
PGDIT Automation System

Software Design and Analysis

Team Outlaws

Institute of Information Technology

University of Dhaka

PGDIT Automation System

Software Design Report

Submitted to

Nadia Nahar

Lecturer

Institute of Information Technology

University of Dhaka

Submitted by

Team Name:

Outlaws

Team Members:

Md. Reshad Mollick – 0709

Md. Nafis Faysal - 0707

Md. Aquib Azmain – 0718

Abu Rafe Md. Jamil – 0722

Sabbir Hussain Meraj - 0726

BSSE Session: 2014-2015



**Institute of Information Technology
University of Dhaka**

16-11-2017

Abstract

This report is about the software design and analysis report of project "PDGIT automation system". It contains architectural design, component design and interface design of PGDIT automation system.

TABLE OF CONTENTS

Chapter 1: Architectural Design.....	10
1.1 Representing the system in context	10
1.2 Define archetypes	11
1.3 Refine the Architecture into Components.....	13
1.4 Describe Instantiations of the System	14
1.5 Structured Design.....	15
Chapter 2: Component Design.....	29
2.1 Design classes that correspond to the problem domain	29
2.2 Design classes that correspond to the infrastructure domain	29
2.3 Class elaboration	30
2.3 a) Class collaboration	36
2.3 b) Method call sequence for individual modules.....	41
2.3 c) Appropriate Interfaces.....	46
2.3 d) Elaborate Attributes.....	46
2.3 e) Describe Processing Flow.....	48
2.3 f) State Diagram.....	75
2.4 Persistent Data.....	80
2.5 Elaborated Deployment	80
Chapter 3: Interface Design	81
3.1 Interface Analysis.....	81
User Analysis	81
Task Analysis	84
3.2 Interface Design Steps	88
Define interface objects and actions	88

Define events that will cause the state of the user interface to change	112
Depict each interface state as it look to end user.....	115

Content of Figures

Figure 1: PGDIT Automation System in context	10
Figure 2: Archetypes - attributes.....	11
Figure 3: Archetypes - Methods.....	12
Figure 4: Components of PGDIT System	13
Figure 5: Instantiations of PDGIT system.....	14
Figure 6: Level-0 DFD	15
Figure 7: Level-1 DFD	16
Figure 8: Level-2(Authentication) DFD.....	16
Figure 9: DFD of Registration process.....	17
Figure 10: Mapping of registration process.....	17
Figure 11: DFD of password recovery	18
Figure 12: Mapping of password recovery	18
Figure 13: DFD of Sign in.....	19
Figure 14: Transaction mapping of sign in.....	19
Figure 15: DFD of Admission administration	20
Figure 16: Mapping of admission administration	20
Figure 17: DFD of application payment	21
Figure 18: Mapping of application payment	21
Figure 19: DFD of scrutinizing process	22
Figure 20: Mapping of scrutinizing process	22
Figure 21: DFD of seat plan generation process	23
Figure 22: Mapping of seat plan generator	23
Figure 23: Written mark process.....	24
Figure 24: Mapping of written exam mark process	24
Figure 25: Mapping of viva schedule generator	25
Figure 26: DFD of admission payment process.....	26
Figure 27: Mapping of admission payment	26
Figure 28: DFD of course result generation process.....	27
Figure 29: Mapping of result generation process	27
Figure 30: DFD of result publication process	28

Figure 31: Mapping of course result publisher	28
Figure 32: Class elaboration of User	30
Figure 33: Class elaboration of Student.....	30
Figure 34: Class elaboration of Teacher.....	31
Figure 35: Class elaboration of Receptionist	32
Figure 36: Class elaboration of Scrutinizer.....	32
Figure 37: Program Chairperson.....	33
Figure 38: Class elaboration of Exam Controller	34
Figure 39: Class elaboration of Applicant.....	35
Figure 40: Class elaboration of Database operator.....	35
Figure 41: Class collaboration of authentication.....	36
Figure 42: Class collaboration of update Profile	36
Figure 43: Class collaboration of new admission session	36
Figure 44: Class collaboration of Apply.....	37
Figure 45: Class collaboration of view payment board.....	37
Figure 46: Class collaboration of update payment status	37
Figure 47: Class collaboration of update status of eligibility	38
Figure 48: Class collaboration of add viva schedule	38
Figure 49: Class collaboration of add new room.....	39
Figure 50: Class collaboration of result submission.....	39
Figure 51: Class collaboration of view result.....	39
Figure 52: Class collaboration of update result.....	40
Figure 53: Class collaboration of view supplementary list	40
Figure 54: Class collaboration of view course result	40
Figure 55: Method call sequence of Authentication	41
Figure 56: Method call sequence of Teacher registration.....	41
Figure 57: Method call sequence of Student account creation.....	42
Figure 58: Method call sequence of Candidate management	42
Figure 59: Method call sequence of Course management	43
Figure 60: Method call sequence of Result management.....	43
Figure 61: Method call sequence of User management.....	44
Figure 62: Method call sequence of Admission session	44

