# MyDoctor

Источник <https://dev.to/alicefolvaiter/software-development-diagrams-base-exercise-mpk>

The **MyDoctor** application aims to be a management tool for the appointments of a doctor.

A hospital has multiple offices.

The users of the application can be doctors and patients.

The doctors can apply to practice in offices and create a schedule for an office. The schedules in different offices can’t overlay.

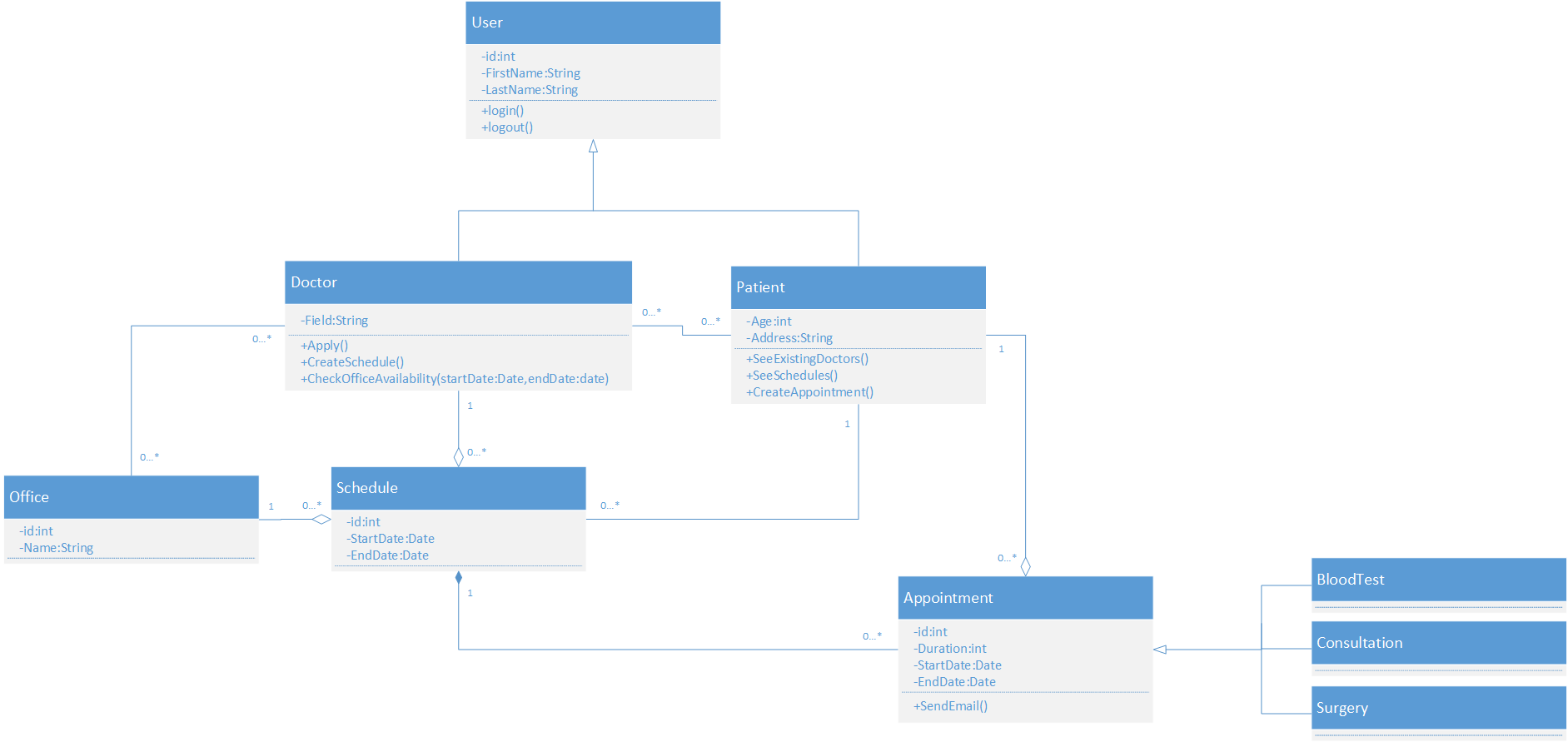
📝 *Example:  
Doctor Ana is available in Office 4 on the 4th of September during 1 PM - 5PM.  
Doctor Ana can’t practice in Office 5 on the 4th of September during 3PM - 8 PM, but she can practice in Office 5 on the 4th of September during 5:30PM - 8 PM.*

The patients can see the existing doctors in the system, the schedule of the offices and can book appointments for specific doctors and for specific schedules. The appointments can be of 3 types:

* Blood Test - 15 mins
* Consultation - 30 mins
* Surgery - 60 mins

The booking of an appointment will not be possible if another appointment is already booked at the same time frame. An email is sent to the patient with the confirmation of the appointment.

📝 *Example:****Action 1****: User Mike will create a blood test booking for Doctor Ana for the 4th of September starting with 15:30 PM → Possible****Action 2****: User Mike will create an intervention booking for Doctor Ana for the 4th of September starting with 15:00 PM → Not Possible****Action 3****: User Mike will create an intervention booking for Doctor Ana for the 4th of September starting with 16:00 PM → Possible*

