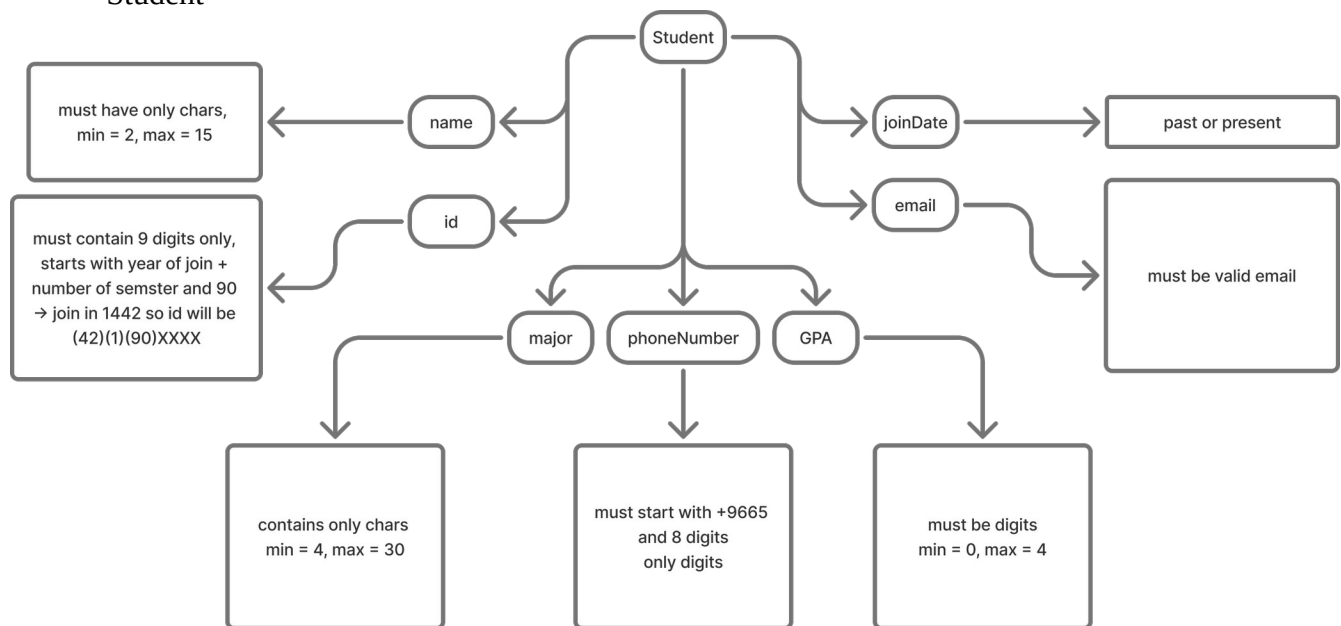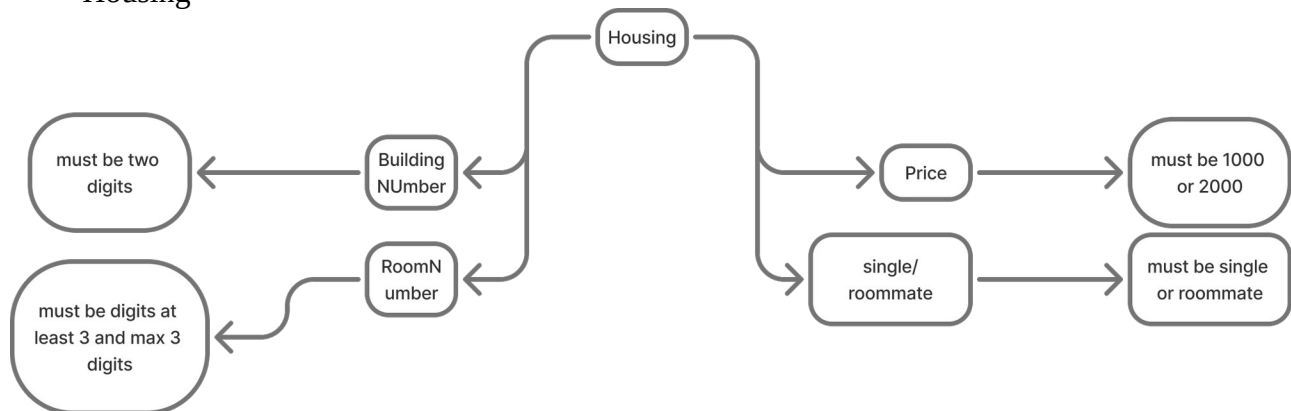It is Jubail Industrial college edugate it has