**CS F212 – DATABASE SYSTEMS**

**Hostel Allocation Project [Documentation]**

***By –***

***Tanveer Singh 2020A7PS0084P***

***Utkarsh Darolia 2020A7PS0981P***

1. **System Requirement Specification**
2. **Functional Requirements**

Students should be able to:

* Create a wing and invite their fellow batchmates, or join a wing created by someone else

The wing leader should be able to:

* Give choice of hostel(s) preferred from a set of hostels
* Lock their wing so that others cannot join it until it is unlocked
* Give choice of single/double rooms preferred

Given a set of wings with hostel preferences, the database system should be able to:

* Make sure that each student is in maximum one wing, and there is only one wing leader per wing (consistency requirement which can be checked after every operation performed by the students)
* Allocate hostels and room numbers in such a way that maximum number of wings get their preferred hostels
* Allocate a set of room numbers to the wing, after which the wing members can distribute the rooms amongst themselves
* Randomly allocate the remaining rooms to students that are not in a wing

The administrator should have all privileges to add or delete student records, data about hostels, and so on. These privileges would not be granted to the students.

1. **Data Requirements**

* Information about the students, such as Student\_ID, Name, Phone Number, Year such as first/second etc (to ensure that wings are only created between similar year students)
* Information about the hostels, such as Name, Number of single and double rooms, total capacity
* Data of room numbers available in each hostel, and whether they are single or double rooms
* The rest of the information (such as wing sizes and room numbers allocated) will be filled by the students and then allocated by the DBMS.

1. **Assumptions**

* We assume that all hostels are NOT identical, so some hostels may have more rooms or different room numbers.
* The idea behind the DBMS is to give wings room numbers within a specific range. So, if there is a wing of size 5, they would be granted room numbers from X to X + 4 in the same hostel.
* Rooms are of two types – single or double.
* A wing of single rooms may have a size of maximum 6, and a double wing may have a size of maximum 12.
* Dataset assumptions : We assume a minimal containing 500 students and 3 hostels. The sole girls’ hostel is Meera Bhawan, or “MR”.

1. **System Modelling**

**Comments (may or may not be included in final documentation)**

Relationship between STUDENT and ROOM – Lives In. Cardinality – Each student lives in exactly 1 room. Each room may be occupied by 0 to 2 students. From student side, the participation is total. Some rooms, however, may be empty. (To meet this requirement in the beginning, the room numbers can be randomly allocated to students). Cardinality limits – (0..2) on room side and (1..1) on student side

In STUDENT entity, the Year attribute can be a derived attribute, from ID? But that is only true for BITS ID, may not be true for other years.

Entity HOSTEL – Described by Hostel ID. Other attributes – name, Single rooms, double rooms, capacity.

Is ROOM a weak entity set? Since it cannot exist without a HOSTEL.

HOSTEL Relationships – Make ROOM a weak entity set of HOSTEL.

Fourth Entity – WING. Represented by a unique WING ID. Can also be represented by Student ID of leader. Will have a complex attribute initially (multi valued) – Member ID {multivalued}. Another multivalued attribute – Preferred Hostel. Room Type – Single or Double.

ROOM is a weak entity set, it is owned by HOSTEL.

