IT - Insurance Team

Projektbeschreibung:

Geplant ist ein Serverprogramm für die Verwaltung der Kundendaten einer Versicherungsagentur. Der Angestellte soll sich dann über ein Login am Server anmelden können (z.B. per Webseite, App oder Desktop-programm -> 4. Semester). Dort sollen dann alle Kundendaten (Allg. Daten und aktuelle Verträge mit allen Infos), Dokumente (Verträge, Beratungsprotokolle, Briefverkehr und Beweismittel für Regulierung) und Schäden eingesehen, bearbeitet, oder auch hinzugefügt bzw. gelöscht werden können.

Ist Zustand:
Die bisherige Kundendatenverwaltung erfolgt über einen externen Server, auf welchem nur die allgemeinen Kundendaten liegen mit einer Auflistung der aktuellen Verträge. Die Verträge an sich sind jedoch in Papierform in Ordnern abgelegt. Daher muss man vor dem Außendienst alle nötigen Dokument heraussuchen und ggf. Kopieren. Der Nachteil daran ist, dass man einen großen zeitlichen Aufwand hat und während eines Kundenbesuchs nicht auf evtl. zusätzlich benötigte Dokumente zugreifen kann.

geplante Verbesserung:
Durch die Serveranwendung mit Datenbank (auf der alle Dokumente eingepflegt werden können) lassen sich mehrer Wege des Zugriffs realisieren und man kann dadurch auch ‘on-the-run’ auf Dokumente zugreifen. Also zum Beispiel im Büro über ein Desktop-programm und unterwegs per Webseite auf dem Laptop oder per App auf dem Handy. Sodass man auch während eines Kundenbesuchs sich ggf. ein benötigtes Dokument vom Server herunterladen und ansehen kann.

Brainstorming:

Funktionen:

- Login und Benutzerkontenverwaltung
- Verwalten von Kunden und Kundendaten (anlegen, bearbeiten, löschen)
- Verwalten von Dokumenten (einpflegen, herunterladen)
- Verwalten von Schäden
- Suchfunktion (von Kunden, und Dokumente eines Kunden)
- vllt zusätzliche Funktionen: Filterfunktion, (Sortierfunktion)

Klassen:

- Programm-controller
- Login
- Benutzer
- Kunden
- Versicherungsklassen
- Dokumente
- Schäden
- Suchen
- vllt: Up-Download-Klasse

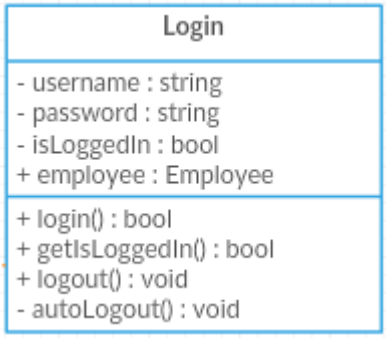
Kommunikation und Verwaltung:

- Google-Dokumente (Projektplanung, Terminkalender)
- Facebook (Kommunikation)
- GitHub (Programmverwaltung)

Klassenbeschreibung:

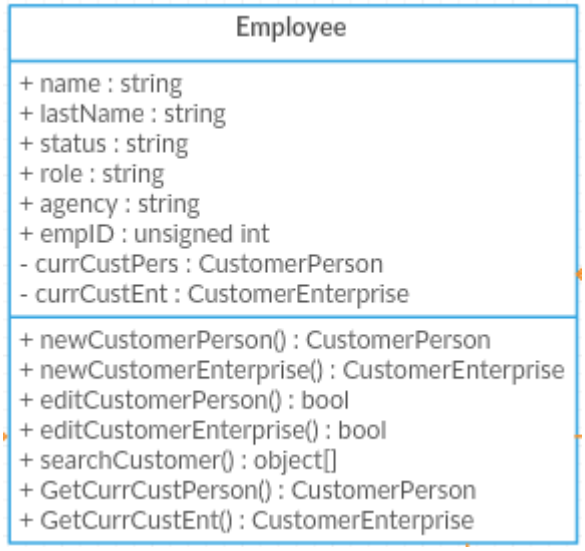
- Login:

  - username, password, isLoggedIn, employee

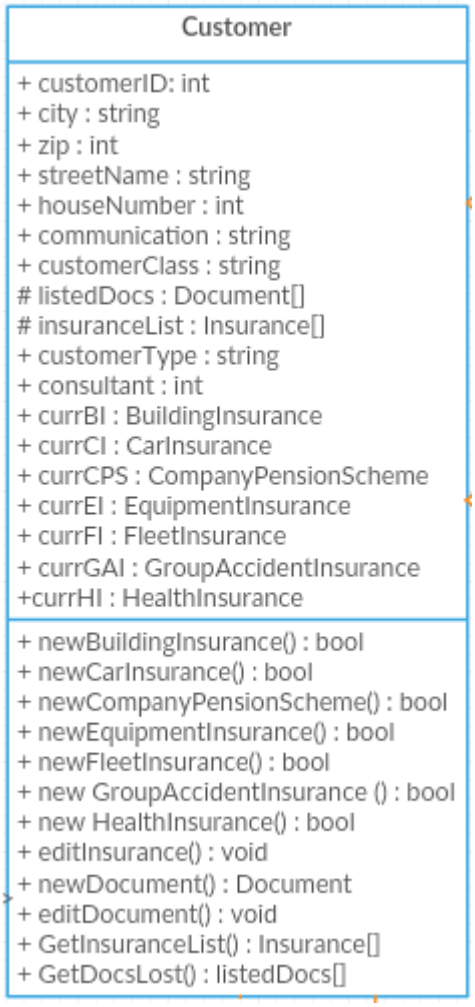


- Employee (Angestellter):