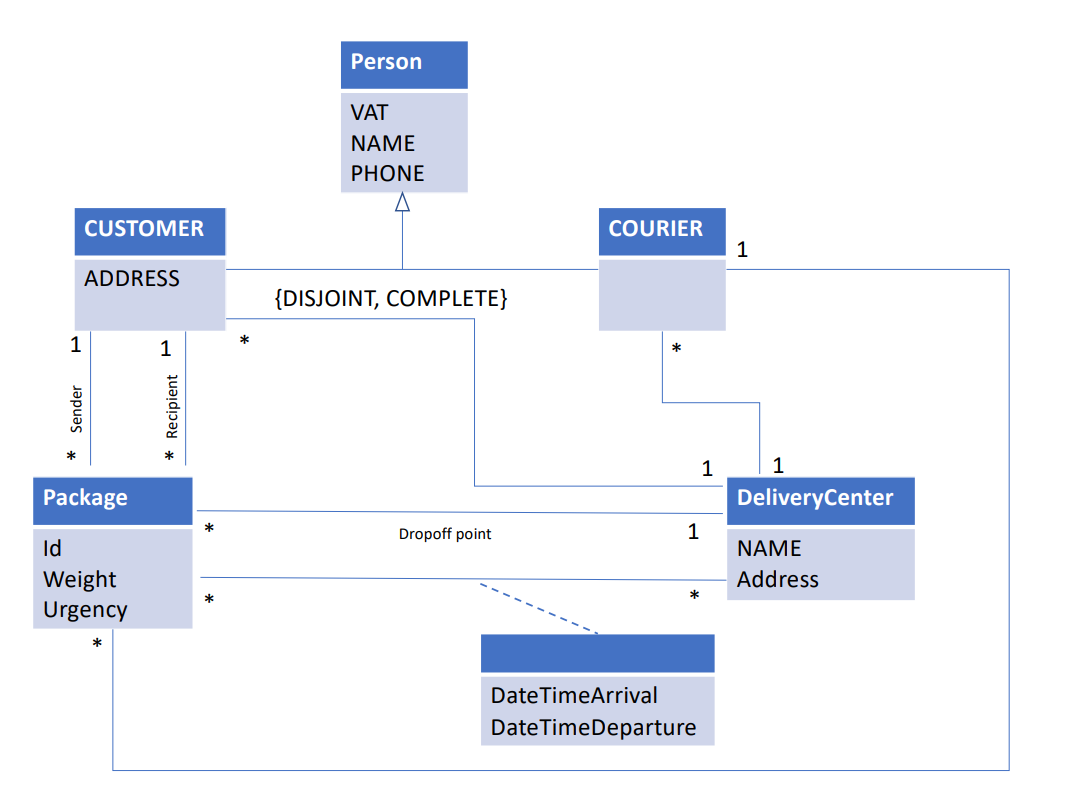


# Deliveries

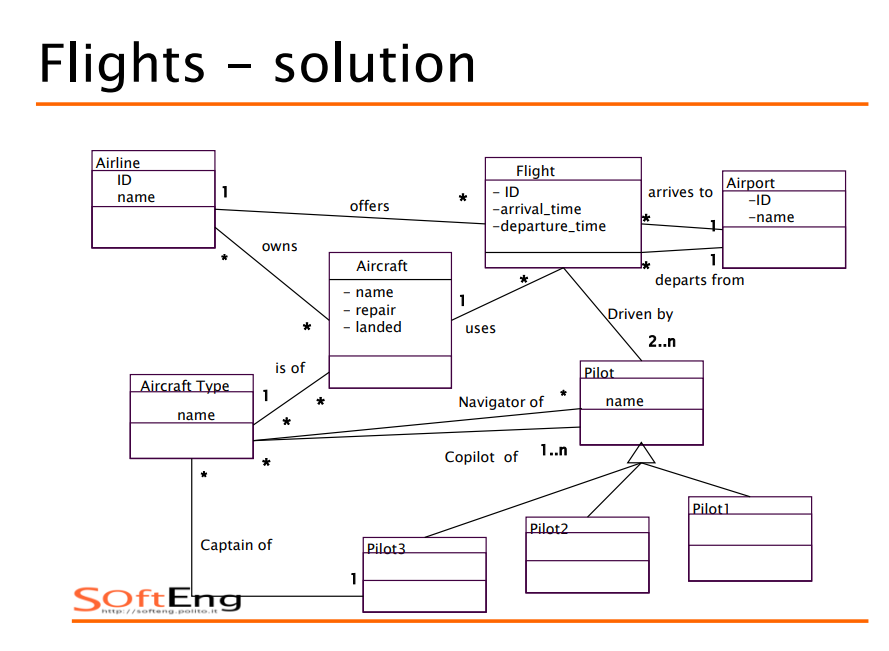
Источник [UML Class Diagrams Exercises | The caffeinated engineer (silvae86.github.io)](https://silvae86.github.io/teaching/exercises/uml/class/#:~:text=,%E3%80%9015%E2%80%A0Bicycles%E3%80%91)

The owner of a small delivery company plans to have an information system that allows him to save data about his customers and deliveries. After some time studying the problem, he reached the following requirements:

* Each customer has a VAT number, a name, a phone number and an address. There are no two clients with the same VAT number.
* When a customer wants to send a package to another customer, he just has to login to the company website, select the customer he wants to send the package to, enter the package’s weight and if the delivery is normal or urgent. He then receives an unique identifier code that he writes on the package.
* The package is then delivered by the customer at the delivery center of his choosing. A delivery center has a unique name and an address.
* Each client has an associated delivery center. This delivery center is chosen by the company and it is normally the one closest to the customer’s house.
* The package is them routed through an internal system until it reaches the delivery center of the recipient.
* The package is then delivered by hand from that delivery center to the recipient by a courier.
* Couriers have a single VAT number, a name and a phone number. Each courier works in a single delivery center.
* A courier is assigned to a packet as soon as the packet is introduced in the system.

Flights

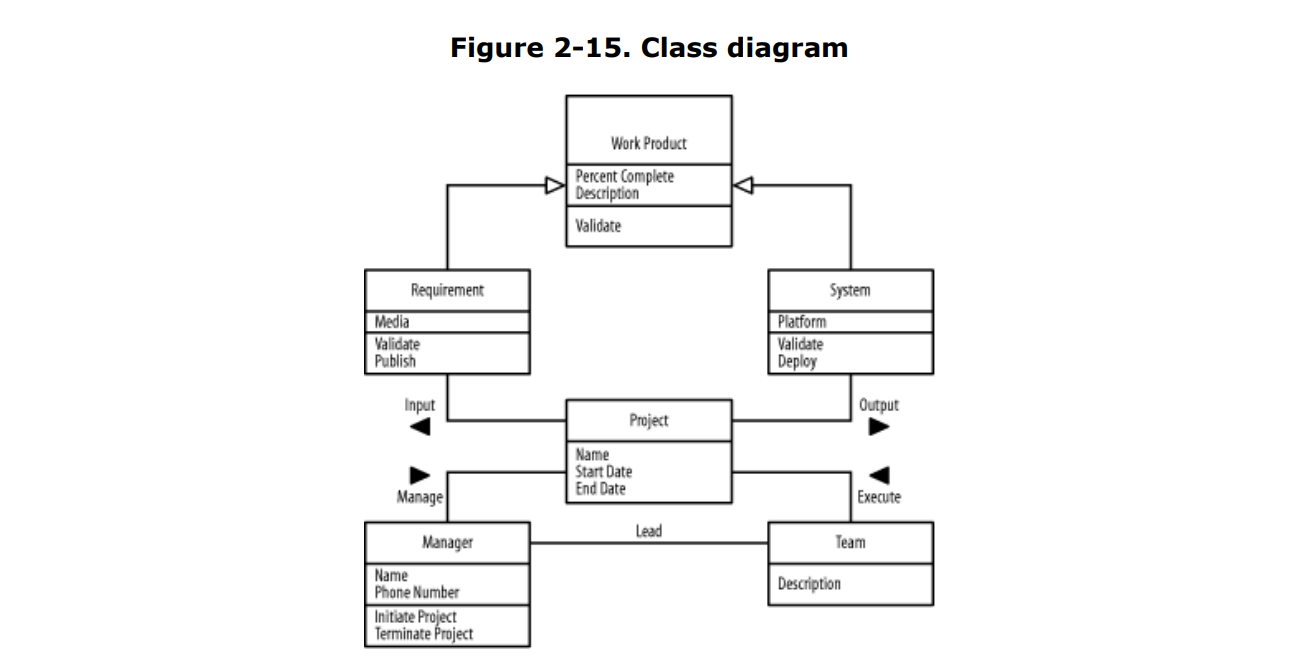
Источник Class Diagram Exercises - SoftEng Polito

