

**NAME** TingTing Shao

**STUDENT NUMBER** R0925882

**COURSE NAME** Principles of Database Management

**ACADEMIC YEAR** 2022-2023

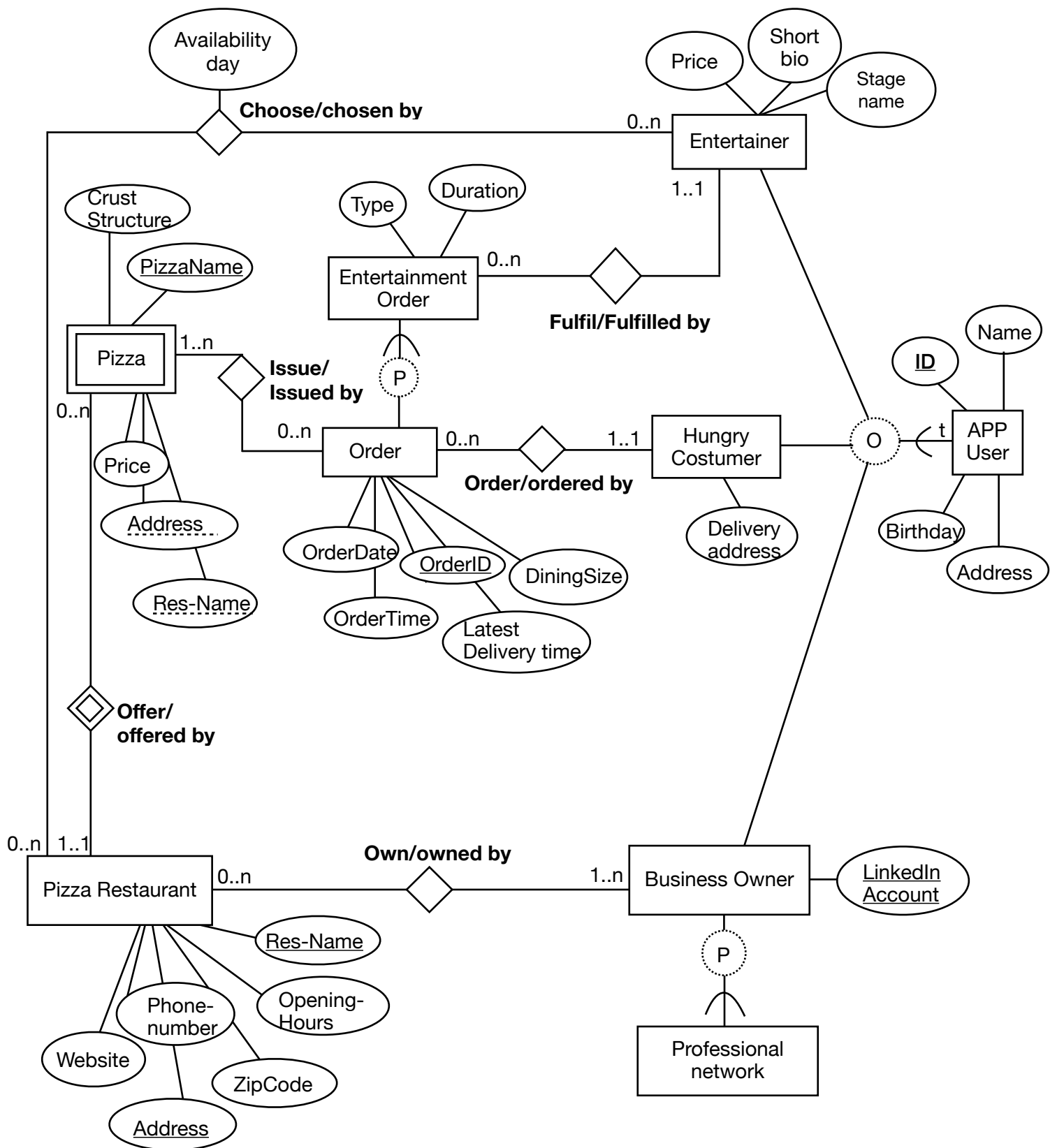
**FACULTY** Bioscience Engineering (Bioinformatics)

## Question1

### Aspects which can not be included in EER model:

1. Temporal constraints can not be modelled. Examples are: add owners to professional network based on their LinkedIn account. Lasted delivery time can not precede OrderTime
2. Can not guarantee consistency across multiple relationship types.  
Examples are: business owner should work in the pizza restaurant(s) he/she owns; entertainer should work in the restaurant he/she chooses to work for; an order can only be assigned to a restaurant which it can supply.
3. It is not possible to specify the set of values that can be assigned to an attribute type. For example, the duration of the entertainment order must be positive.
4. Additionally, EER model can not support the definition of functions. For example, it can not calculate the total order price.