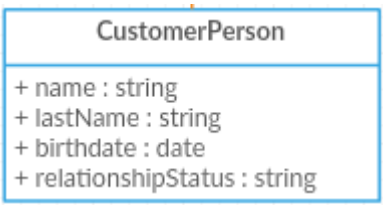
  - name, lastName, status(Innendienst, Chef, Außendienst), role, agency(in welcher Agentur), empID, currentCustomerPerson, currentCustomerEnterprise



- Customer:
    - customerID, City, Zip-Code, Streetname, Housenumber, communication(phone, Email,...), CustomerClass(vip, normal, potential customer), listed Document, List of Insurances, customer type(VN,VP,BZ), consultant (EmpID), current Insurances(listed)

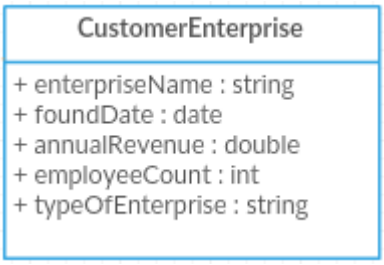


- Person:
    - name, lastname, birthdate, relationship status



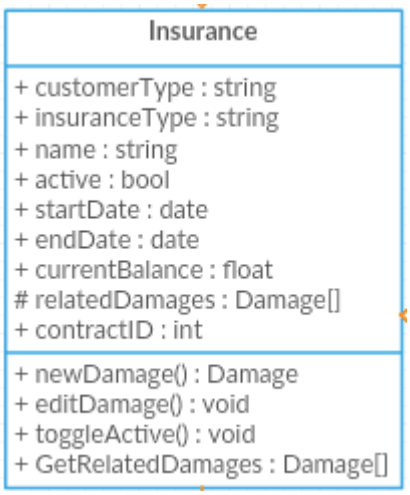
-Enterprise:

- Enterprisename, date of foundation, annual revenue, number of employees, type of the Enterprise



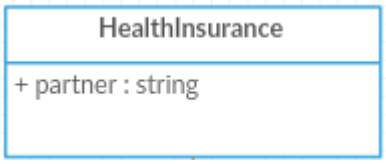
- Insurances:

- Customertype(Person/Enterprise), Insurancetype(personal related, object related), name, status(active/inactive), Startingdate, Enddate, payed Interrests, damages, contractID

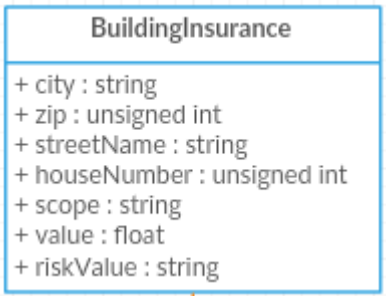


Subclasses:

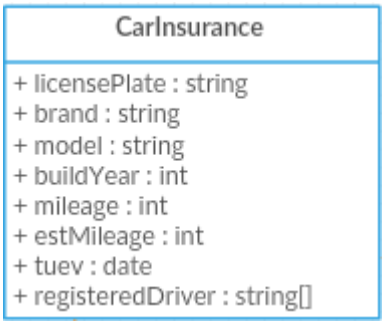
- Healthinsurance: cooperation partner



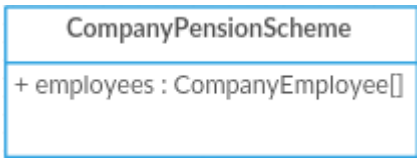
- Buildinginsurance(Person/Enterprise): laction, scope(what belongs to the building), value of the bulding, riskvalue



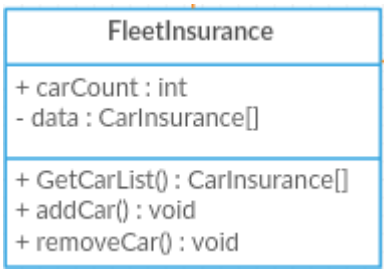
- Car Insurance: licensePlate, brand, model, buildyear, mileage, estimated yearly mileage, tuev, registered driver



- Company Pension Scheme: Employee of the Enterprise



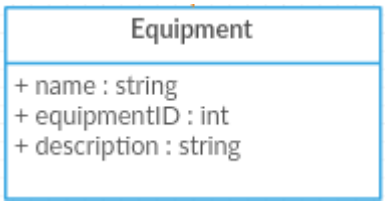
- Fleet Insurance: Count of all insured Cars, Data-Array of Car Insurance (see above)