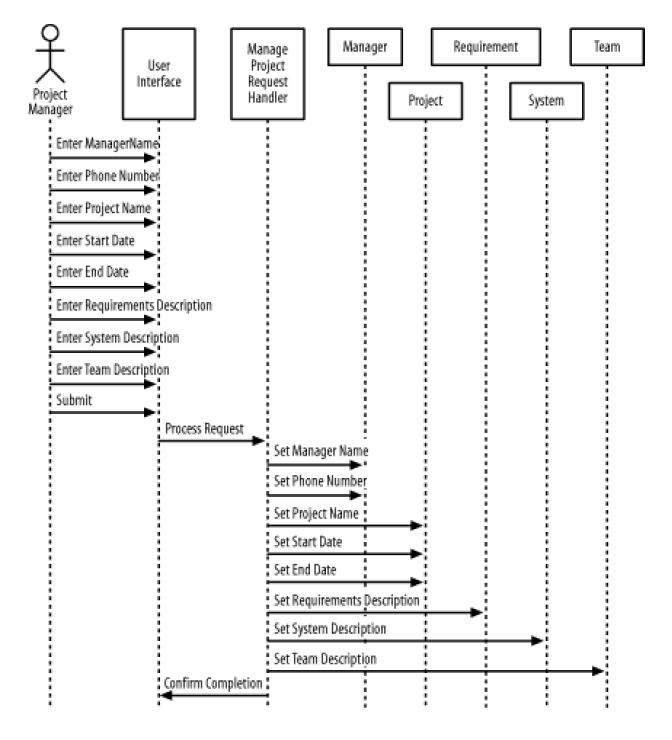
We want to model a system for management of flights and pilots. An airline operates flights. Each airline has an ID. Each flight has an ID a departure airport and an arrival airport: an airport as a unique identifier. Each flight has a pilot and a co-pilot, and it uses an aircraft of a certain type; a flight has also a departure time and an arrival time. An airline owns a set of aircrafts of different types. An aircraft can be in a working state or it can be under repair. In a particular moment an aircraft can be landed or airborne. A company has a set of pilots: each pilot has an experience level: 1 is minimum, 3 is maximum. A type of aeroplane may need a particular number of pilots, with a different role (e.g.: captain, co-pilot, navigator): there must be at least one captain and one co-pilot, and a captain must have a level 3.

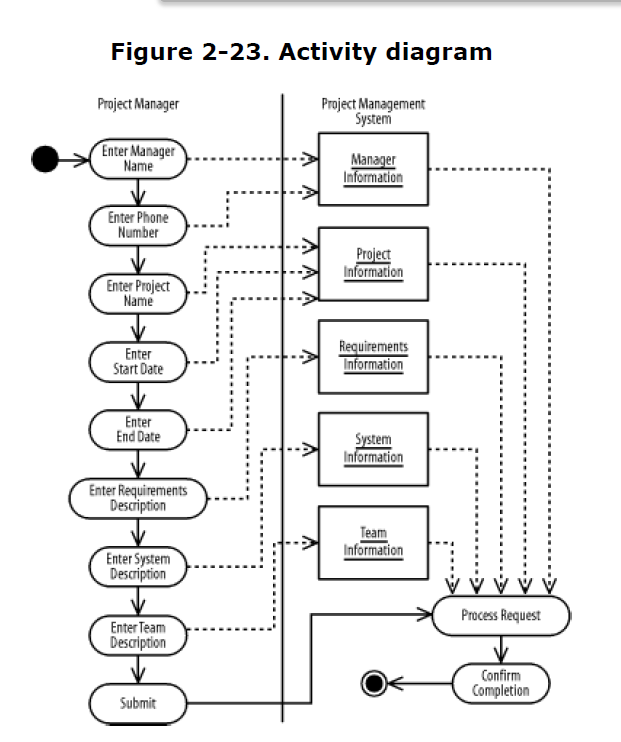
# Project Management System

**Источник**: Learning UML By Sinan Si Alhir

A project manager uses the project management system to manage a project. The project manager leads a team to execute the project within the project's start and end dates. Once a project is created in the project management system, a manager may initiate and later terminate the project due to its completion or for some other reason. As input, a project uses requirements. As output, a project produces a system (or part of a system). The requirements and system are work products: things that are created, used, updated, and elaborated on throughout a project. Every work product has a description, is of some percent complete throughout the effort, and may be validated. However, validation is dependent on the type of work product. For example, the requirements are validated with users in workshops, and the system is validated by being tested against the requirements. Furthermore, requirements may be published using various types of media, including on an intranet or in paper form; and systems may be deployed onto specific platforms.



