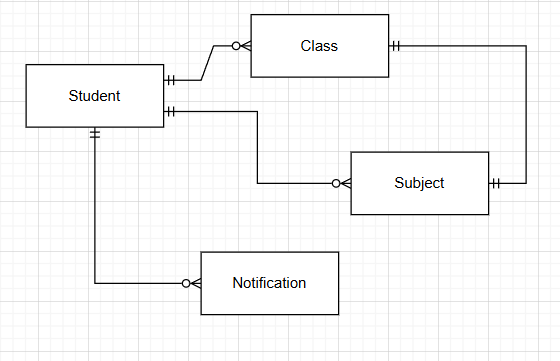
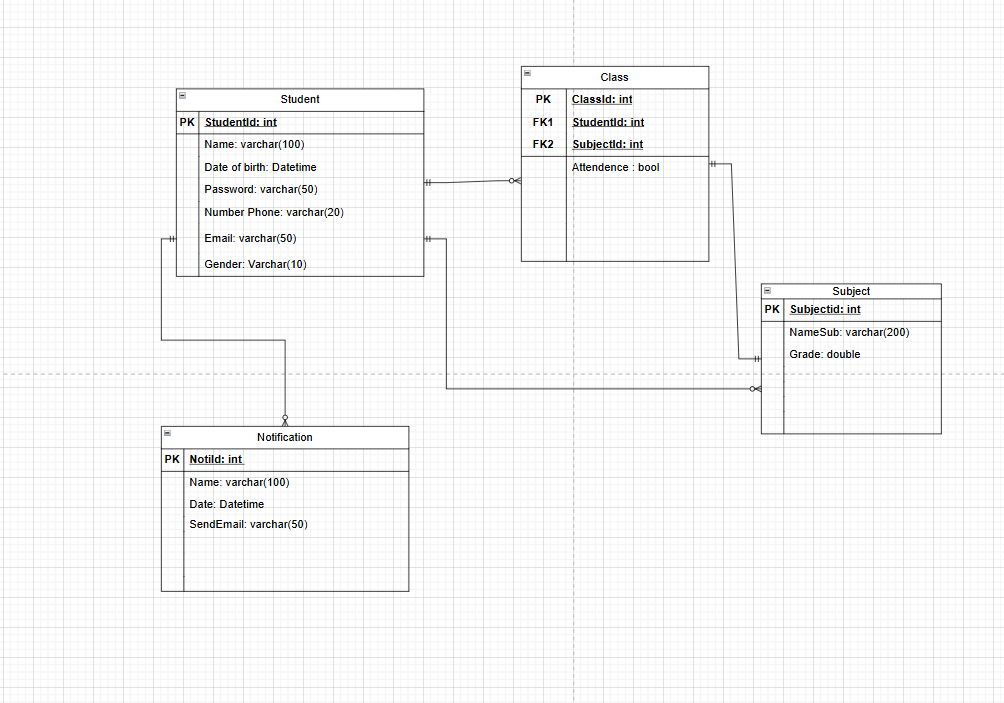