# EER MODEL



## Question2

---

**EMPLOYEE**(Employee ID)

Employee ID: primary key

**EMP\_CERTIFICATES** (CertificateID, CertificateName, *Employee ID*)

This new relation between employee and the certificates has two attribute types: CertificateID, CertificateName, and Employee ID. The first is the primary key. The last one is the foreign key referring to the EMPLOYEE relation. NULL not allowed, on delete/update cascade.

**VENUE**(vName)

vName: primary key

**ASSIGN\_IN**(*Employee ID*, vName, Date)

Employee ID: foreign key refers to Employee ID in relation EMPLOYEE:

NULL not allowed, on delete/update cascade

vName: foreign key refers to vName in relation VENUE: NULL not allowed, on delete/update cascade

Primary key: the combination of EmployeeID and vName

**SHOP**(vName)

vName: primary key refers to vName in relation VENUE

**RESTAURANT**(vName, Capacity)

vName: primary key refers to vName in relation VENUE

**ATTRACTION**(vName, Type, Manufacturer, Date of last inspection)

vName: primary key refers to vName in relation VENUE

**QUEUE**(qID, Waiting time, Status, vName, Queue types)

qID and vName together construct the primary key

vName: foreign key refers to vName in relation ATTRACTION: NULL not allowed, on delete/update cascade

**Queue types** include Stand By Queue, Single Rider Queue, and Fast Pass Queue

**ENTRANCE\_TICKET**(Ticket ID, Guest Name, Valid On, Price Category)

Ticket ID: the primary key

**FAST\_PASS\_TICKET**(Time slot, Ticket ID, *qID*)

Time slot and Ticket ID together construct the primary key

Ticket ID: foreign key refers to in Ticket ID relation in

ENTRANCE\_TICKET. NULL not allowed, on delete/update cascade.

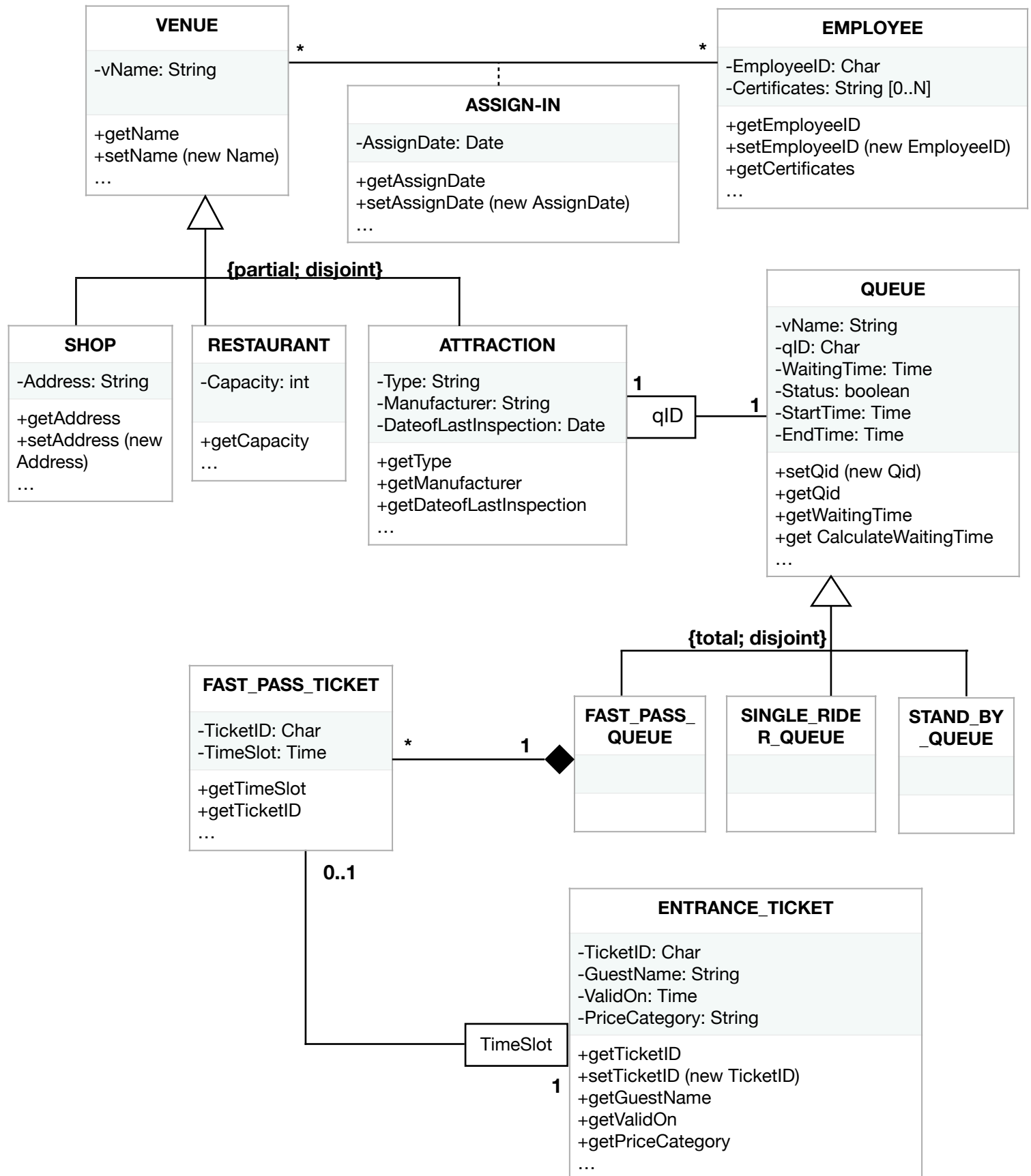
qID: foreign key refers to in qID relation in QUEUE: NULL not allowed, on delete/update cascade.