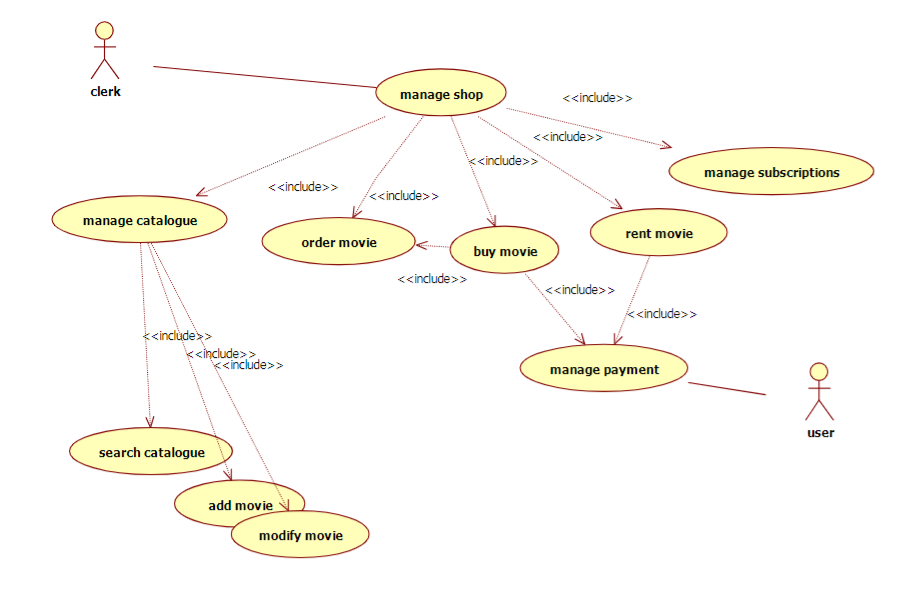


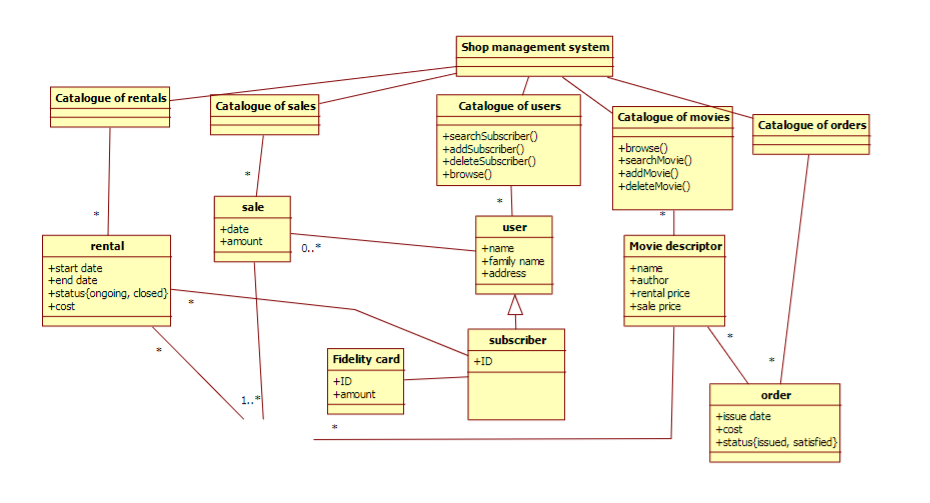


# Movie-Shop

Источник <https://softeng.polito.it/tongji/SE/ex/classDiagrams.pdf>

♣ Design a system for a movie-shop, in order to handle ordering of movies and browsing of the catalogue of the store, and user subscriptions with rechargeable cards. ♣ Only subscribers are allowed hiring movies with their own card. ♣ Credit is updated on the card during rent operations. ♣ Both users and subscribers can buy a movie and their data are saved in the related order. ♣ When a movie is not available it is ordered .





# Private parking lot

<https://www.visual-paradigm.com/tutorials/from-problem-description-to-models/>

Saturn Int. management wants to improve their security measures, both for their building and on site. They would like to prevent people who are not part of the company to use their car park.

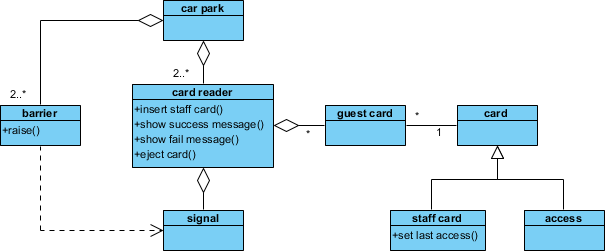
Saturn Int. has decided to issue identity cards to all employees. Each card records the name, department and number of a company staff, and give them access to the company car park. Employees are asked to wear the cards while on the site.

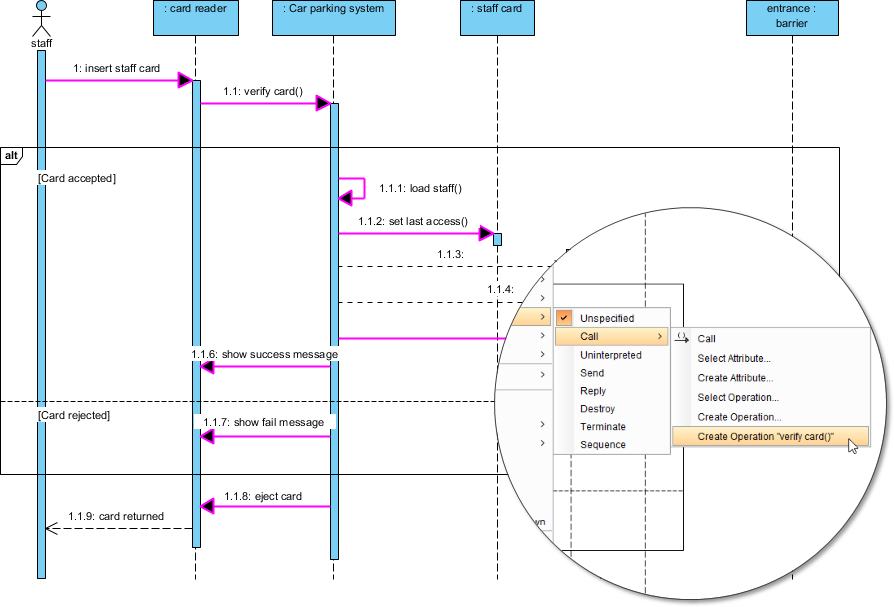
There is a barrier and a card reader placed at the entrance to the car park. When a driver drives his car into the car park, he/she inserts his or her identity card into the card reader. The card reader then verify the card number to see if it is known to the system. If the number is recognized, the reader sends a signal to trigger the barrier to rise. The driver can then drive his/her car into the car park.

There is another barrier at the exit of the car park, which is automatically raised when a car wishes to leave the car park.

A sign at the entrance display “Full” when there are no spaces in the car park. It is only switched off when a car leaves.

There is another type of card for guests, which also permits access to the car park. The card records a number and the current date. Such cards may be sent out in advance, or collected from reception. All guest cards must be returned to reception when the visitor leaves Saturn Int.

