht	Abhängigkeitsregel	Schichtenname
1.	↓	
2.	↓	
3.	↓	
4.	↓	

b) Das folgende Komponentendiagramm stellt die Softwarearchitektur des Bison-Tracking-Systems dar, welches die Anwendungsfälle: Melden von Bison Sichtungen (*Record Sighting*), Registrieren (*Registration*) und Genehmigen (*Approve*) von Forschern sowie das Erstellen von wissenschaftlichen Berichten (*Report Creator*) und Verteilungsvorhersagen (*Distribution Predictor*) unterstützt. Ordnen Sie nun jede Komponente einer der Schichten der sauberen Architektur zu, indem Sie die Nummer der Schicht in die mit "✎" markierten Bereiche eintragen. Identifizieren Sie **zusätzlich** zwei Abhängigkeiten, die gegen die Abhängigkeitsregel (*dependency rule*) verstoßen. Markieren Sie hierzu die entsprechenden Schnittstellen \bigcirc mit X. (12 Punkte)



Der Quelltext in Listing 1 stellt einen Ausschnitt der Implementierung der Nutzerverwaltung (UserAccounts) im Bison-Tracking-System dar. Genauer werden hier alle angemeldete Nutzer (Account) inklusive Forscher (Scientist) verwaltet. Die hierfür entwickelte Anwendung enthält jedoch einige Verstöße gegen Praktiken des sauberen Programmierens (Clean-Code-Prinzipien und Coding Conventions aus der Vorlesung).

- c) Identifizieren Sie **sechs unterschiedliche** Arten von Verstößen gegen die Praktiken des sauberen Programmierens (Clean-Code-Prinzipien und Coding Conventions aus der Vorlesung) und zugehörige Zeilen im Quelltextbeispiel (nächste Seite). Benennen Sie jeweils die **Art** des Verstoßes mit der zugehörigen **Zeilennummer**. (6 Punkte)

```

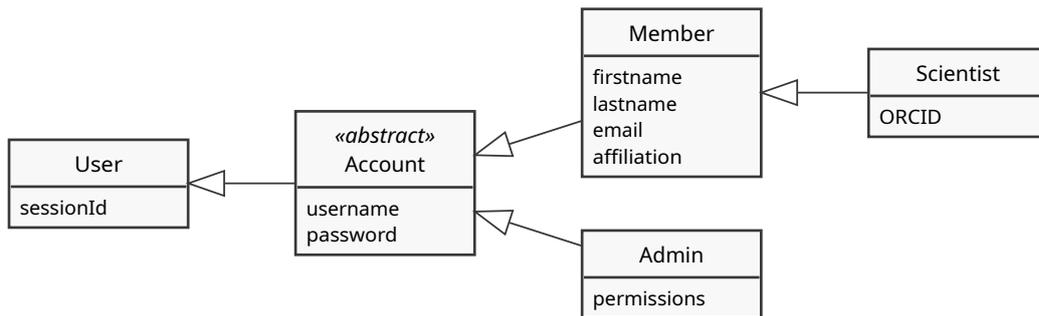
1  public class Account{
2      public final String username;
3      private String password;
4      String affiliation="";
5      public Account(String username,String password){
6          //TODO: Check that username and password is not empty
7          this.username=username;
8          this.password=password;
9      }
10     public boolean changePassword(String oldpw, String newpw){
11         if (newpw!=null && verifyPassword(oldpw)){
12             password=newpw;
13             return true;
14         } else
15             return false;
16     }
17     public void setAffiliation(String affiliation){
18         //if (affiliation!=null && affiliation.isBlank())
19         this.affiliation=affiliation;
20     }
21     public boolean verifyPassword(String password){
22         return password.equals(this.password);
23     }
24     /*...*/}
25     public Scientist extends Account{
26         public Scientist(Sring)
27         public void setAffiliation(String affiliation){
28             if (affiliation!=null && affiliation.isBlank())
29                 this.affiliation="Researcher at "+affiliation;
30         }
31         /*...*/}
32     public class UserAccounts{
33         private HashMap<String, Account> contains = new HashMap<>();
34         public boolean register(String username, String password){
35             return addAccount(new Account(username,password));
36         }
37         public boolean register(String username, String password, int publications){
38             return addAccount(new Scientist(username,password));
39         }
40         private boolean addAccount(Account a){
41             return a!=null && !contains.containsKey(a.username) && contains.put(a.username,a)==null;
42         }
43         public Account findByName(String name){
44             return contains.get(name);
45         }
46         public Scientist findScientistByName(String name){
47             var account=findByName(name);
48             return (account instanceof Scientist s ? s : null);
49         }
50         /*...*/}

```

Listing 1: UserAccounts-Klasse mit Bad Smells

Aufgabe 6: Entwurfsmuster für Unternehmenssoftware (12 Punkte)

Das Domänenmodell des Bison-Tracking-Systems enthält unter anderem die folgende Klassenhierarchie für die unregistrierten Nutzer (*User*) und registrierten Nutzer (*Account*) im System, letztere sind entweder Mitglieder (*Member*), Forscher (*Scientists*) oder Administratoren (*Admin*). Um diese Klassenhierarchie in einer Datenbank abzubilden, sollen Sie in dieser Aufgabe die unterschiedlichen Objekt-Relationalen Strukturmuster (*object-relational structural patterns*), welche in der Vorlesung behandelt wurden, anwenden.



