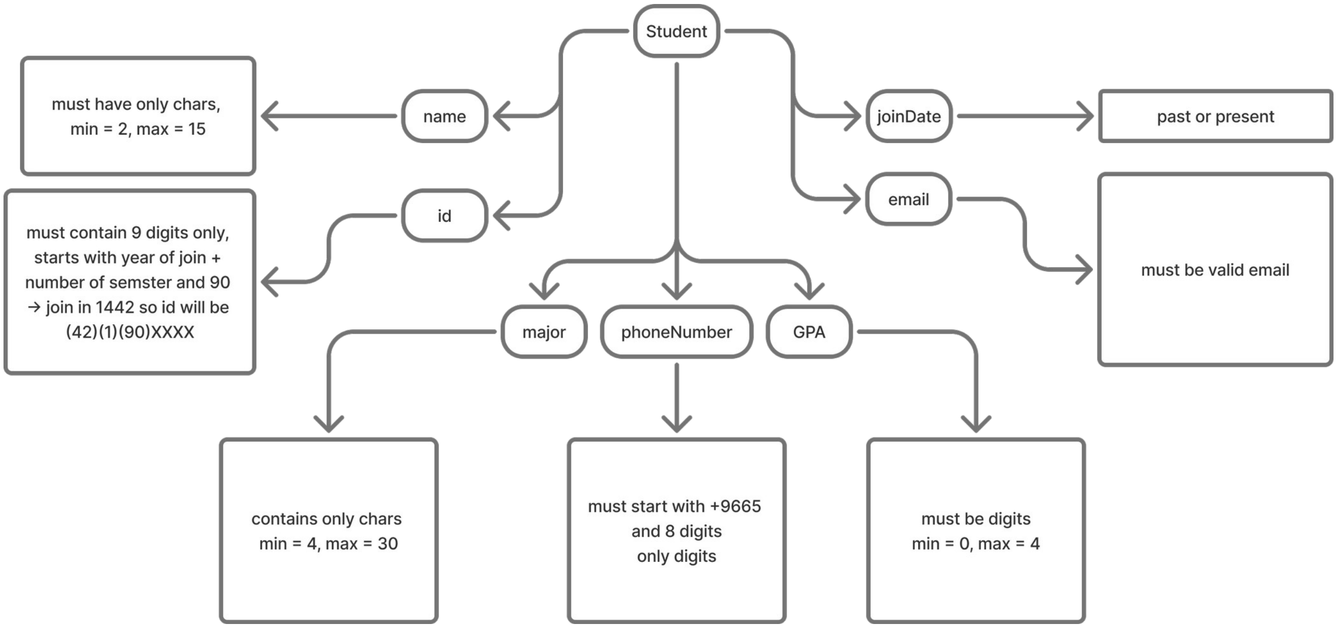
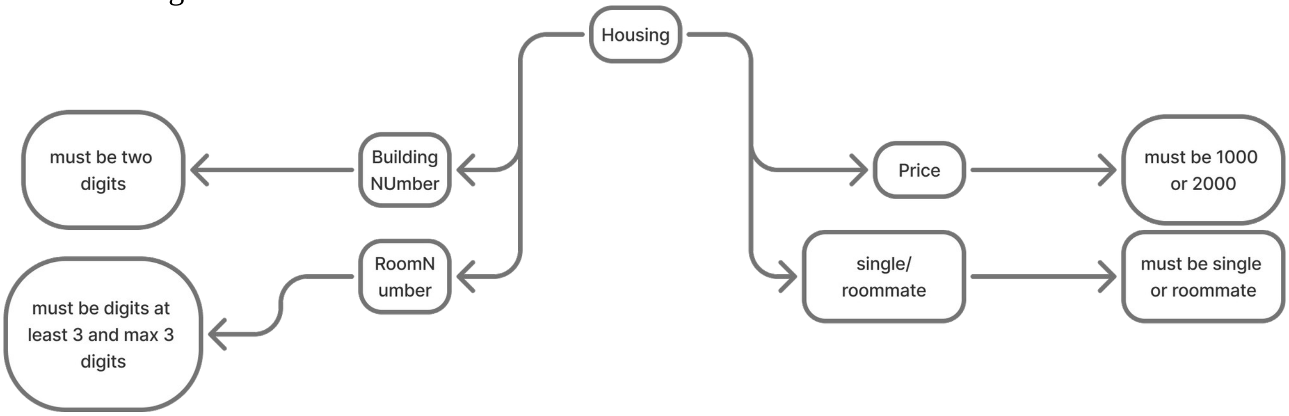
Diagrams for each entity

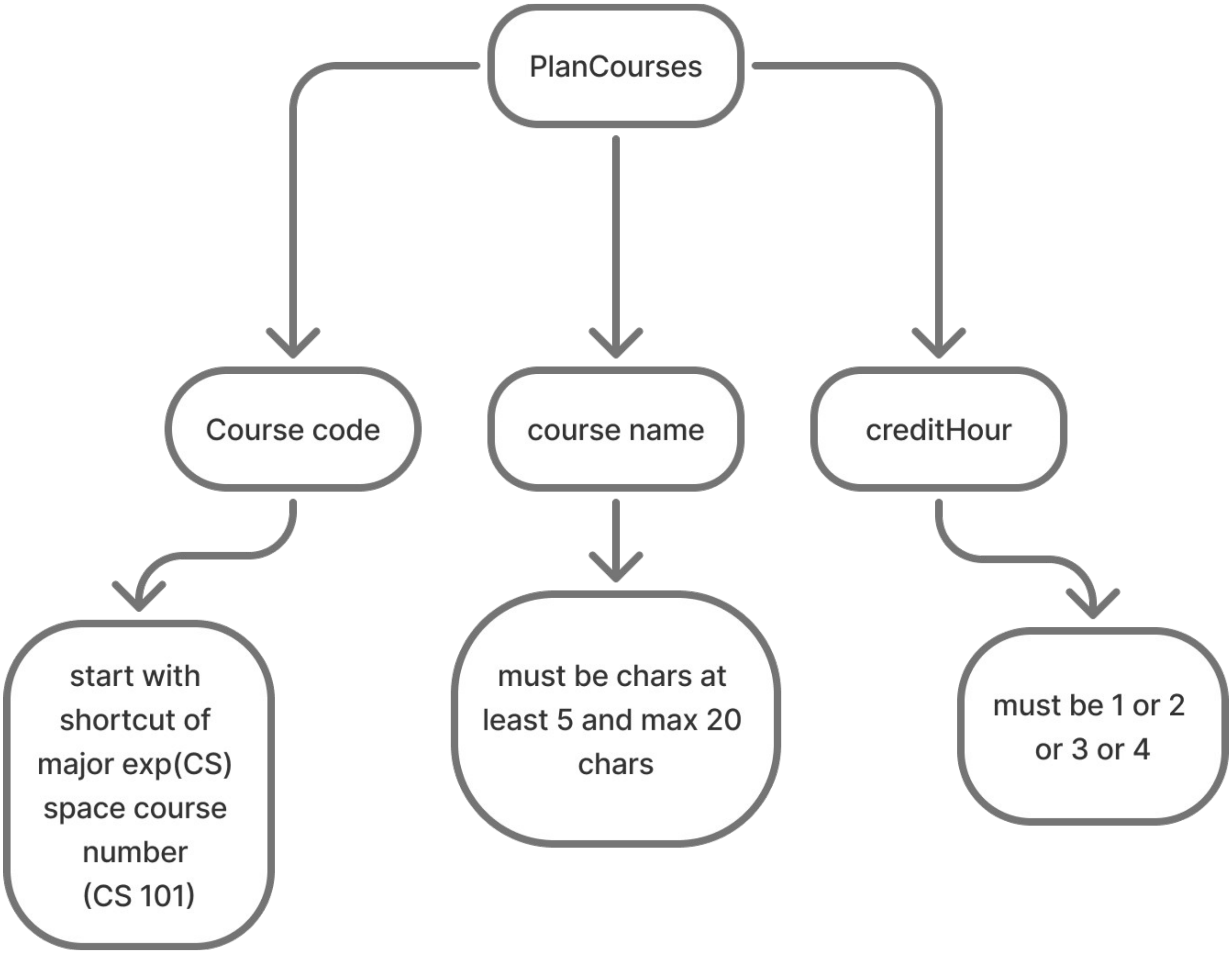
- Student


Student
name
must have only chars, min = 2, max = 15
id
must contain 9 digits only, starts with year of join + number of semster and 90 → join in 1442 so id will be (42)(1)(90)XXXX
joinDate
past or present
email
must be valid email
major
contains only chars min = 4, max = 30
phoneNumber
must start with +9665 and 8 digits only digits
GPA
must be digits min = 0, max = 4