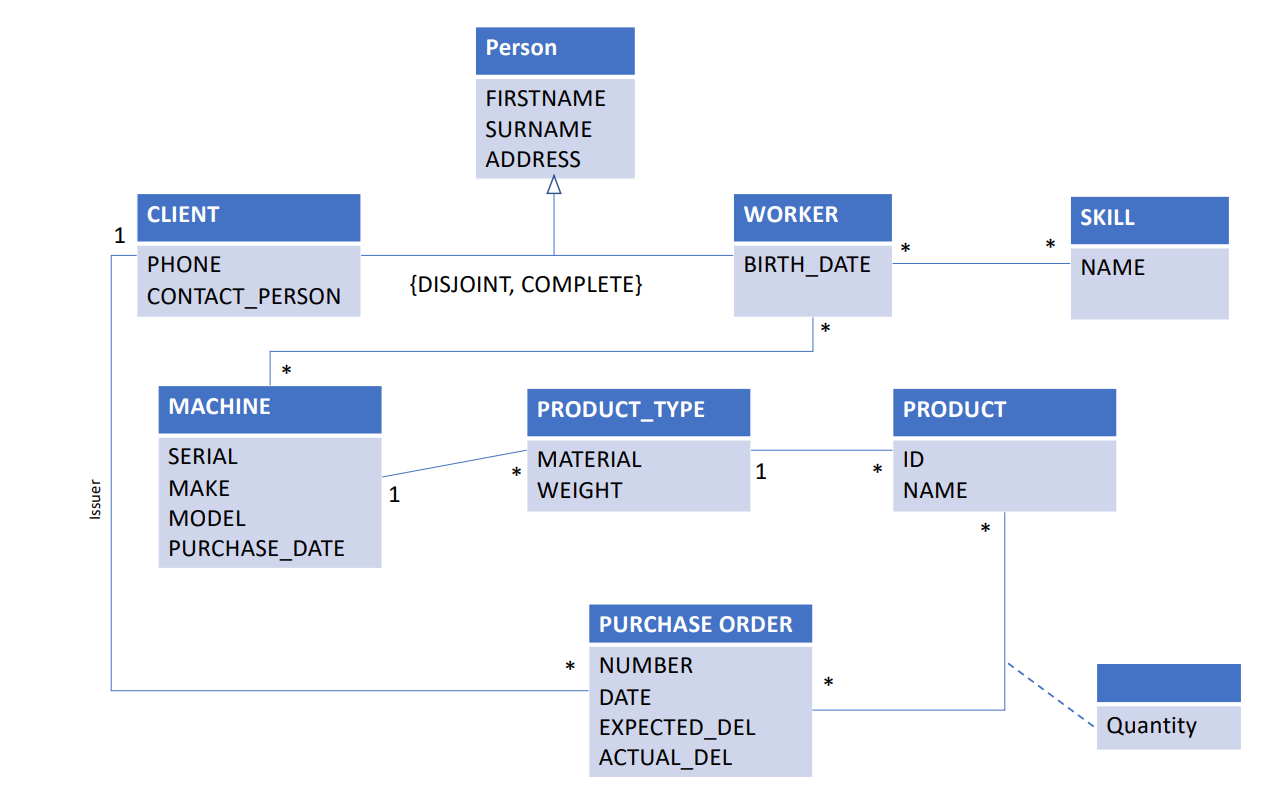




# Factory

Источник [UML Class Diagrams Exercises | The caffeinated engineer (silvae86.github.io)](https://silvae86.github.io/teaching/exercises/uml/class/#:~:text=,%E3%80%9015%E2%80%A0Bicycles%E3%80%91)

* A factory has several machines. Each one of them is operated by several workers.
* A worker might work in more than one machine.
* In this factory, several products of different types, are produced. Each different type of product is produced in a single machine. But, the same machine can produce more than one type of product.
* Products from the same type are all produced from the same single material and have the same weigth.
* Clients can issue purchase orders. Each order has a list of the desired products and their quantity.
* For each worker, the following data should be stored in the database: name (first and last), birth date, address and a list of his skills.
* For each machine, the following data should be stored: serial number, make, model and purchase date.
* For each client, the followig data should be stored: name, address, phone number and name of the contact person (if any).
* For each purchase order, the following date should be stored: order number, date it has been made, expected and actual delivery date.



# Restaurant Management System

Источник: <https://ieee-dataport.org/documents/dataset-text-requirements-models#>

Purpose

Restaurant management systems are a type of restaurant management software intended to assist with the tasks associated with the day-to-day management of a restaurant.

Scope

A restaurant management system may also run on your tablet, so waiting staff can take orders and alert the kitchen and counter even before leaving the table.

Requirements

Restaurant management systems can include the following:

Point-of-sale System: Used to input orders and record payments. The system captures all transactions, including charges, payments, voids, promo deals, and expenses.

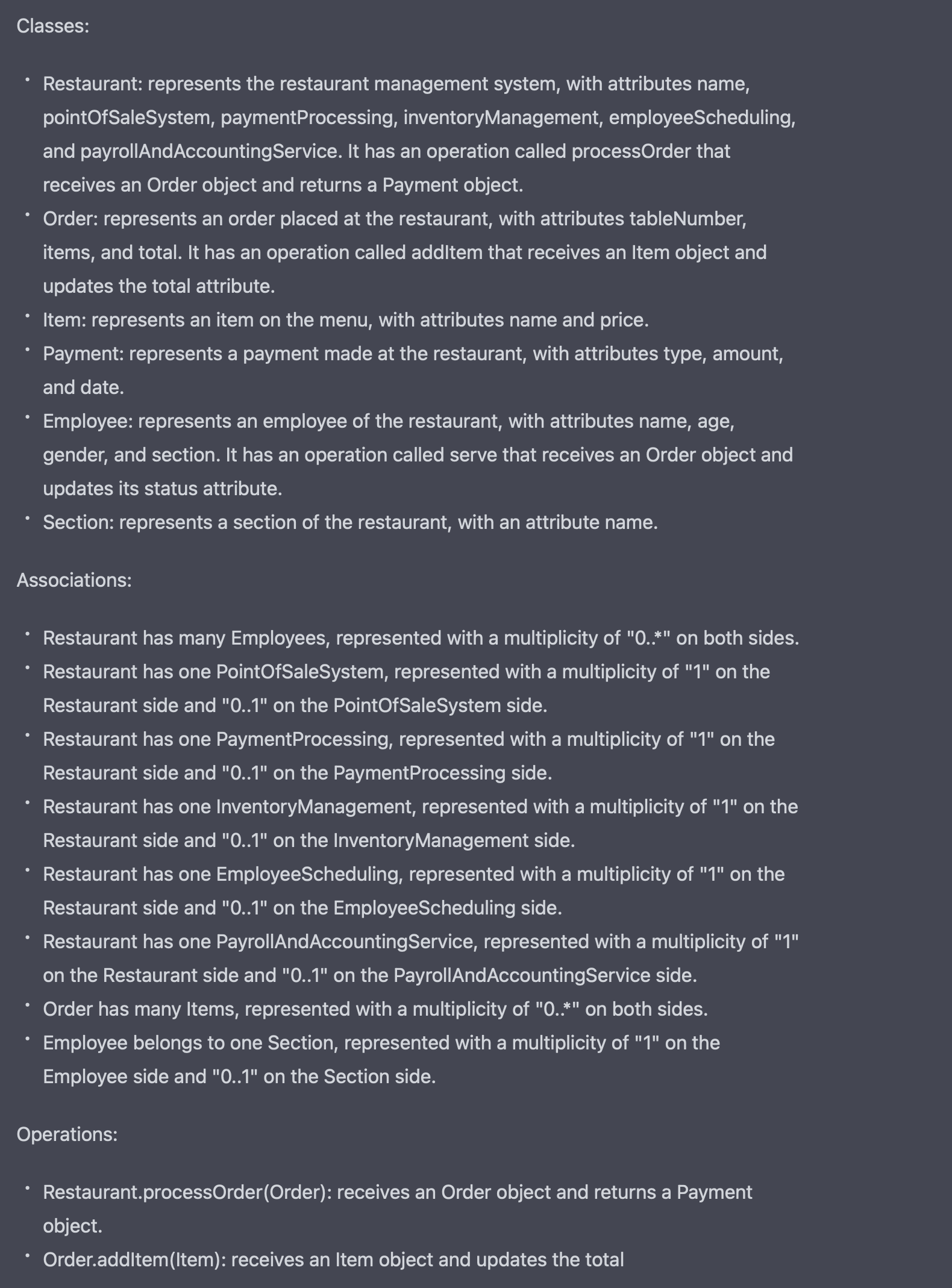
Payment Processing: This allows you to accept various payment options like credit or debit cards and mobile payments.

