

# Prutt 06 Hemtenta

## Model Answers

## **Question 1: Data Dictionary**

### **Noun List:**

Amount: amount of sale for a commission

Bank account: destination for direct deposit payment

Commissioned Record: records commission bonuses

Commission rate: a percentage of sale on commission

Company cashier: An actor who initiates payments

Date: Workday date or date of sale

Eight hours per day: standard working day for overtime definition

Employee: can be hourly rate or salaried

Employment record: worker details including name

Friday: payment day for hourly rate employees

Hourly rate: rate of pay per hour

Hourly Record: records an hourly rate of pay

Hours worked: hours worked

Method of payment: can be mailed paycheck, held for pickup or direct deposit

Name: of an employee

Number of hours worked: work effort

One point five times normal rate: Standard overtime rate

Paycheck: a paper form of payment

Payment: a specific payment to an employee on a particular working day

Payroll application: the main application to be defined

Postal address: for a mailed paycheck we need a postal address

Salaried Record: records monthly salary

Sales Receipt: receipt of amount obtained by a commissioned employee for closing a sale

Second Friday: payment day for salaried employees

Service charge: additional union charge for work performed

Specified date: the date of termination of employment of an employee

Time card: a record of date and number of hours worked, one for each day

Union: trade union for workers

Weekly dues: weekly fees to union

## **Verb List**

Deduct: deducting money from payments for weekly union dues and union service charges

Direct deposit: a computer transaction to transfer payments to an employee bank account?

Generate payments: the action of making specific payments each working day.

Pay: produce payments using the payroll program for employees

Submit: a union may submit a weekly service charge

*//Note below I have chosen to subclass employment records rather than employees*

## **Relational Phrase List**

Commissioned Record has a commission rate: one record has one commission rate for a commissioned employee

Direct deposit is a method of payment

Direct deposit has a bank account

Employee is a hourly rate employee

Employee is a salaried employee

Employee has a employment record: one employee has one unique employment record

Employee has a method of payment: one employee has one method of payment

Employee has a payment: one employee has many payments, one on each payday

Employee belongs to Union: an employee may or may not be a member of (one?) union.

Employee has a service charge: an employee who is a member of a union may have many service charges (for different services?)

Employment record has a name: one record has one name

Hourly Record is a Employment record

Hourly Record has a hourly rate: one record has one hourly rate for an hourly rate employee

Held for pickup is a method of payment

Mailed paycheck is a method of payment

Mailed paycheck has a postal address

Salaried Record has a monthly salary: one record has one monthly salary for a salaried employee

Salaried record is a employment record

Sales receipt has a amount

Sales receipt has a date

Time card has a date: one time card has one date (today's?)

Time card has a hours worked: one time card has one hours worked field

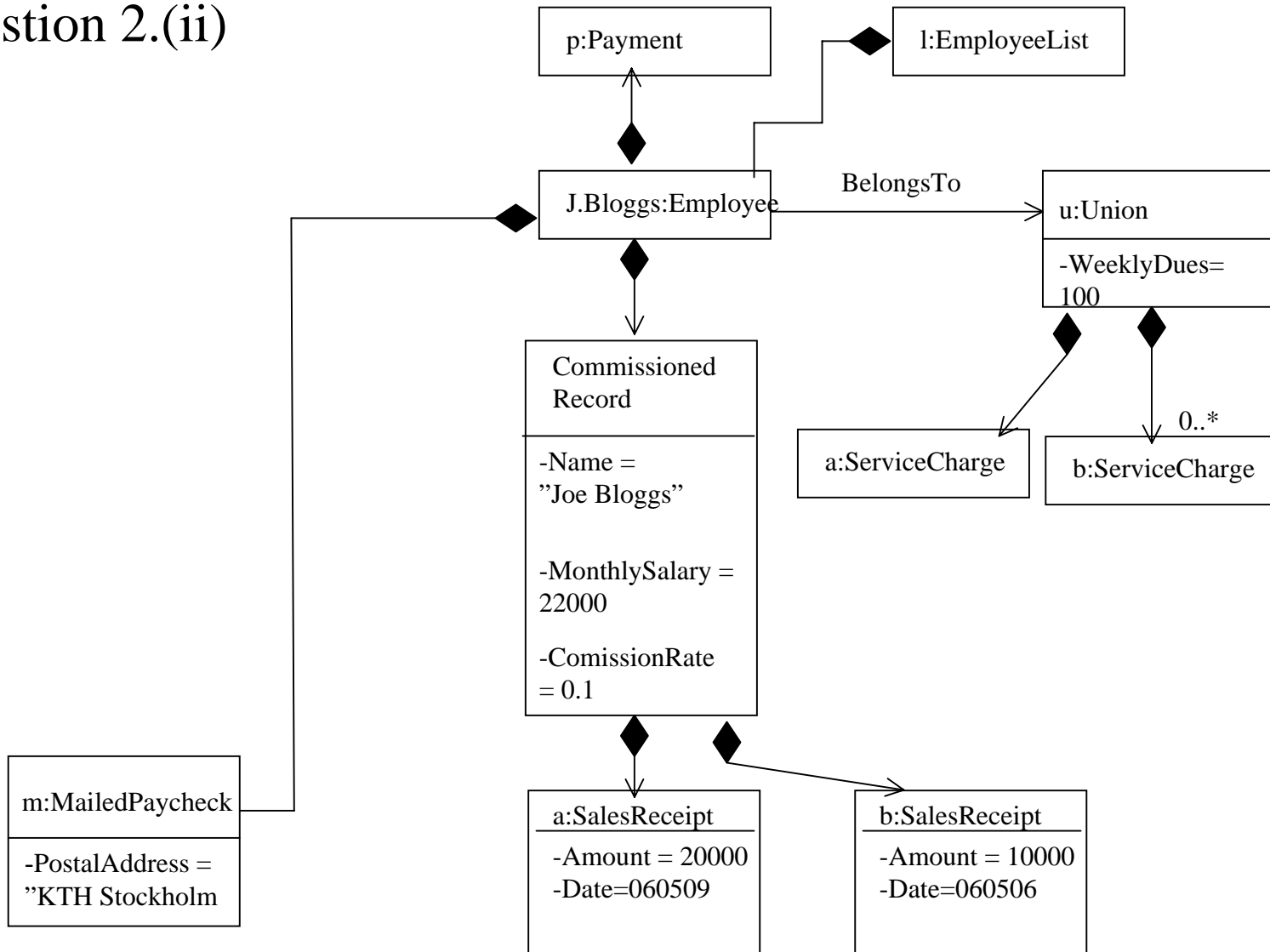
Union has a weekly dues: One union has one weekly dues rate for all members

Union has a service charge: One union has a service charge for a member

Payment has a specified date: a payment has a specified date when an employee ceases work



# Question 2.(ii)



### **Question 3.**

**3.(i)** The waterfall model seems an ideal choice here. No other model seems a reasonable alternative to accept as a different answer.

#### **Strengths**

1. The company is large and bureaucratic, requiring documentation and good project visibility. Waterfall documentation satisfies this.
2. The company has a high staff turnover. So probably many employees are new and inexperienced. Waterfall works well with inexperienced staff.
3. The application type is well known to the company, so a thorough user requirements analysis should be possible early on.
4. An existing application is to be re-engineered. This also confirms that the user requirements are known early. They could be extracted from the existing application.
5. The company will maintain the software, so good documentation will be essential.
6. The company has a good reputation to maintain in this area, so quality control will be essential. Waterfall emphasis of requirements-before-design and design-before-coding will emphasise quality.

7. The Cobol implementation has very little documentation. This seems to be necessary for future maintainance by our company (we may save money in the long term if we produce some). Waterfall will naturally produce such documentation

## **Weaknesses**

Waterfall has no obvious weaknesses on this project.

1. Late arrival of code may be one, but on the other hand our client may be able to continue using the existing application until the new system arrives.
2. We may end up swimming upstream, but the chances seem minimal since the requirements should be clear from the start.
3. The administrative overhead is high, but our large IT company is inherently bureaucratic so presumably has admin staff to handle this.

**(0.5 points per argued case, roughly!)**

**3.(ii)** A 3 or 4 tier client server architecture seems appropriate here since the environment target is a distributed architecture. **(1 point)**

In a 3-tier system, the basic employee information could be stored in a relational or better still an object-oriented database. The application logic would contain the code for payroll calculation. The client side code would simply contain GUIs for data display and entry. **(3 points, 1 per tier)**

We could apply the Model-View-Control (MVC) pattern to this application, by using the client side processing to implement M and V, and the server side processing to implement the C. **(0.5 point client side, 0.5 point server side)**

## Question 4.(i)

```
<!ELEMENT EmployeeList (Employee*) >
<!ELEMENT Employee (EmploymentRecord, MethodofPayment, Payment*,
    Union?) >
<!ELEMENT EmploymentRecord (HourlyRecord | SalariedRecord) >
<!ATTLIST EmploymentRecord Name CDATA #REQUIRED>
<!ELEMENT HourlyRecord (TimeCard*) >
<!ATTLIST HourlyRecord HourlyRate CDATA #REQUIRED>
<!ELEMENT TimeCard (EMPTY) >
<!ATTLIST TimeCard
    Date CDATA #REQUIRED
    HoursWorked CDATA #REQUIRED
>
<!ELEMENT SalariedRecord (CommissionedRecord?) >
<!ATTLIST SalariedRecord MonthlySalary CDATA #REQUIRED>
<!ELEMENT CommissionedRecord (SalesReceipt*) >
<!ATTLIST CommissionedRecord CommissionRate CDATA #REQUIRED>
<!ELEMENT SalesReceipt (EMPTY) >
```

<!ATTLIST SalesReceipt Amount CDATA #REQUIRED Date CDATA #REQUIRED>

<!ELEMENT MethodofPayment (HeldforPickup | DirectDeposit | MailedPaycheck )>

<!ELEMENT HeldforPickup (EMPTY)>

<!ELEMENT DirectDeposit (EMPTY)>

<!ATTLIST DirectDeposit BankAccount CDATA #REQUIRED>

<!ELEMENT MailedPaycheck (EMPTY)>

<!ATTLIST MailedPaycheck PostalAddress CDATA #REQUIRED>

<!ELEMENT Payment (EMPTY)>

<!ELEMENT Union (ServiceCharge\*)>

<!ATTLIST Union WeeklyDues CDATA #REQUIRED>

<!ELEMENT ServiceCharge (EMPTY)>

## Question 4.(ii)

< EmployeeList>

<Employee>

<EmploymentRecord Name = 'J.Bloggs'>

<SalariedRecord MonthlySalary = '22000'>

<CommissionedRecord CommissionRate = '0.1'>

<SalesReceipt Amount = '20000' Date = '060509'>

</SalesReceipt>

<SalesReceipt Amount = '10000' Date = '060506' >

</SalesReceipt>

</CommissionedRecord>

</SalariedRecord>

</EmploymentRecord>

<MethodofPayment>

<MailedPaycheck PostalAddress = 'KTH Stockholm'>

</MailedPaycheck>

</MethodofPayment>

```
<Payment>
</Payment>
<Union WeeklyDues = '100'>
    <ServiceCharge>
    </ServiceCharge>
    <ServiceCharge>
    </ServiceCharge>
</Union >
</Employee>
< /EmployeeList>
```

## Question 4.(iii)

Looking through the class diagram, we see that every instantiated class contains objects with attributes that can appear as valid elements or attributes in a corresponding XML file that will be valid for our DTD. Looking at all the constraints of the class diagram we see that an object which does not violate these constraints will lead to an XML file that does not invalidate the DTD. These facts should be checked by the student individually for every class, attribute and constraint, in order to achieve full marks.