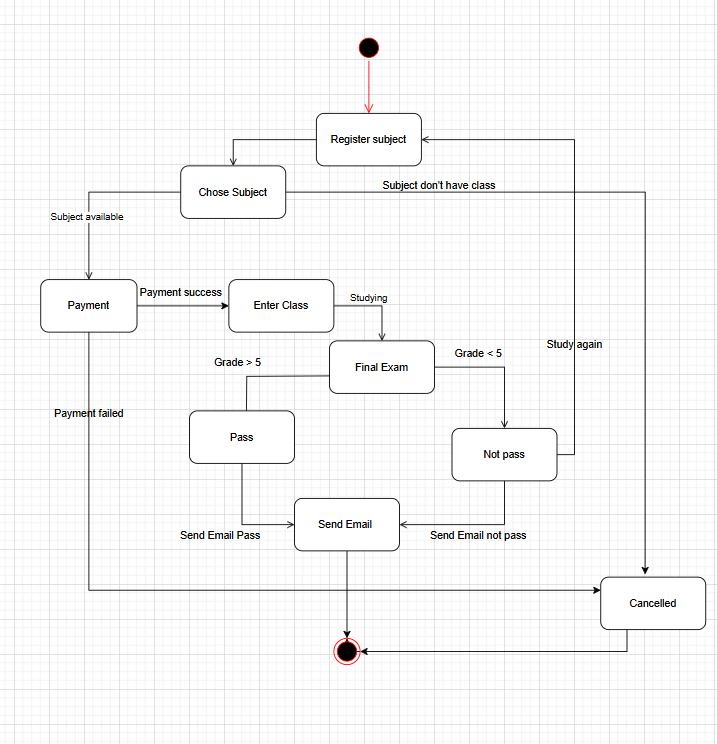
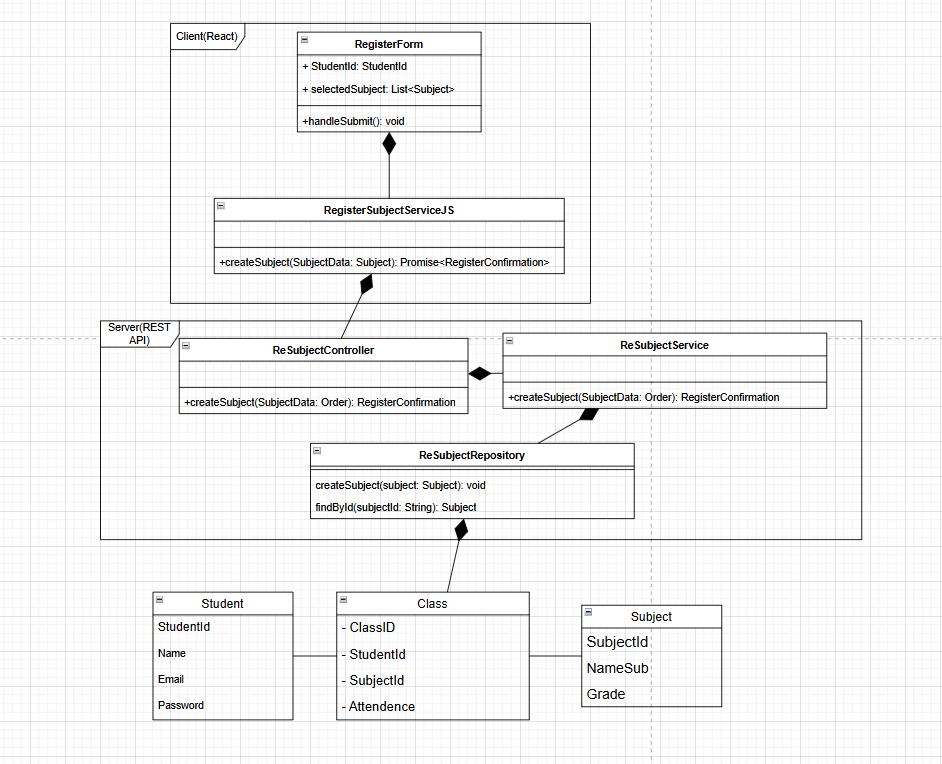
Part 1  
1.1 List the main entities  
  