student, instructor, users, housing system etc I chose this web site to do this diagrams and do validate to each case
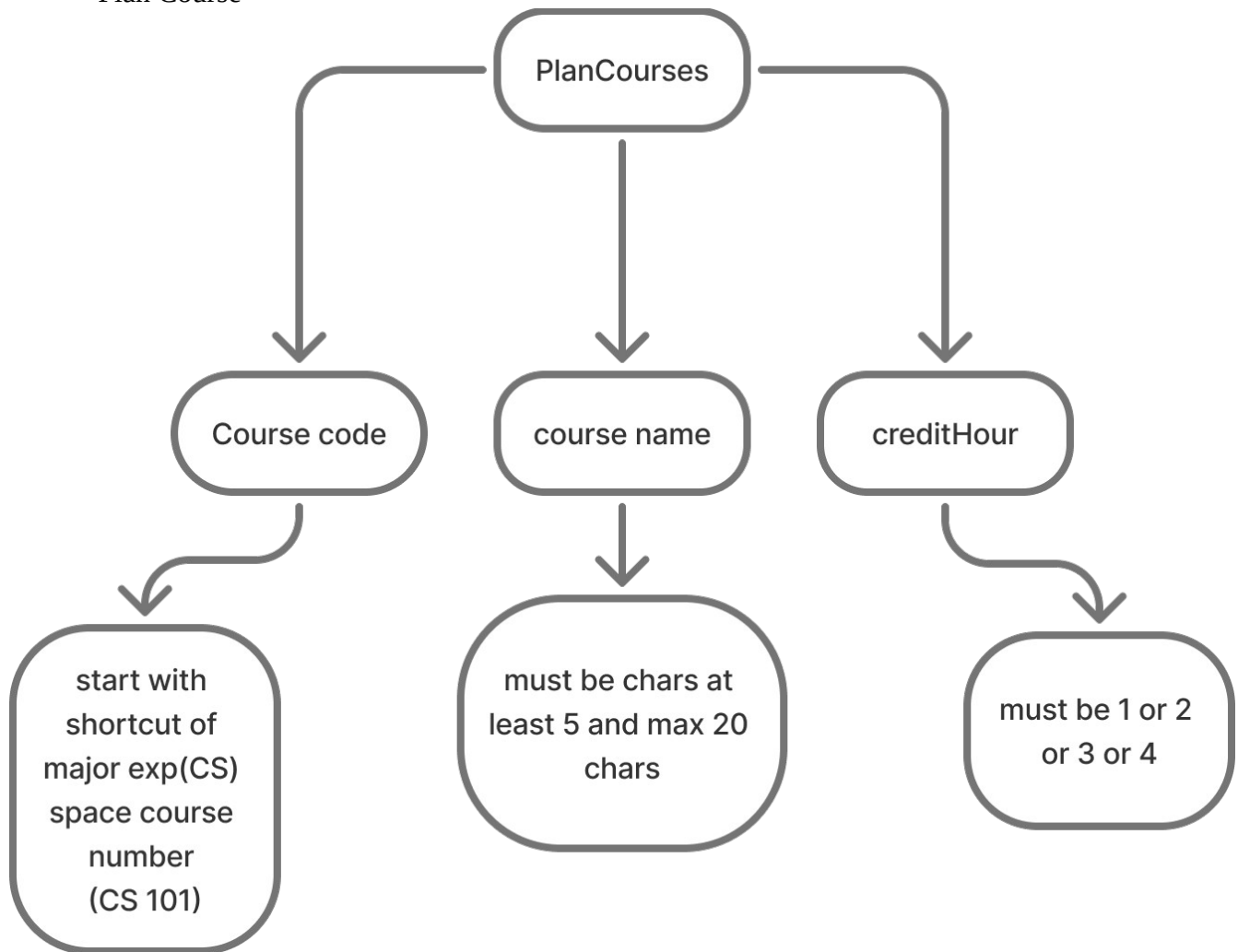