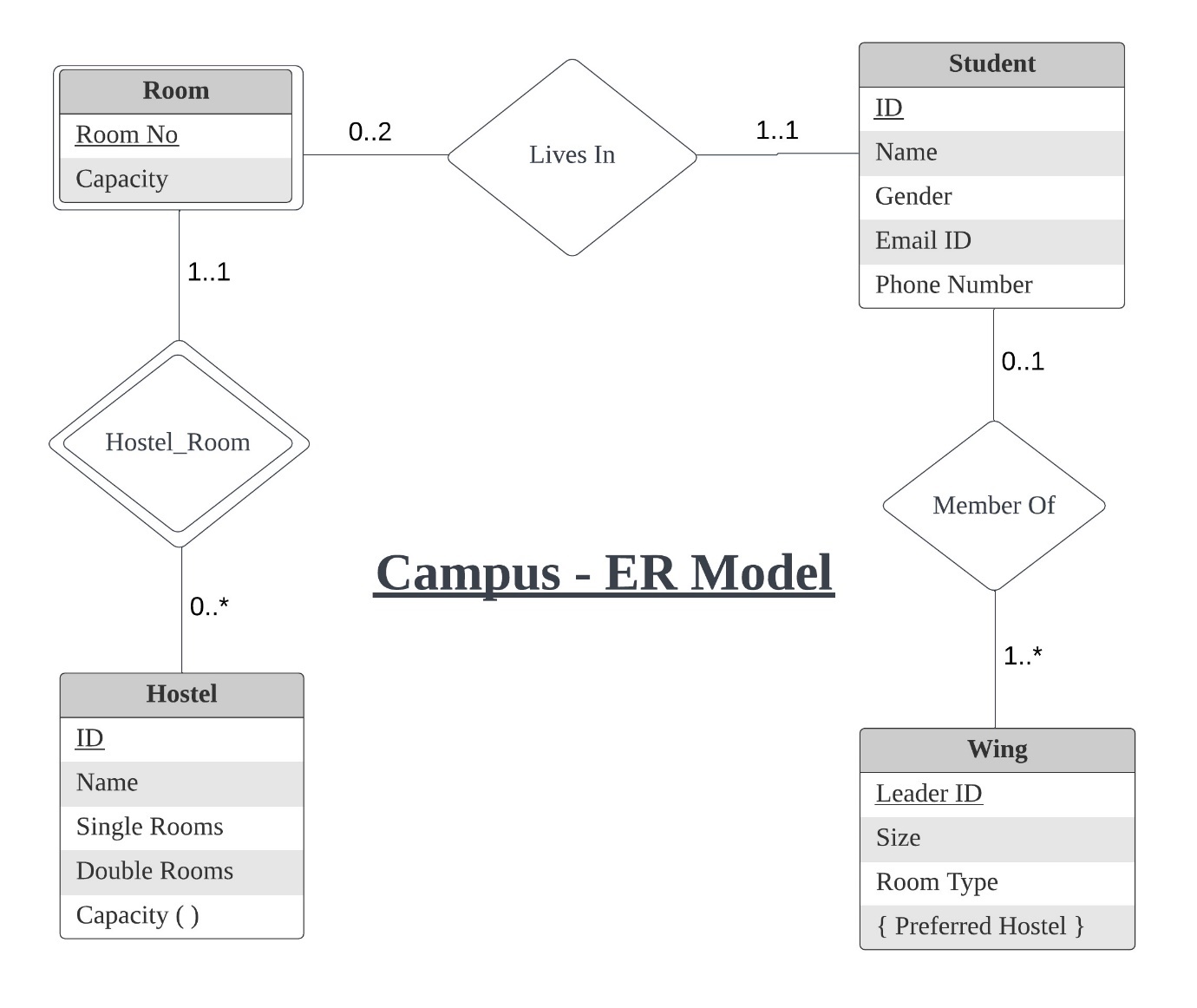
Relationship between STUDENT and WING – Member of. A student may be a member of 0 to 1 wing. A wing may have 1 or more students. (Max limit? Currently Infinity).

// TODO: Remove capacity, as it is a derived attribute. Derived attributes are not explicitly represented in the relational data model. However,

they can be represented as stored procedures, functions, or methods in other data models.

//TODO : MALES AND FEMALES IN DIFFERENT HOSTELS!

1. **E-R Diagram**



**Description of Relationships**

* The LIVES IN relationship is between Student and Room. A Student lives in one and only one Room. The participation of Student in this relationship is total. A Room may have 0, 1 or 2 occupants (2 is for double rooms only).
* The HOSTEL\_ROOM relationship is the identifying relationship for the weak entity set Room. The identifying entity set is Hostel. The discriminator attribute is Room No.
* The MEMBER OF relationship is between Student and Wing. A Student may be a member of at most one Wing, however, the participation is not total because a Student may not be in any Wing. A Wing can have multiple students.

1. **Schema Design**

The schemas derived from the E-R diagram in 2(a) are as follows:

* *Student (Student\_ID, Name, Gemder, Email\_ID, Phone\_Number)*
* *Hostel (Hostel\_ID, Name, Single\_rooms, Double\_rooms)*

In hostel, *Capacity* is a derived attribute and will be expressed as a procedure.

* *Wing (Leader\_ID, Size, Room\_type)*
* *Wing\_hostel (Leader\_ID, Preferred\_Hostel)*
* *Room (Hostel\_ID, Room\_No, Capacity)*
* *Lives\_in (Student­\_ID, Hostel\_ID, Room\_No)*
* *Member\_of (Student\_ID, Leader\_ID)*

1. **Data Normalisation**

We will make the necessary changes to convert this schema into BCNF. For a relation R to be in BCNF, every non-trivial functional dependency of the form A🡪B must have A as a super key of R.

* In relation *Student,* there are 3 minimal candidate keys – *Email\_ID, Phone\_Number* and *Student\_ID.* All functional dependencies will have either of these 3 on the left side. Name and Gender00 cannot uniquely determine any other attributes.
* In relation *Hostel,* there are 2 minimal candidate keys – *Hostel\_ID* and *Name.* Apart from these, no other attribute can uniquely determine the other attributes.
* In all other relations, there is one unique candidate key which determines the other attributes.

The entire schema is already in BCNF as every non-trivial dependency of the form A🡪B has A as a super key.

Furthermore, in every schema R, all non-trivial multivalued dependencies of the form A🡪B have A as a super key of R. Hence, the entire schema is also in 4NF.

1. **List of Tables Required**