Restaurant Inventory Management: Used to monitor current stock, the cost of the items, and more. The system shows the order details and table number when the cook prepares the order. The waiter in the service section serves the customer. When the waiter serves, the system shows the waiter's name and table number.

Employee Scheduling: Check employee attendance, shifts, and time off.

The restaurant has more than one employee. Each employee has a name, age, and gender information. The restaurant has more than one section. These sections are kitchen, service, and cash. Personnel works according to their sections. Personnel can be cooks, waiters, and cashiers. The cook stays in the kitchen and prepares the orders. The waiter in the service section serves the customer. The cashier prepares the check according to the order details in the system.

Payroll and Accounting Service: Used to manage employee payroll and monitor the restaurant's cash flow.



# Hotel reservation

The total price of the reservation depends on the time of year and the type of room. There are three types of rooms in the hotel. There are 100 odd numbered rooms which types are standard, and six even numbered rooms which are large-deluxe. The types of remaining rooms are Deluxe.

The price of standard rooms:

$60 for the summer season,

$40 for the spring season,

And $25 for the autumn season per night per person. The room takes up to 4 people. The price is calculated according to the number of people entering the room.

Rates for the deluxe room:

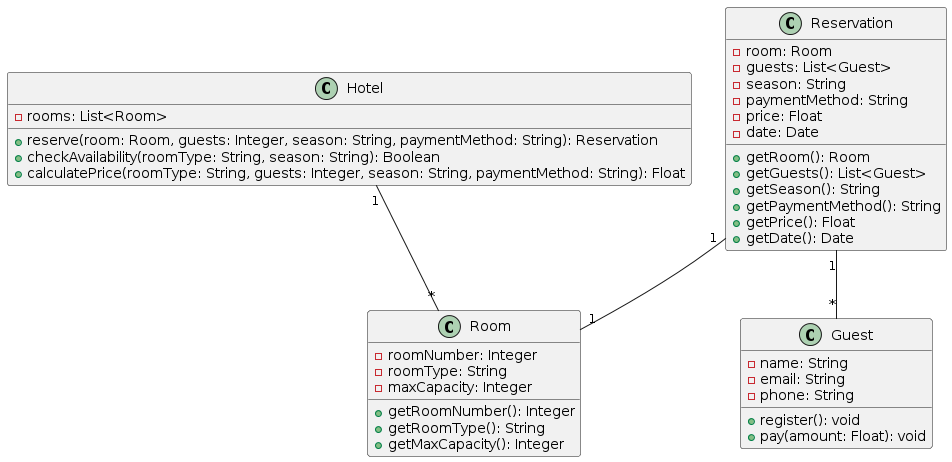
$90 per person per night in the summer,

$75 in the spring,

And $ 50 in the autumn. The room takes up to 4 people. The price is calculated according to the number of people entering the room.

While calculating the price for Large Deluxe rooms, the number of people is not considered. It is $ 400 daily in the summer, $ 300 in the spring, and $ 250 in the autumn. There can be a maximum of 7 people in this room.

The user must be registered for the first time to the system. She must enter for pre-payment, 'H' for payment with a wire transfer, and 'S' for a standard reservation. A 15% discount will be applied on the price calculated when pre-paid reservation is selected. When payment using a wire transfer is selected, if the number of guests exceeds three persons, 50% of 1 guest's fee and 75% of 1 guest's fee will be deducted from the calculated amount. In standard reservations, customers make payments at the end of the accommodation. If all the rooms are reserved before, the user will be asked to make a choice again. If the user wishes to make another reservation transaction, the loop must start, and all transactions must be repeated.



# Donation

Purpose

This study aims to design a new software that allows bank customers to make donations to the various organizations such as orphanages etc. Finishing this project, customers will be able to make their good-willing activities easily with their banks.

Scope

Study will include “Donation” process on both web and mobile systems. There will be two separate applications for the mobile platforms which are Android and iOS. They should have same function and features for the account creation process. Project will also contain the back-end communication system for 3rd party charity organizations.

Requirements

Menu Placement

“Donation” process should be under the “Accounts” menu.

It should be added to the bottom of the page.

Customer

Users can transfer their money to the charity organizations via web or mobile banking system.

System should check if user is eligible for donation or not.

For being eligible, user should have at least one available, active, account and the selected account balance should be enough for the donation.

The organization list should be retrieved from the internal banking systems. Bank staff can add or remove the organization from the donation list. When banking staff create the list and publish it into the online banking, users can see the list from the mobile app. Users should see a dropdown menu which has related charity organizations.

Software System

On the donation process, user should select an account and organization information.

If selected account type is a savings account, system should display a pop-up stating that user is continuing with a savings account.

After user selects organization and account information, system should ask for amount of money to be transferred.

After these 3 inputs, system should display a confirmation screen.

