STUDENT RESULT MANAGEMENT SYSTEM

Created by:

HARSHIT GAJIPARA BHAVYA BHUT

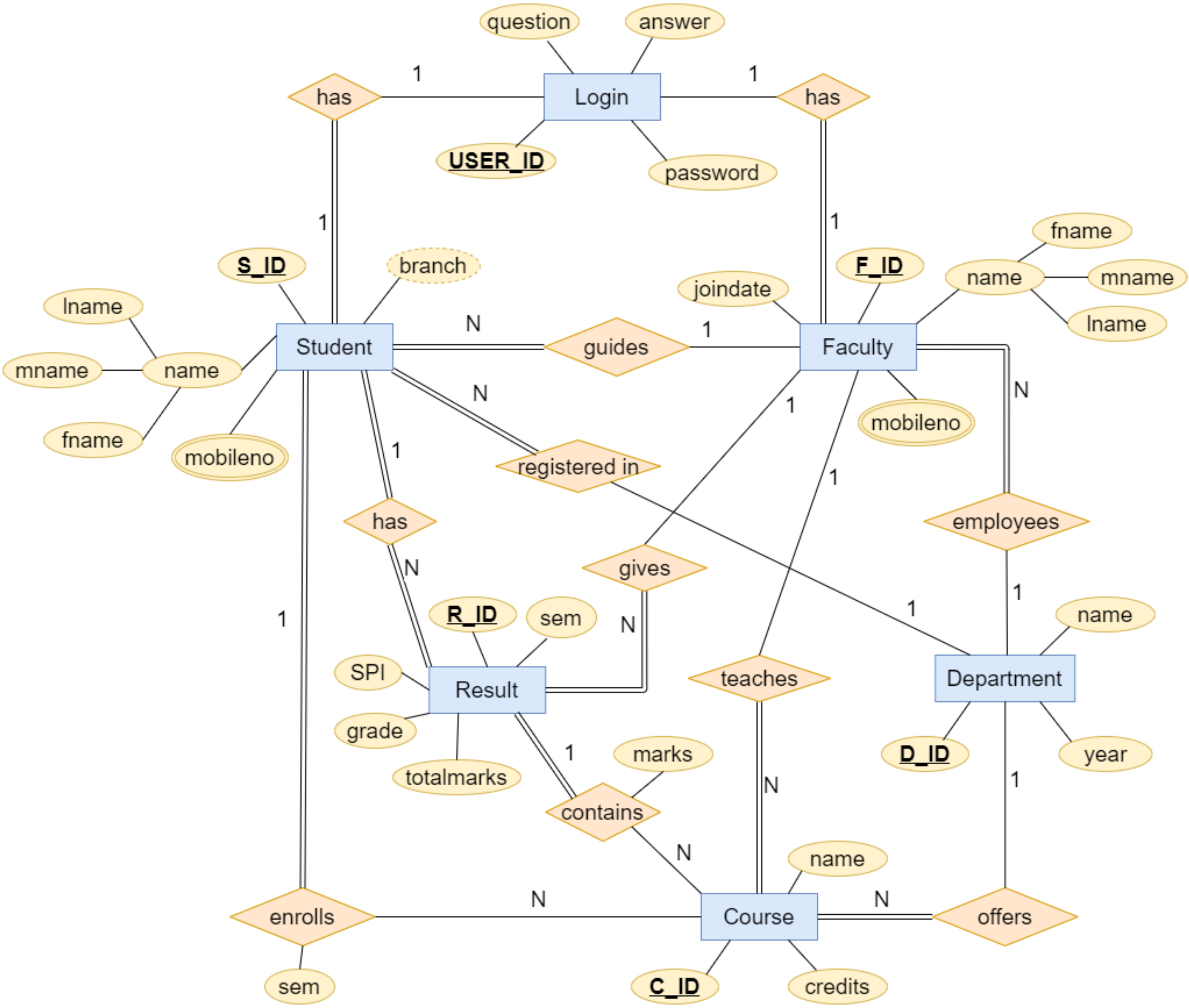
FUNCTIONAL REQUIREMENTS:

1. Application made using python program should provide graphical user interface for this database system.
2. Verification of password and Authentication of user when he/she tries to log into the system.
3. Authorization of user whenever user log into system and provide permissions accordingly.
4. Insert, search and update data records of students and faculty in system.
5. Generate result using combination of courses, marks, semester of student.
6. Student can check his/her rank throughout the department in any particular course.
7. Provides interface to install result as pdf/image file on computer or laptop and view it outside of the application.

Roles:

Admin – can perform above all functions and acts as system administrator
Student – can change own password, view and print own result
Faculty – can change own password, view, modify and print result of any student under same department

E-R DIAGRAM:


question
answer
has
1
Login
1
has
USER_ID
password
1
1
S_ID
branch
joindate
F_ID
fname
name
mname
lname
lname
mname
name
Student
N
guides
1
Faculty
fname
N
N
mobileno
1
mobileno
registered in
1
1
has
employees
N
gives
1
1
N
R_ID
sem
name
SPI
Result
teaches
Department
grade
1
marks
D_ID
year
totalmarks
contains
N
1
N
name
enrolls
N
Course
N
offers
sem
C_ID
credits

ENTITIES:

As shown in ER diagram, this database has 6 entities:

- Student
- Faculty
- Course
- Department
- Login
- Result

RELATIONSHIPS:

Relations between different entities in database are as below:
i.e. RelationshipName (Entity1, Entity2)

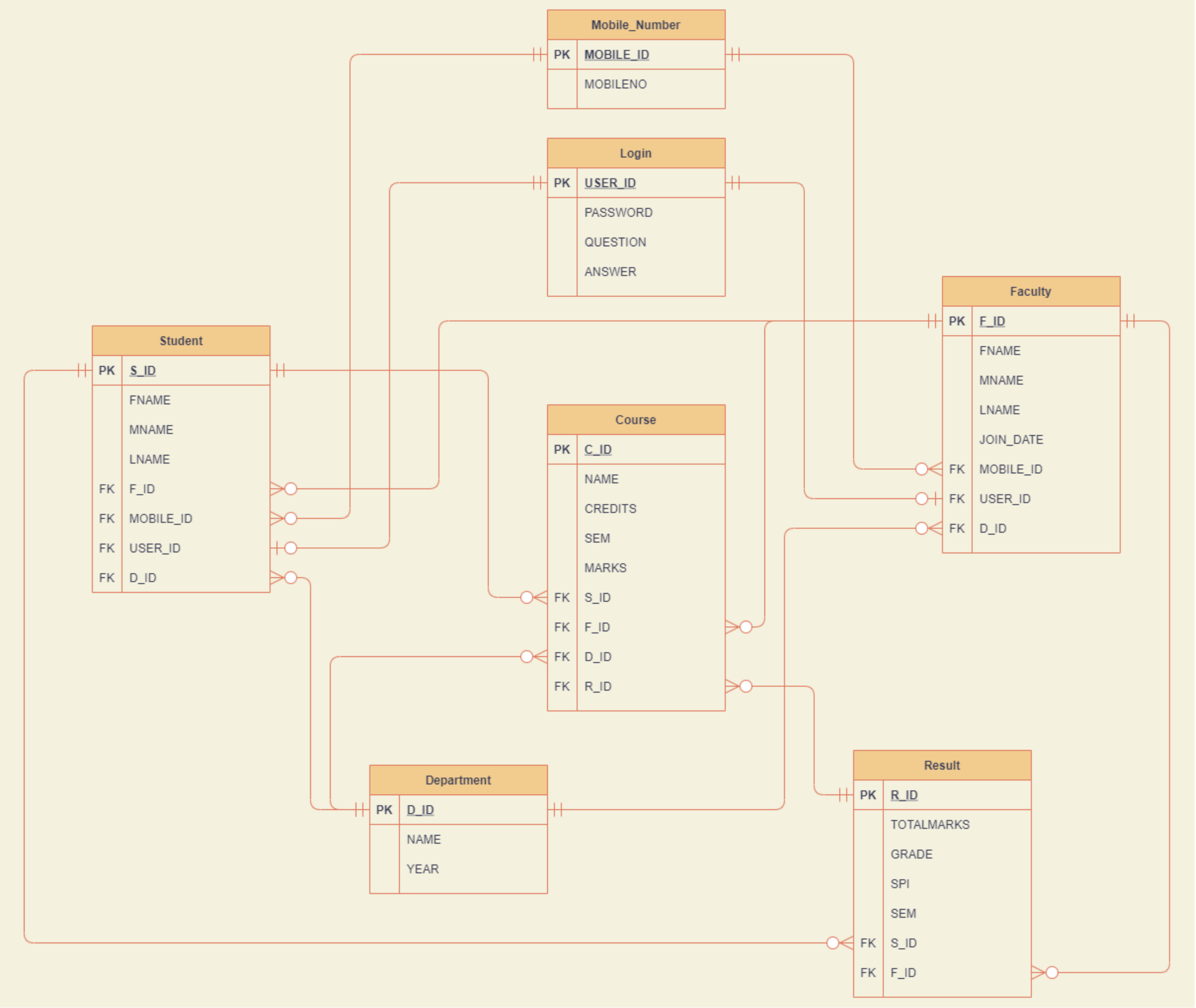
One to Many Type Relationships:

- Has (Student, Result)
- Enrolls (Student, Course)
- Guides (Faculty, Student)
- Registered_In (Student, Department)
- Gives (Faculty, Result)
- Teaches (Faculty, Course)
- Employees (Department, Faculty)
- Offers (Department, Course)
- Contains (Result, Course)

One to One Type Relationships:

- Has (Student, Login)
- Has (Faculty, Login)

RELATIONAL MODEL:



Relational Schema:

- Relational schema is represented as entities with their attributes. Relational schema of this database is as follow:

  - ***Student*** (s_id,fname,mname,lname,f_id,mobile_id,user_id,d_id)
  - ***Faculty*** (f_id,fname,mname,lname,join_date,mobile_id,user_id,d_id)
  - ***Course*** (c_id,name,credits,sem,marks,s_id,f_id,d_id,r_id)
  - ***Department*** (d_id,name,year)
  - ***Result*** (r_id,totalmarks,grade,SPI,sem,s_id,f_id)
  - ***Login*** (user_id,password,question,answer)

- Now Student and Faculty entity have multiple-value attribute Mobile_Number, so to represent it we need one more table.

- ***Mobile_Number*** (mobile_id,mobileno)

Using above schemas and after adding different constraints to system, we can create student result database management system. This system, implemented with help of GUI by python can then perform different tasks and fulfill its functional and non-functional requirements.