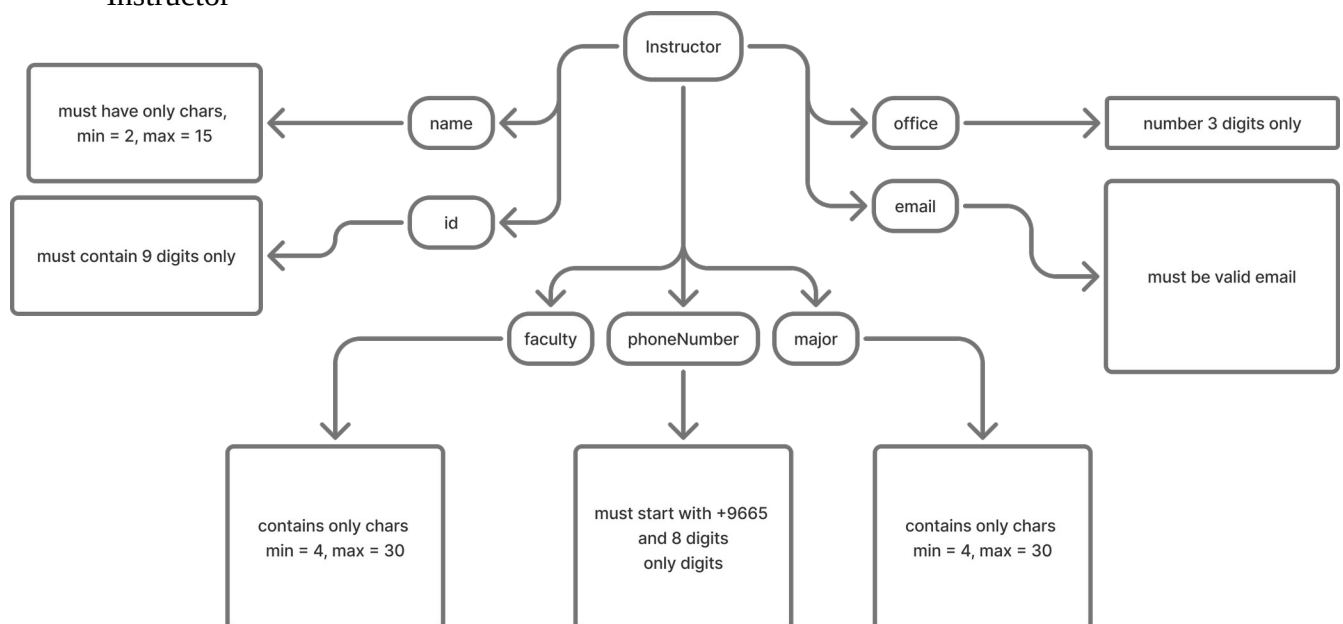
Diagrams for each entity
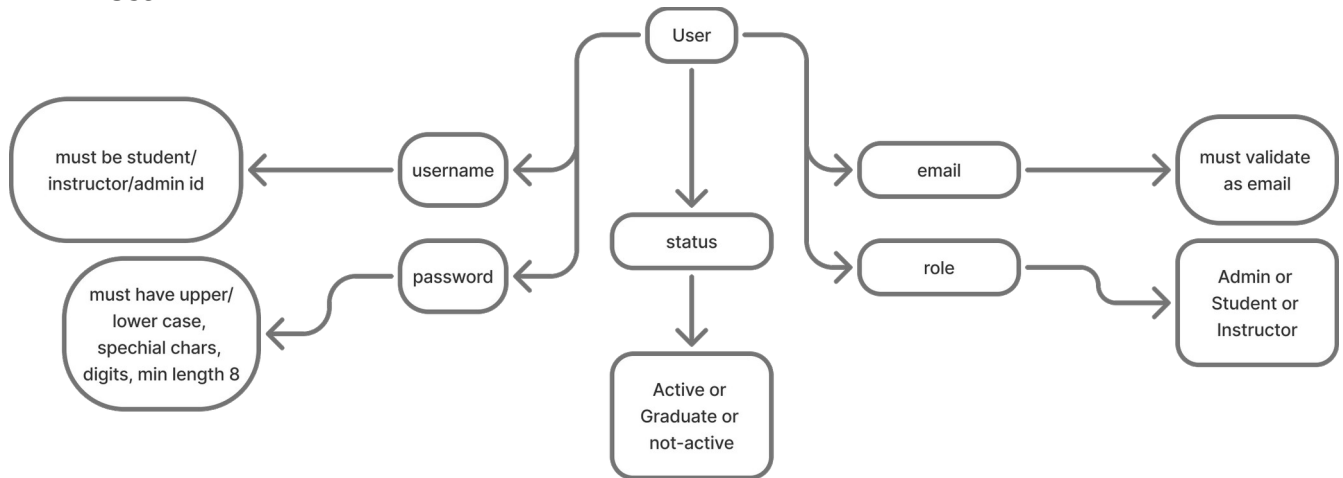
- Student

- Housing

- Plan Course

```
                          ┌─────────────────┐
                          │   PlanCourses   │
                          └─────────────────┘
              ┌──────────────────┼──────────────────┐
              ▼                  ▼                  ▼
      ┌──────────────┐   ┌──────────────┐   ┌──────────────┐
      │ Course code  │   │ course name  │   │  creditHour  │
      └──────────────┘   └──────────────┘   └──────────────┘
              ▼                  ▼                  ▼
      ┌──────────────┐   ┌──────────────┐   ┌──────────────┐
      │  start with  │   │ must be chars│   │ must be 1 or 2│
      │  shortcut of │   │ at least 5   │   │  or 3 or 4   │
      │ major exp(CS)│   │ and max 20   │   │              │
      │ space course │   │    chars     │   │              │
      │    number    │   │              │   │              │
      │   (CS 101)   │   │              │   │              │
      └──────────────┘   └──────────────┘   └──────────────┘
```

- Instructor

```
                                 ┌──────────────┐
                                 │  Instructor  │
                                 └──────────────┘
  ┌────────────────────┐       ┌────────┐
  │ must have only chars,│◄────│  name  │           ┌────────┐        ┌──────────────────┐
  │  min = 2, max = 15   │     └────────┘           │ office │───────►│ number 3 digits only │
  └────────────────────┘                            └────────┘        └──────────────────┘
  ┌────────────────────┐       ┌────────┐           ┌────────┐        ┌──────────────────┐
  │ must contain 9      │◄────│   id   │           │ email  │        │                  │
  │ digits only         │     └────────┘           └────────┘───────►│ must be valid email│
  └────────────────────┘                                             │                  │
                         ┌─────────┬──────────────┬─────────┐        └──────────────────┘
                         ▼         ▼              ▼
                   ┌─────────┐┌─────────────┐┌───────┐
                   │ faculty ││ phoneNumber ││ major │
                   └─────────┘└─────────────┘└───────┘
                         ▼              ▼              ▼
               ┌──────────────┐ ┌──────────────┐ ┌──────────────┐
               │contains only │ │must start with│ │contains only │
               │   chars      │ │ +9665        │ │   chars      │
               │min = 4, max  │ │ and 8 digits │ │min = 4, max  │
               │   = 30       │ │ only digits  │ │   = 30       │
               └──────────────┘ └──────────────┘ └──────────────┘
```

- User

User

must be student/
instructor/admin id

username

must have upper/
lower case,
spechial chars,
digits, min length 8

password

status

Active or
Graduate or
not-active

email

must validate
as email

role

Admin or
Student or
Instructor

Student {

@Pattern(regexp = "^[A-Za-z]{2,15}$", message = "Name must be only letters, 2 to 15 characters")

@NotEmpty(message = "Name is required")

   private String name;

@Pattern(regexp = "^[0-9]{9}$", message = "ID must be exactly 9 digits")

@Pattern(regexp = "^\\d{2}[1-3]90\\d{4}$", message = "ID must follow format: YY+Semester+90+XXXX")

@NotEmpty(message = "Name is required")

   private String id;