- a) Erstellen Sie ein Datenbankschema nach dem Strukturmuster *Single Table Inheritance*. Erstellen Sie hierzu ein UML-Klassendiagramm der nötigen Tabellen (dargestellt als UML-Klasse mit dem `<<table>>` Stereotyp) und ihrer Attribute. Führen Sie, falls nötig, neue Attribute ein, um *Primärschlüssel* und *Vererbung* darzustellen. (3,5 Punkte)

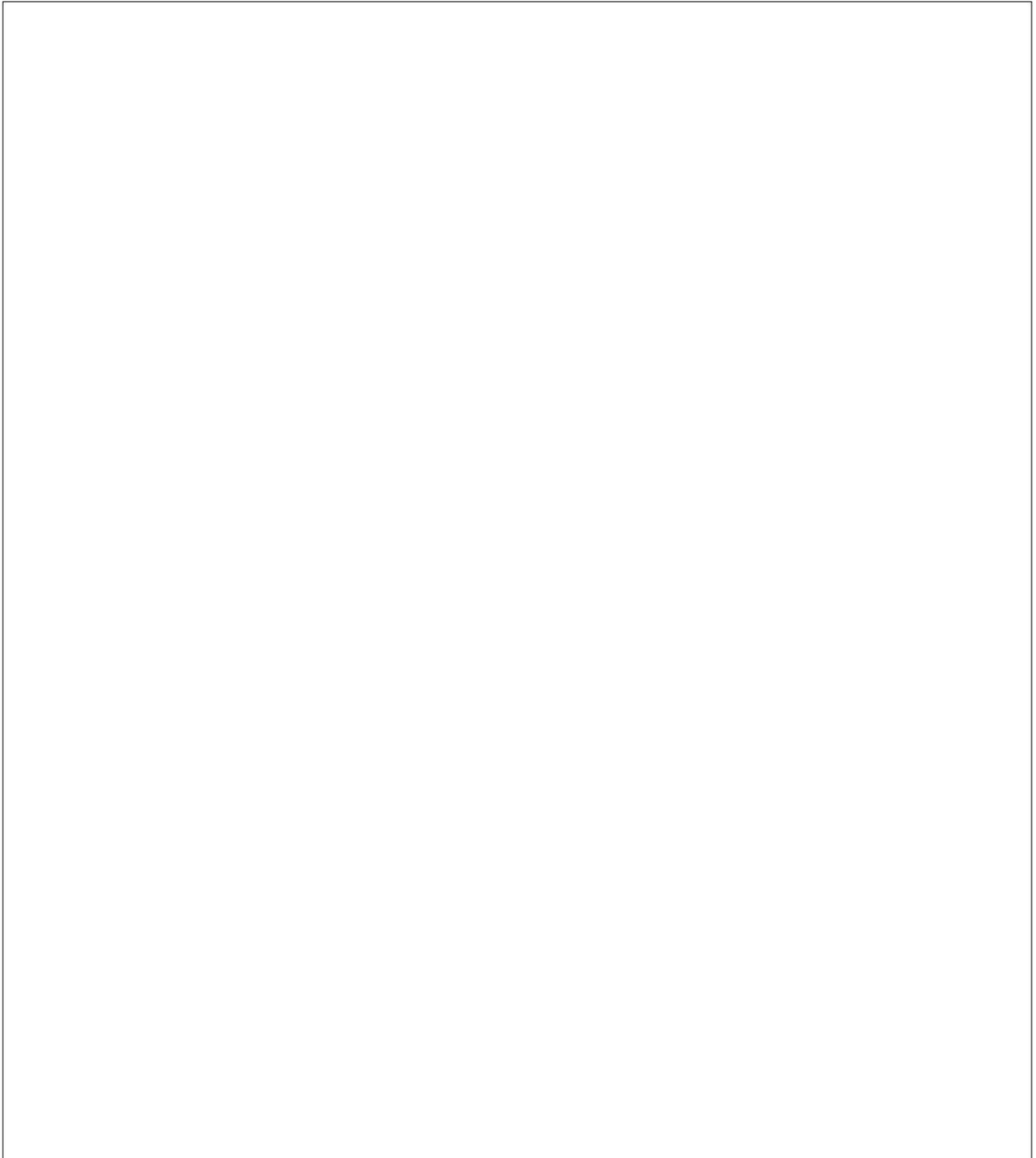
Hinweis: Datentypen (wie zum Beispiel `int`, `long`, `Date`, `String`, ...) müssen nicht angegeben werden.

Name:

Matrikelnummer:

- b) Erstellen Sie ein Datenbankschema nach dem Strukturmuster *Concrete Table Inheritance*. Erstellen Sie hierzu ein UML-Klassendiagramm der nötigen Tabellen (dargestellt als UML-Klasse mit dem <<table>> Stereotyp) und ihrer Attribute. Führen Sie, falls nötig, neue Attribute ein, um die *Primärschlüssel* und *Vererbung* darzustellen. (7,5 Punkte)

Hinweis: Datentypen (wie zum Beispiel int, long, Date, String, ...) müssen nicht angegeben werden.



Name:

Matrikelnummer:

- c) Geben Sie stichpunktartig einen Vorteil und einen Nachteil der *Single Table Inheritance* gegenüber der *Concrete Table Inheritance* an. (1 Punkt)

Name:

Matrikelnummer:

Name:

Matrikelnummer:
