Student Class:

```
public class Student {
    private String studentNumber;
    private String name;
    char gender;
    private char grade;
    public Student(String studentNumber, String name, char gender, char
            grade) {
        this.studentNumber = studentNumber;
        this.name = name;
        this.gender = gender;
        this.grade = grade;
    }
    public String getStudentNumber() {
        return studentNumber;
    }
    public String getName() {
        return name;
    }
    public char getGender() {
        return gender;
    }
    public char getGrade() {
        return grade;
    }
}
```

StudentList Class:

```
public class StudentList {
    private int maxSize;
    private int position;
    private Student[] listEntry;
    StudentList(int n){
        maxSize=n;
        listEntry=new Student[maxSize];
        position=-1;
    }
    boolean isListEmpty(){
        return (position==-1);
    }
    boolean isListFull(){
        return (position==maxSize-1);
    }
    int listSize(){
        return (position+1);
    }
    void insertLast(Student data){
        if (isListFull()){System.out.println("List is full.");
        }
        else {
            listEntry[++position]=data;
        }
    }
```

```
    void insert(int p,Student data){
        if (isListFull()){
            System.out.println("List is full.");
        } else if (p<0 || p>listSize()) {
            System.out.println("Not in the range.");
        }
        else {
            for (int i=listSize();i>p;i--){
                listEntry[i]=listEntry[i-1];
            }
            listEntry[p]=data;
            position++;
        }
    }
    Student delete(int p){
        if (isListEmpty()){
            System.out.println("List is empty.");
        } else if (p>0 || p<listSize()) {
            System.out.println("Not in the range.");
        }
        else {
            Student element=listEntry[p];
            for (int i=p;i<listSize()-1;i++){
                listEntry[i]=listEntry[i+1];
            }
            position--;
            return element;
        }
        return null;
    }
    Student retrieveList(int p){
        if (isListEmpty()){
            System.out.println("List is empty.");
        } else if (p<0 || p>listSize()) {
            System.out.println("Not in the range.");
        }
        else {
            Student element=listEntry[p];
            return element;
        }
        return null;
    }
    void replace(int p,Student data){
        if (isListEmpty()){
            System.out.println("List is empty.");
        } else if (p<0 || p>listSize()) {
            System.out.println("Not in the range.");
        }
        else {
            listEntry[p]=data;
        }
    }
    void traverseList(){
        for (int i=0;i<listSize();i++){
            System.out.println("Student
Number:"+listEntry[i].getStudentNumber()+" | Student
Name:"+listEntry[i].getName()+" | Gender:"+listEntry[i].getGender()+" |
```

```
Grade:"+listEntry[i].getGrade());
        }
    }
    void sort(){
        for (int i = 0; i < listEntry.length; i++) {
            int j = i;
            while (j > 0) {
                if (listEntry[j].getGrade() < listEntry[j - 1].getGrade())
                { Student temp = listEntry[j];
                    listEntry[j] = listEntry[j - 1];
                    listEntry[j - 1] = temp;
                }
                j--; }
        }
    }
    public Student[] binarySearch(char target){
        return binarySearch(target,0,listSize()-1);
    }
    private Student[] binarySearch(char target,int min, int max){
        if (min>max){
            return null;
        }
        else {
            int mid=(min+max)/2;
            if (listEntry[mid].getGrade()==target){
                int start=mid;
                int end=mid;
                while (start>min && listEntry[start-1].getGrade()==target){
                    start--;
                }
                while (end<max && listEntry[end+1].getGrade()==target){
                    end++;
                }
                Student[] result=new Student[end-start+1];
                System.arraycopy(listEntry,start,result,0,end-start+1);
                return result;
            } else if (listEntry[mid].getGrade()<target) {
                return binarySearch(target,mid+1,max);
            }
            else {
                return binarySearch(target,min,mid-1);
            }
        }
    }
}
```

Main Class:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        StudentList stList=new StudentList(10);
        stList.insertLast(new Student("PS/2017/280","Kamal",'M','B'));
        stList.insertLast(new Student("PS/2017/149","Nirmal",'F','B'));
        stList.insertLast(new Student("PS/2017/045","Sarath",'M','C'));
        stList.insertLast(new Student("PS/2017/73","Kasuni",'F','A'));
        stList.insertLast(new Student("PS/2017/301","Chanaka",'M','C'));
        stList.insertLast(new Student("PS/2017/312","Akila",'F','A'));
        stList.insertLast(new Student("PS/2017/105","Dasuni",'F','A'));
        stList.insertLast(new Student("PS/2017/016","Amal",'M','A'));
        stList.insertLast(new Student("PS/2017/198","Binura",'M','B'));
        stList.insertLast(new Student("PS/2017/151","Sithara",'F','A'));
        stList.traverseList();
        System.out.println();
        stList.sort();
        System.out.println("After sorting:-");
        stList.traverseList();
        System.out.println();
        Scanner input=new Scanner(System.in);
        System.out.print("Enter Grade: ");
        char grade=input.next().charAt(0);
        Student[] result = stList.binarySearch(grade); if (result != null)
        {
            System.out.println("Students with Grade " + grade + ":"); for
        (Student student : result) {
            System.out.println(student.getStudentNumber() + " | " +
                    student.getName() + "\t| " + student.getGender() + "\t| "
+
                    student.getGrade());
        }
        } else {
            System.out.println("No students found" + grade);
        }
    }
}
```

Output:

```
Student Number:PS/2017/280 | Student Name:Kamal | Gender:M | Grade:B
Student Number:PS/2017/149 | Student Name:Nirmal | Gender:F | Grade:B
Student Number:PS/2017/045 | Student Name:Sarath | Gender:M | Grade:C
Student Number:PS/2017/73 | Student Name:Kasuni | Gender:F | Grade:A
Student Number:PS/2017/301 | Student Name:Chanaka | Gender:M | Grade:C
Student Number:PS/2017/312 | Student Name:Akila | Gender:F | Grade:A
Student Number:PS/2017/105 | Student Name:Dasuni | Gender:F | Grade:A
Student Number:PS/2017/016 | Student Name:Amal | Gender:M | Grade:A
Student Number:PS/2017/198 | Student Name:Binura | Gender:M | Grade:B
Student Number:PS/2017/151 | Student Name:Sithara | Gender:F | Grade:A

After sorting:-
Student Number:PS/2017/73 | Student Name:Kasuni | Gender:F | Grade:A
Student Number:PS/2017/312 | Student Name:Akila | Gender:F | Grade:A
Student Number:PS/2017/105 | Student Name:Dasuni | Gender:F | Grade:A
Student Number:PS/2017/016 | Student Name:Amal | Gender:M | Grade:A
Student Number:PS/2017/151 | Student Name:Sithara | Gender:F | Grade:A
Student Number:PS/2017/280 | Student Name:Kamal | Gender:M | Grade:B
Student Number:PS/2017/149 | Student Name:Nirmal | Gender:F | Grade:B
Student Number:PS/2017/198 | Student Name:Binura | Gender:M | Grade:B
Student Number:PS/2017/045 | Student Name:Sarath | Gender:M | Grade:C
Student Number:PS/2017/301 | Student Name:Chanaka | Gender:M | Grade:C

Enter Grade: C
Students with Grade C:
PS/2017/045 | Sarath    | M | C
PS/2017/301 | Chanaka   | M | C
```