@Pattern(regexp = "^[A-Za-z]{4,30}$", message = "Major must be only letters, 4 to 30 characters")

@NotEmpty(message = "Name is required")

   private String major;

@Pattern(regexp = "^\\+9665\\d{8}$", message = "Phone must start with +9665 and be followed by 8 digits")

@NotEmpty(message = "Name is required")

   private String phoneNumber;

@DecimalMin(value = "0.0", message = "GPA must be at least 0.0")

@DecimalMax(value = "4.0", message = "GPA must be at most 4.0")

@NotNull(message = "GPA is required")

   private Double GPA;

@Email(message = "Invalid email format")

@NotEmpty(message = "Email is required")

   private String email;

@PastOrPresent(message = "Join date must be in the past or present")

@NotNull(message = "Join date is required")

   private LocalDate joinDate;

}

Housing {

@Pattern(regexp = "^\\d{2}$", message = "Building number must be exactly 2 digits")

@NotEmpty(message = "Building number is required")

private String buildingNumber;

@Pattern(regexp = "^\\d{3}$", message = "Room number must be exactly 3 digits")

@NotEmpty(message = "Room number is required")

private String roomNumber;

@Pattern(regexp = "^(1000|2000)$", message = "Price must be 1000 or 2000")

@NotEmpty(message = "Price is required")

  private String price;

  @Pattern(regexp = "^(single|roommate)$", message = "Must be either 'single' or 'roommate'")

@NotEmpty(message = "Type is required")

  private String type;

}

PlanCourses {

  @Pattern(regexp = "^[A-Z]{2,5} \\d{3}$", message = "Course code must be like 'CS 101'")

@NotEmpty(message = "Course code is required")

  private String courseCode;

    @Pattern(regexp = "^[A-Za-z]{5,20}$", message = "Course name must be 5 to 20 alphabetic characters")

@NotEmpty(message = "Course name is required")

  private String courseName;

  @Min(value = 1, message = "Credit hours must be between 1 and 4")

  @Max(value = 4, message = "Credit hours must be between 1 and 4")

@NotNull(message = "Credit hours is required")

  private int creditHours;

}

Instructor {

@Pattern(regexp = "^[A-Za-z]{2,15}$", message = "Name must be only letters, 2 to 15 characters")
@NotEmpty(message = "Name is required")
   private String name;

@Pattern(regexp = "^\\d{9}$", message = "ID must be exactly 9 digits")
@NotEmpty(message = "Id is required")
   private String id;

@Pattern(regexp = "^[A-Za-z]{4,30}$", message = "Faculty must be only letters, 4 to 30 characters")
@NotEmpty(message = "Faculty is required")
   private String faculty;

@Pattern(regexp = "^\\+9665\\d{8}$", message = "Phone must start with +9665 and be followed by 8 digits")
@NotEmpty(message = "Phone number is required")
   private String phoneNumber;

@Pattern(regexp = "^[A-Za-z]{4,30}$", message = "Major must be only letters, 4 to 30 characters")
@NotEmpty(message = "Major is required")
   private String major;

@Email(message = "Invalid email format")
@NotEmpty(message = "Email is required")
   private String email;

@Pattern(regexp = "^\\d{3}$", message = "Office number must be exactly 3 digits")
@NotEmpty(message = "Office number is required")
   private String office;
}

User {

@Pattern(regexp = "^\\d{9}$", message = "Username must be a 9-digit ID")
@NotEmpty(message = "Username is required")
private String username;

@Pattern(
regexp = "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@$!%*?&])[A-Za-z\\d@$!%*?&]{8,}$",
message = "Password must have uppercase, lowercase, digit, special character, and at least 8 characters"
)
@NotEmpty(message = "Password is required")
private String password;

@Pattern(
regexp = "^(Active|Graduate|not-active)$",
message = "Status must be either Active, Graduate, or not-active"
)
@NotEmpty(message = "Status is required")
private String status;

@Email(message = "Invalid email format")
@NotEmpty(message = "Email is required")
private String email;

@Pattern(
regexp = "^(Admin|Student|Instructor)$",
message = "Role must be either Admin, Student, or Instructor"
)
@NotEmpty(message = "Role is required")
private String role;
}