CS132: Software Engineering

HW1: UML Practice

In this homework we will use UML to describe the functionality of a system and specify its interactions with users. When drawing UML diagrams, please use software recommended in class(Visio /Visual Paradigm/ draw.io). Please try to answer all the questions in English and submit a .pdf report on Blackboard. The deadline for submission is Mar 6th at 23:59.

1 Class diagram (25pt)

A company called Warm-Up has several departments. Each department is managed by a head, who is also a staff. Each department carries out at least one project and has at least one staff. The department head is responsible for issuing policies and evaluating other staffs. Staffs must be assigned to one, but possibly more departments. Each staff takes part in at least one project but each project may involve many different employees. The staffs have the right to get the salary and requesting for leave.
In addition to the above description, we know the department names, the staff names, the staff ids, the project names, and the department head ids.
Please draw a **Class diagram** using the information given above.

2 Use Case diagram (25pt)

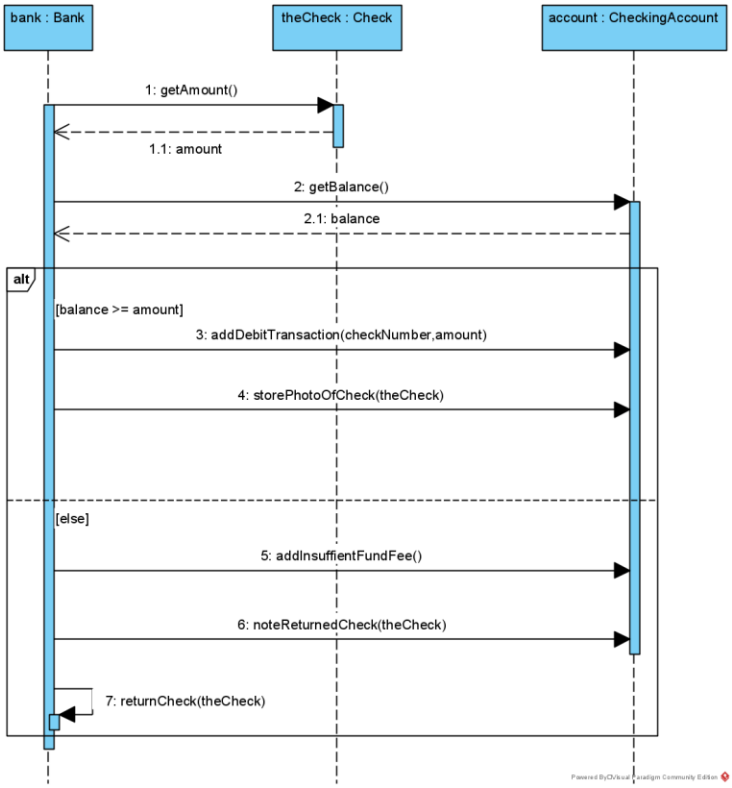
Gezi Wang is a staff member of this company. He is in charge of the company's cashier and accounting operations, and often needs to operate ATM machines. Draw an **Use Case diagram** to illustrate what he can do with the ATM system.
(Hint: The ATM system should at least include functions for logging in, ejecting the card, transferring money, withdrawing money, making deposits, checking balances, and printing receipt)

3 Activity diagram (25pt)

Draw an **activity diagram** to illustrate the process for Gezi Wang using the ATM to transfer money. Make sure to use **Swim Lane** to group activities for different roles.
(Hint: You should at least consider the case of incorrect PIN and insufficient balance/invalid transactions)

4 Sequence diagram

Observe the sequence diagram and answer the corresponding questions



4.1 Determine whether the following statements are correct (12pt)

getAmount() is an asynchronous message.

True **False**

The execution order of getAmount() and getBalance() is not guaranteed unless “strict” operator is added.

True **False**

An opt fragment cannot be enclosed in other combined fragments such as strict or par, while alt can be enclosed in such fragments.

True **False**

4.2 Drawing Practice (13pt)

Now, according to the known sequence diagram and the template given below, let us change the **alt** operator to the **opt** operator, so that the two diagrams are equivalent.