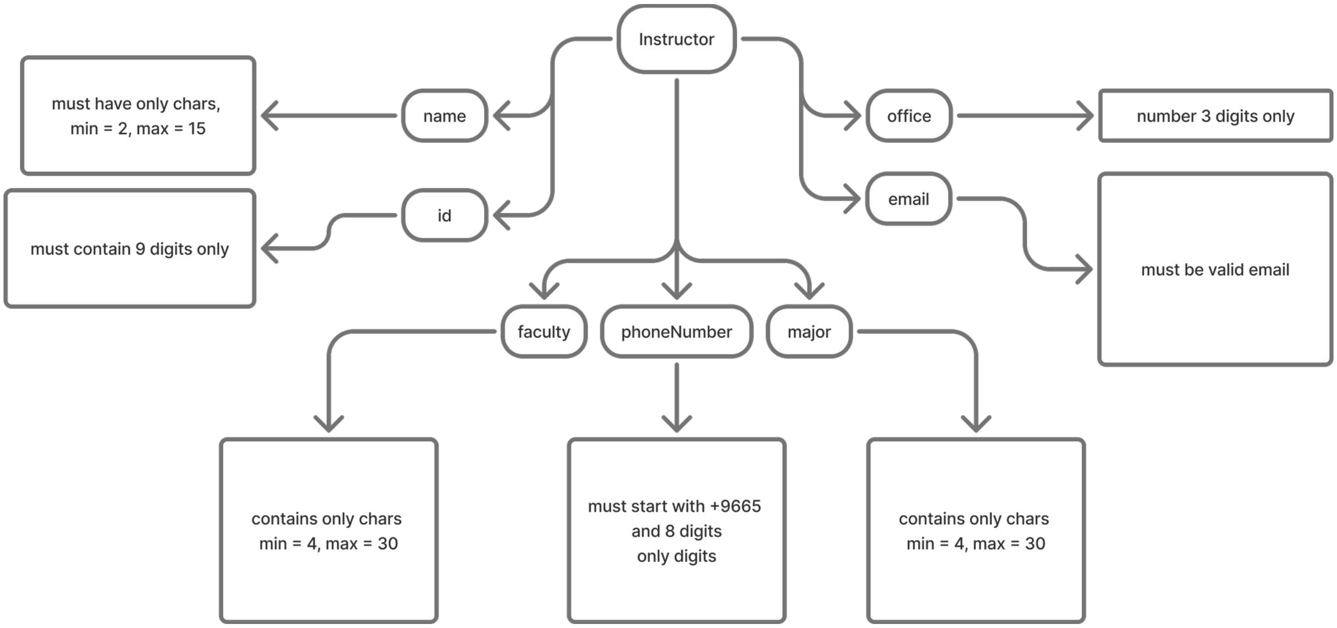
- Housing



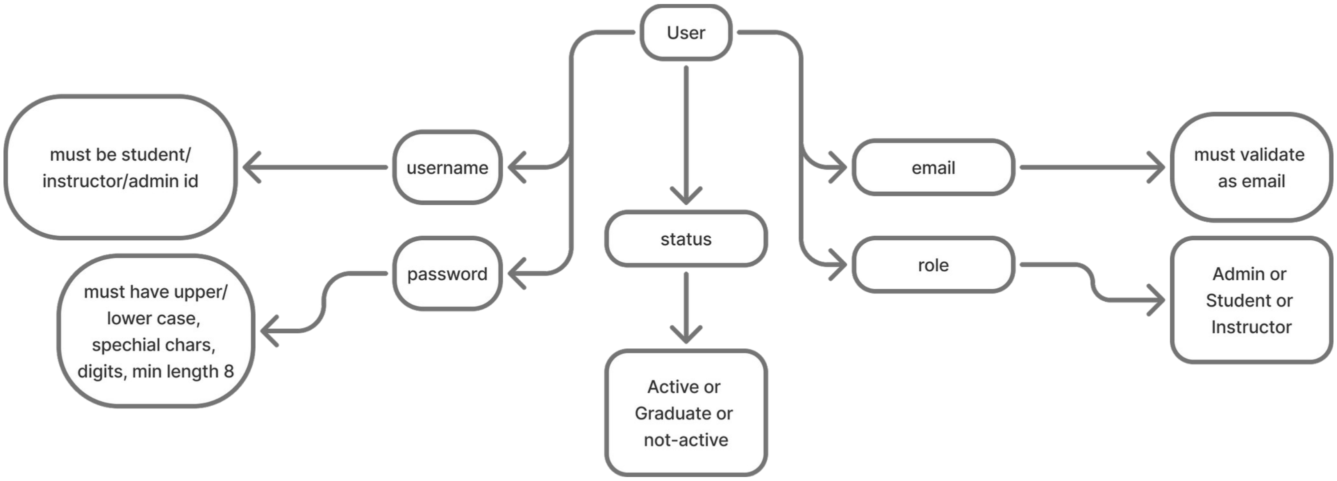
- Plan Course


PlanCourses
Course code
course name
creditHour
start with shortcut of major exp(CS) space course number (CS 101)
must be chars at least 5 and max 20 chars
must be 1 or 2 or 3 or 4

- Instructor


Instructor
name
must have only chars, min = 2, max = 15
id
must contain 9 digits only
office
number 3 digits only
email
must be valid email
faculty
contains only chars min = 4, max = 30
phoneNumber
must start with +9665 and 8 digits only digits
major
contains only chars min = 4, max = 30

- User


User
username
must be student/ instructor/admin id
password
must have upper/ lower case, spechial chars, digits, min length 8
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")
    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")
    private String id;

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

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

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

    @PastOrPresent(message = "Join date must be in the past or present")
    private LocalDate joinDate;
}
Housing {
    @Pattern(regexp = "^\\d{2}$", message = "Building number must be exactly 2 digits")
    private String buildingNumber;
    @Pattern(regexp = "^\\d{3}$", message = "Room number must be exactly 3 digits")
    private String roomNumber;
    @Pattern(regexp = "^(1000|2000)$", message = "Price must be 1000 or 2000")
    private String price;
    @Pattern(regexp = "^(single|roommate)$", message = "Must be either 'single' or 'roommate'")
    private String type;
}
```

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

     @Pattern(regexp = "^[A-Za-z ]{5,20}$", message = "Course name must be 5 to 20 alphabetic characters")
    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")
    private int creditHours;
}
Instructor {
@Pattern(regexp = "^[A-Za-z]{2,15}$", message = "Name must be only letters, 2 to 15 characters")
    private String name;

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

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

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

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

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

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

```
User {

// must be 9 digits only (ID of student/instructor/admin)
@Pattern(regexp = "^\\d{9}$", message = "Username must be a 9-digit ID")
private String username;

// password must have uppercase, lowercase, special char, digit, min 8 characters
@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"
    )
private String password;

    // must be one of the allowed values
@Pattern(
regexp = "^(Active|Graduate|not-active)$",
message = "Status must be either Active, Graduate, or not-active"
    )
private String status;

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

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