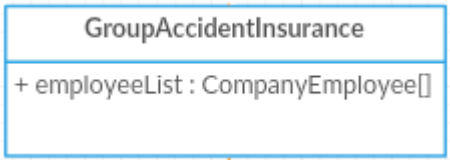
-

- Equipment Insurance (Machinery, Electronics):

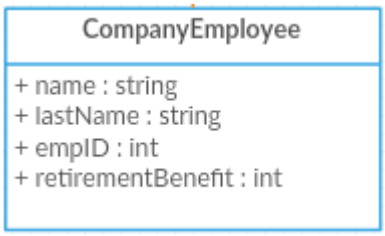
- Equipment-Part (Name, unique Equipment-ID, Description)



- Group Accident Insurance: List of all Employees

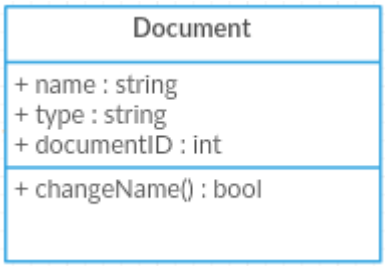


- Employee of foreign Company: Name, Last Name, unique Employee-ID, Retirement-Benefit



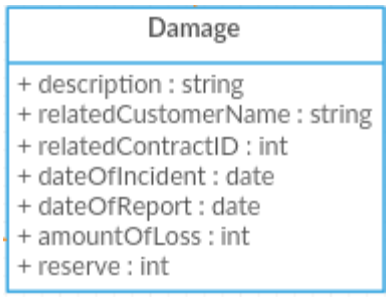
- Dokumente:

  - Name, Type (Contract, Consulting Protocol, Correspondence, Evidence), Dokument-ID



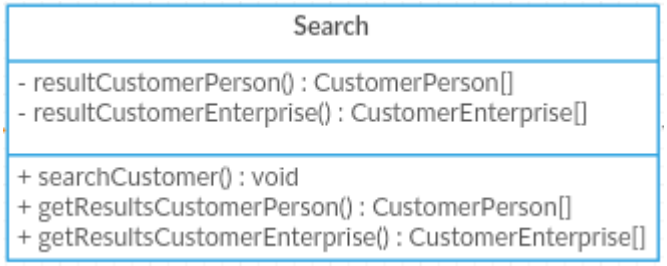
- Schäden:

  - Description, Related Customer, Related Contract-ID, Date the Incident happend, Date the Incident was reported, Amount of lost value, Rest of the Reserve



- Suche:

  - ResultCustomerPerson/Enterprise



Kalender und Projektplanung:

bis 08.11:

Ideenfindung und Vorbereitung der ersten Präsentation.

08.11 Vorstellungs-präsentation

21.11-28.11

Plan:

Festlegung Zeitplan, Überlegung welche Klassen werden gebraucht? wie viele? Organisatorisches

Erledigt:

23.11.: Projektbeschreibung ausarbeiten, Brainstorming und Klassen-überlegung, festlegung der Kommunikationswege und der Projektverwaltung

28.11-5.12

Plan:

Erweiterung der Klassenauswahl (Schäden und Versicherungen), Ausarbeiten der Klassen

Erledigt:

30.11.: Detaillierte ausarbeitung der Klassen

5.12-12.12

Plan:

Erstellen des UML-Klassendiagramms, Ausarbeiten der Zwischenpräsentation, Einrichten des Repository

Erledigt:

07.12.: Repository via GitHub eingerichtet, UML-Klassendiagramm geplant, Ausarbeiten der Klassen mit Funktionen

12.12.: Ausarbeiten der Zwischenpräsentation

13.12 Zwischenpräsentation

2.1-9.1

Plan:

Aufteilung der Arbeit, Vertrautmachen mit JUnit, Beginn Programmierarbeit

Erledigt:

04.01.: Festlegung des weiteren Zeitplans, Programmieren der ersten Klasse, Hinzufügen der restlichen Klassen als Dummies

16.1-23.1

Erledigt:

18.01.: Ausarbeiten der Projektstruktur mit Packages. Ersten Ausarbeiten einer Klasse. Genaues Vertrautmachen mit Unit-Tests. Erster Unit-Tests erstellt.

23.1-30.1

Erledigt:

25.01.: Arbeiten an der Customer Klasse, schreiben der SearchCustomer Klassen, Schreiben von Unit-tests für die Employee Klasse, Schreiben der Konstruktoren für die Insurances

26.01.: Kleine Änderung an SearchCustomer

27.01.: Alle Erstellungsmethoden für die Versicherungsklassen geschrieben

30.1-6.2

Erledigt:

30.01.: Änderung an SearchCustomer, Arbeiten an Damages

05.02.: Arbeiten an EditCustomer

06.02.: Hinzufügen von CustomerID, Überarbeiten der Code-Dokumentation, Aktualisieren aller UML-Diagramme, Erstellen der Abschlusspräsentation

7.02 Abschlusspräsentation