  
Student: StudentId[PK], Name, Date of birth, Number Phone, Gender, Email.

Subject: SubjectId[PK], NameSub, Grade  
  
Class: ClassId[PK], SubjectId[FK], StudentId[FK], Attendence  
  
Notification: NotiId[PK], Date, SendEmail  
  
1.2 Conceptual ERD  
  


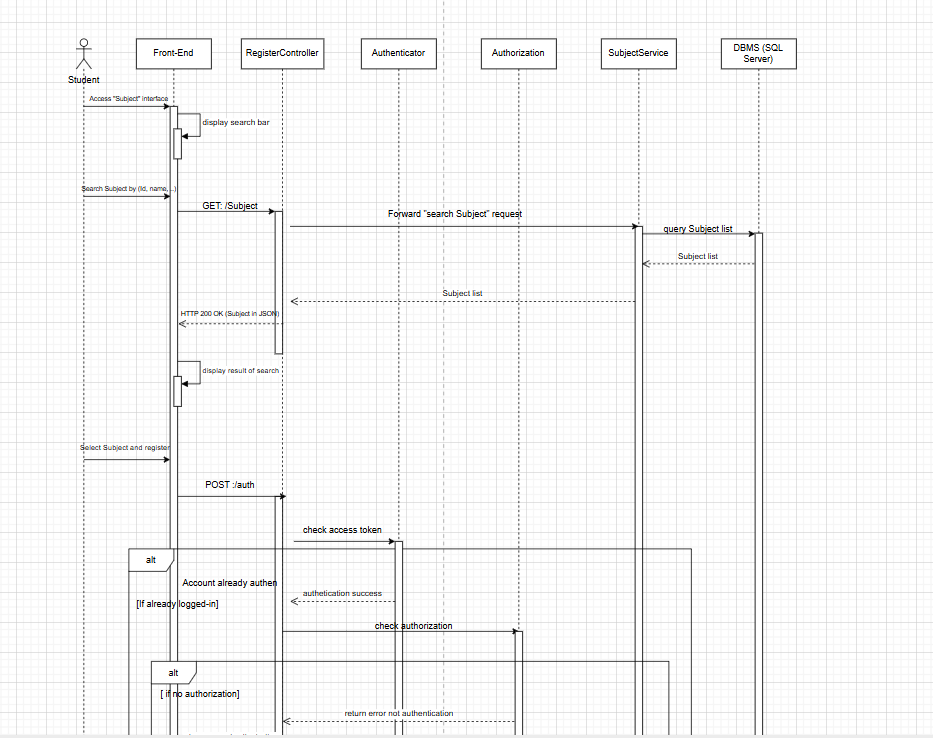
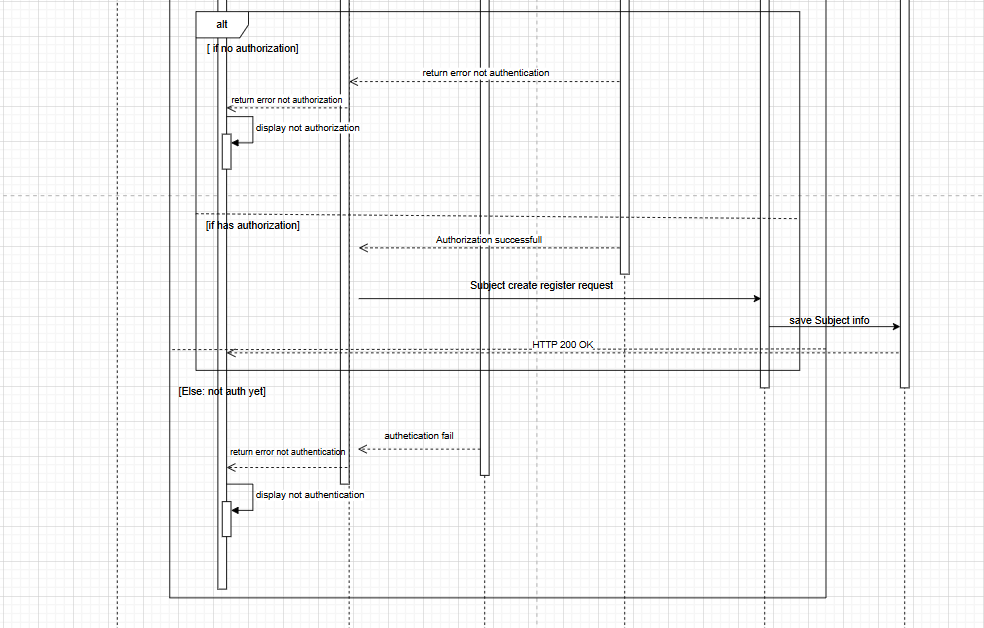
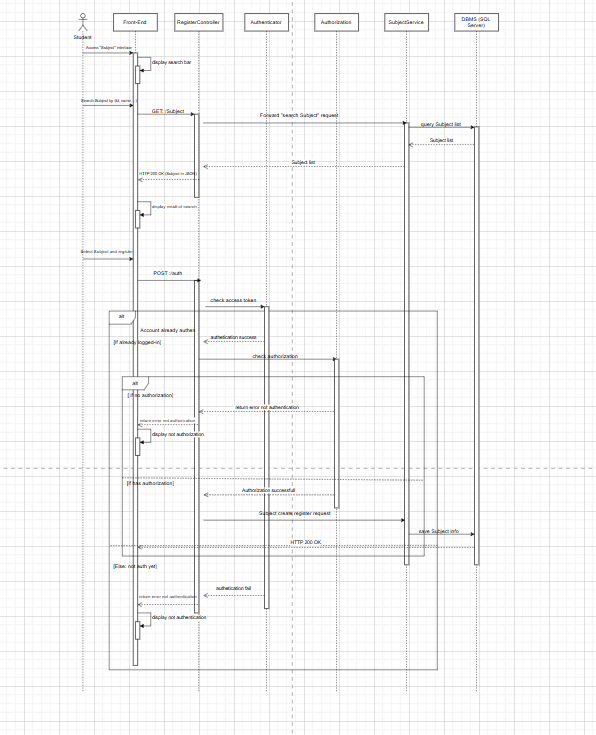
1.3 Logical ERD  
  


1.4 State ERD

  
  
Part 2  
2.1 Class Diagram of register subject



2.2 Sequence Diagram of register subject(sorry for this image so big)

Part 3  
3.1

-The design pattern is Factory method Pattern  
-Pattern family is creational  
  
3.2

### Factory Method

java

abstract class Dialog {

public void renderWindow() {

Button okButton = createButton();

okButton.render();

}

public abstract Button createButton();

}

class WindowsDialog extends Dialog {

public Button createButton() {

return new WindowsButton();

}

}

interface Button {

void render();

}

class WindowsButton implements Button {

public void render() {

System.out.println("Render Windows Button");

}

}

3.3  
-In the situation in software development, this pattern usually using in the system need to create many object but don’t change the core of object, example is the system use to do exam, quiz with many question but don’t change that question.