### **The following semantics were lost in the mapping:**

- The disjointness of the specialisation of Venue to Shop, Restaurant and Attraction can not be enforced.
- The disjointness and completeness the specialisation of queue and queue types(Stand By Queue, Single Rider Queue, and Fast Pass Queue) can not be enforced.
- Attraction having at least one queue can not be enforced.
- Additionally, all four cardinalities can be successfully modelled in the relations between VENUE and EMPLOYEE, FAST\_PASS\_TICKET and FAST\_PASS\_QUEUE, FAST\_PASS\_TICKET and ENTRANCE\_TICKET.
- Examples of extra integrity rules that could be:
  - RESTRICT: the removal or update is halted if referring tuples exist
  - SET NULL (when the foreign key is allowed to be null) or DEFAULT (foreign keys in the referring tuple should be set to their default value)

## Question3

### UML MODEL



## **Some examples of operations that could be added to model extra semantics**

1. Domains with refined variable definitions, such as char, integer, string.....
2. Certificates can be further specified having at most 10, shown as [0..10]
3. Access modifiers (such as '-', '+') are added to specify who can access a variable or method.
4. Use of qualified associations to represent weak entity types to reduce multiplicity, such as the relation between FAST\_PASS\_TICKET and FAST\_PASS\_QUEUE, and the relation between QUEUE and ATTRACTION.
5. The use of composite aggregation to emphasise the dependency and coupling between FAST\_PASS\_QUEUE and FAST\_PASS\_TICKET. Once the FAST\_PASS\_QUEUE does not exist, there won't be any FAST\_PASS\_TICKET.

## Question4

1	<pre>SELECT P.PRODNR, COUNT(S.SUPNR) FROM PRODUCT AS P LEFT OUTER JOIN  SUPPLIES AS S ON (P. PRODNR = S.PRODNR) GROUP BY P.PRODNR</pre>
2	<pre>SELECT PRODNR FROM PO_LINE GROUP BY PRODNR HAVING SUM(QUANTITY) =16</pre>
3	<pre>SELECT P1.PRODNR FROM PRODUCT P1, SUPPLIES S WHERE S.SUPNR=84 AND P1.PRODNR = S.PRODNR AND 5&gt; (SELECT COUNT(*) FROM PRODUCT P2 WHERE P1.PRODNR&lt;P2.PRODNR)</pre>
4	<pre>SELECT PRODNAME FROM PRODUCT WHERE PRODTYPE = (SELECT PRODTYPE FROM PRODUCT GROUP BY PRODTYPE HAVING AVG(AVAILABLE_QUANTITY) &gt;=ALL (SELECT AVG(AVAILABLE_QUANTITY) FROM PRODUCT GROUP BY PRODTYPE))</pre>
5	<pre>SELECT P.PRODNAME, P.PRODNR FROM PRODUCT P WHERE NOT EXISTS (SELECT * FROM SUPPLIES S WHERE P.PRODNR = S.PRODNR)</pre>