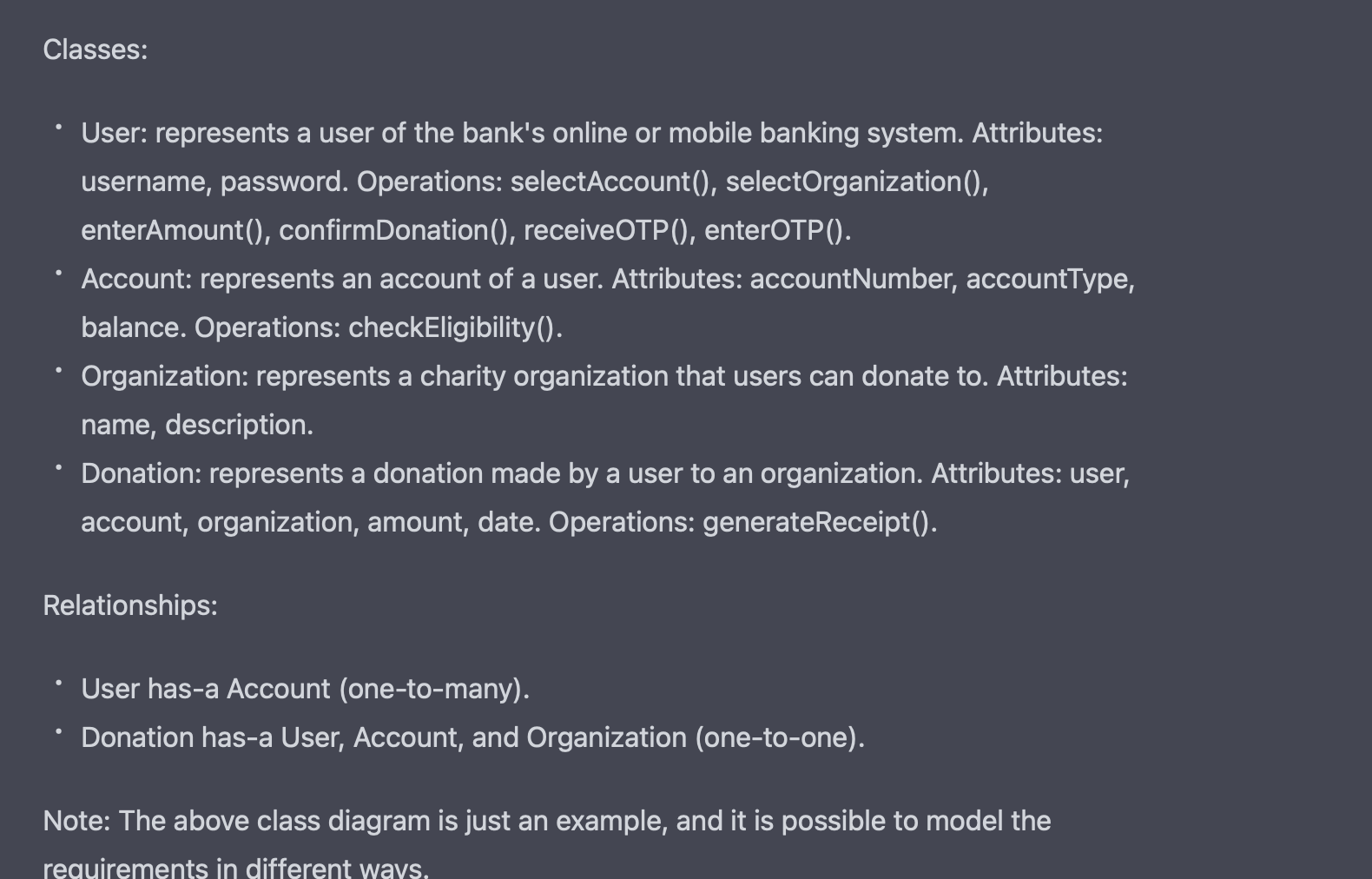
If user confirms the information, system should send an SMS containing a random OTP. When user enters that OTP and continues, result page should be displayed.

System should create an e-receipt and shows to the customer. E-receipt should contain donation information and account details.

Rules

Donation organization and amount of the money should be directly transferred to the banking systems via encrypted tunnel.

Information about the donations should be kept hidden from the mobile or web application storages.



# Employee Management System

Purpose

Complete the design of the Employee class and Manager class, and write a test class to initialize some employee and manager information and output the information of employees and managers who have reached retirement age.

Scope

The system is designed to show a company’s employee hierarchy. The employees of a company are in various positions. In a hierarchy, the manager has subordinates; employees report to them. The employee has no subordinates.

Requirement

A company has two roles: Employee and Manager. The manager is a special employee. Each employee object contains information about the department, name, date of birth, and social insurance number. A company has more than one employee. These employees are included in the categories of workers, salespeople, and managers. In addition, while workers have information such as weekly working hours and hourly rates, the salespersons have information such as fixed salary, amount of sales made, and commission percentage. The managers only get a fixed salary. Workers are separated as shift workers and non-shift workers. Off-shift workers have weekend and official holiday permits. Shift workers receive premiums for working on holidays. The company has production, control, and delivery departments. Each department has to be controlled by a manager. Shift workers can only work in the delivery department.

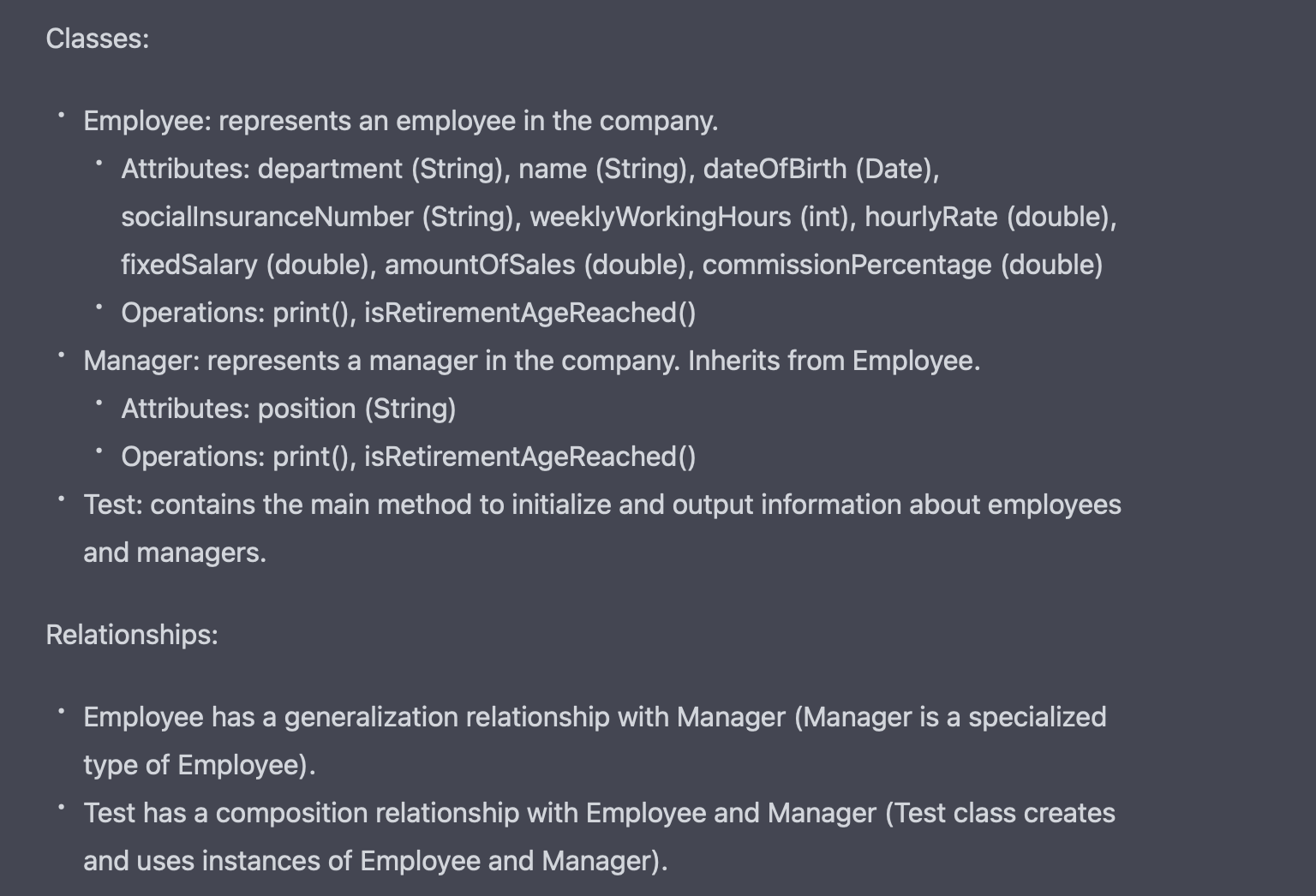
In addition to the above basic information, the manager object also has position information. Employees and managers in the company have the following two basic operations:

Printing

It outputs the basic information of the object.

RetirementControl

It checks whether the retirement age is reached. The company stipulates that the retirement age of employees is 60 years old, and the retirement age of managers is 65 years old.



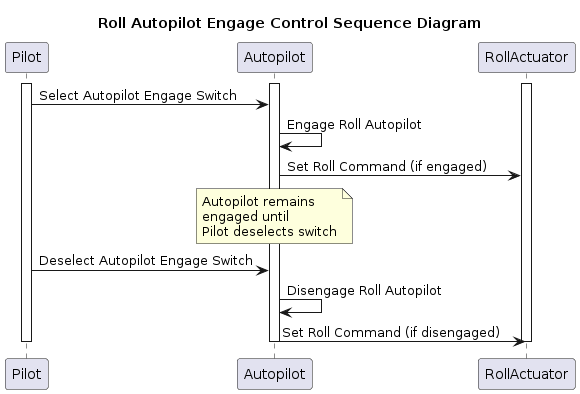
# Autopilot

Источник: PURE: a Dataset of Public Requirements Documents

Roll Autopilot Engage Control

Requirement Roll Autopilot shall engage when the pilot selects the autopilot engage switch in the cockpit and disengage when the switch is deselected. When not engaged, the command to the roll actuator shall be zero.

Rationale The autopilot should only be engaged when the pilot selects it. "



# URBAN MOBILITY SIMULATOR

Источник: <https://zenodo.org/records/4121935#.X53KXRvPxPY>

Requirements:

1.The user authenticates and is logged into the system.

2.The system shows the traffic flows of the city, based on historic data.

2.1 Congested traffic flows are shown in red.

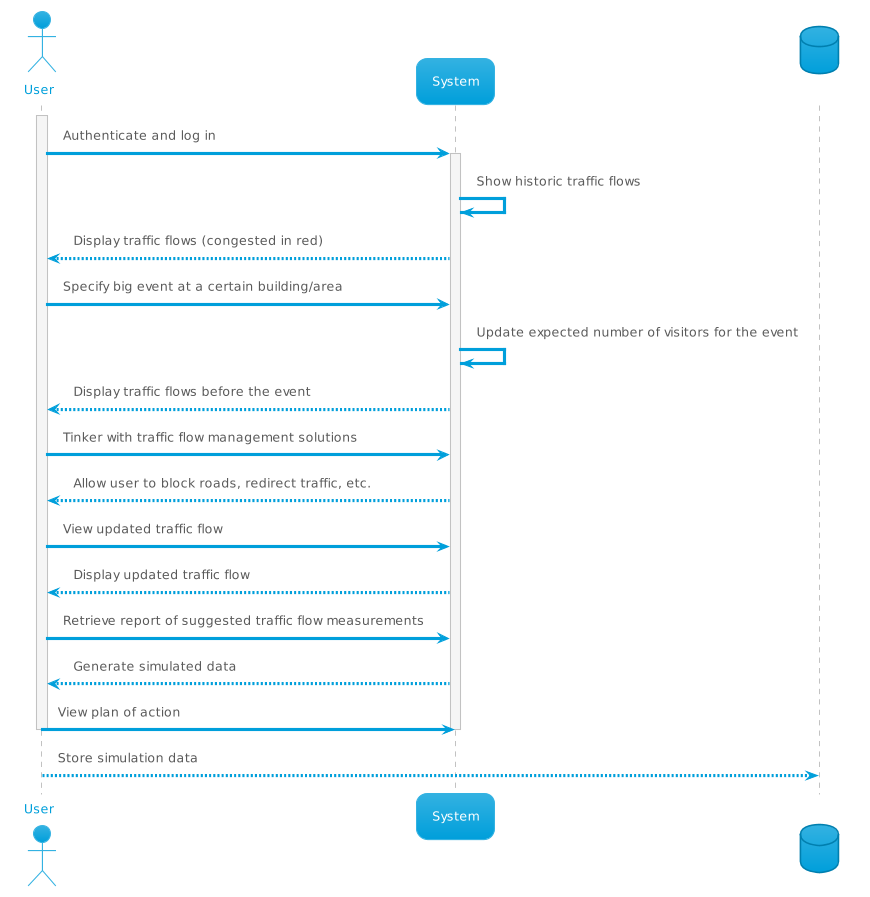
3.The user specifies a big event at a certain building/area with an expected number of visitors.

4.The system shows the traffic flows from before and after the event.

5.The user selects can tinker around with traffic flow management solutions (blocking certain roads off, redirecting traffic, etc.)

6.The system shows the updated traffic flow.

7.The user retrieves a report of suggested traffic flow measurements and plan of action generated by the system (simulated data).



# Redundancy management system

Источник: PURE: a Dataset of Public Requirements Documents

The purpose of this redundancy management system is to prevent errors from propagating past the input portion of an airborne application. The assumed platform configuration is a set of three computers which execute identical software applications, and which each take the same set of input values from sensors on the aircraft. For each set of triplex inputs, the redundancy management system must monitor for failures (or imminent failures), report the failure status of the set to the necessary functions, and choose an acceptable signal for computational use. All of the inputs used in redundancy management are triply redundant and have some type of tie-breaking capability.

A.1.1 Requirements

1. In the no-fail state, a miscompare, which shall be characterized by one branch differing with the other two branches by a unique trip level that lasts for more than the persistence limit, shall be reported to failure management as a failure.

2. In the no-fail state, the mid-value shall be the selected value. Note: a first failure in progress will not affect the method for determining the selected value.

# INTERNATIONAL FOOTBALL ASSOCIATION (IFA) PORTAL

Источник: <https://zenodo.org/records/4121935#.X53KXRvPxPY>

Файл g05-uc-req