Figure 63: Method call sequence of Exam schedule manage	44
Figure 64: Method call sequence of Result manage	45
Figure 65: Method call sequence of Application processing	45
Figure 66: Processing flow of chooseOption.....	48
Figure 67: Processing flow of viewListOfCourse()	49
Figure 68: Processing flow of viewResultSheet(courseID,year)	49
Figure 69: Processing flow of viewSupplementaryList(courseID,year).....	50
Figure 70: Processing flow of viewCourseInformation(courseID,year)	50
Figure 71: Processing flow of updateStudentMark(courseID, year, mark, studentID)	51
Figure 72: Processing flow of updateCourseInformation(courseID, year, outline, description).....	51
Figure 73: Processing flow of updatePDF(courseID,year,username)	52
Figure 74: Processing flow of notifyForSupplementaryExam(courseID,year)	52
Figure 75: Processing flow of Connect().....	53
Figure 76: Processing flow of executeSelectCommand(sql).....	53
Figure 77: Processing flow of executeUpdateCommand(sql)	54
Figure 78: Processing flow of executeInsertCommand(sql)	54
Figure 79: Processing flow of viewPaymentBoard().....	55
Figure 80: Processing flow of getPaymentType ()	55
Figure 81: Processing flow of updatePaymentStatus()	56
Figure 82: Processing flow of viewListOfCourses()	56
Figure 83: Processing flow of getCourseID ()	57
Figure 84: Processing flow of viewResult()	57
Figure 85: Processing flow of changeInstructor()	58
Figure 86: Processing flow of lockResult()	58
Figure 87: Processing flow of viewListOfUsers()	59
Figure 88: Processing flow of addStudents().....	59
Figure 89: Processing flow of viewUserInfo ()	60
Figure 90: Processing flow of getUserInfo()	60
Figure 91: Processing flow of updateInfo()	61
Figure 92: Processing flow of downloaadAdmitCard (applicationID,dateOfBarth)	61

Figure 93: Processing flow of apply(name, fathersName, mothersName, email, contactNo, photo, dateOfBirth, nationality, permanentAddress, sscO, hscA, bachelor, hscMathMark, bachelorMathMark).....	62
Figure 94: Processing flow of veiwHeader()	62
Figure 95: Processing flow of veiwDashboard()	63
Figure 96: Processing flow of viewNotification (notificationID)	63
Figure 97: Processing flow of updatePersonalDate(username, email, mobile, address, password, picture).....	64
Figure 98: Processing flow of verifyPassword(username, password)	64
Figure 99: Processing flow of viewListOfRoom()	65
Figure 100: Processing flow of updateRoomCapacity().....	65
Figure 101: Processing flow of insertRoomDatabase()	66
Figure 102: Processing flow of viewVivaSchedule().....	66
Figure 103: Processing flow of insertScheduleIntoDatabase()	67
Figure 104: Processing flow of insertAdmisionInfointoDatabase()	67
Figure 105: Processing flow of viewListOfEligibleCandidatesOnScrutinize().....	68
Figure 106: Processing flow of viewListOfEligibleCandidatesAfterScrutinize()	68
Figure 107: Processing flow of viewListOfEligibleCandidatesAfterWrittenExam()	69
Figure 108: Processing flow of updateAdmissionInfo	69
Figure 109: Processing flow of viewCourseInformation(courseID,year)	70
Figure 110: Processing flow of viewResultsheet()	70
Figure 111: Processing flow of viewRunningCourses().....	71
Figure 112: Processing flow of getapplicationId()	71
Figure 113: Processing flow of getIndex().....	72
Figure 114: Processing flow of UpdateStatus(applicationId,index)	72
Figure 115: Processing flow of searchByApplicationID(applicationId)	73
Figure 116: Processing flow of searchByIndex(index)	73
Figure 117: Processing flow of searchByIndex	74
Figure 118: State diagram of User class.....	75
Figure 119: State diagram of Student class	75
Figure 120: State diagram of Teacher class.....	76
Figure 121: State diagram of Receptionist class	76

Figure 122: State diagram of Exam Controller class	77
Figure 123: State diagram of Scrutinizer class.....	77
Figure 124: State diagram of Program Chairman class	78
Figure 125: State diagram of Applicant class.....	79
Figure 126: State diagram of Database Operator	79
Figure 127: Elaborated deployment of PGDIT Automation System	80
Figure 128: External (Home)	88
Figure 129: External (Admission – Session ended)	89
Figure 130: External (Admission - Session going admission form)	89
Figure 131: External (Admission - Session going - Admission form - Submit)	90
Figure 132: External (Admission – Session going – Admit card - Available)	90
Figure 133: External (Admission – Session going – Admit card – Not Available)	91
Figure 134: External (Create Account)	91
Figure 135: External (Sing in)	92
Figure 136: External (Sing in-Forgot password)	92
Figure 137: External (Sing in - Blocked)	93
Figure 138: Internal (Teacher: Home)	93
Figure 139: Internal (Teacher: Manage Course- Course Details).....	94
Figure 140: Internal (Teacher: Manage Course- Update Marks)	94
Figure 141: Internal (Teacher: Manage Course- List of Students (Suppli))	95
Figure 142: Internal (Teacher: Viva Schedule).....	95
Figure 143: Internal (Member: Profile)	96
Figure 144: Internal (Student: Home)	96
Figure 145: Internal (Student: View Course).....	97
Figure 146: Internal (Student: View Course	97
Figure 147: Internal (Student: Profile.....	98
Figure 148: Internal (Receptionist: Home)	98
Figure 149: Internal (Receptionist: Search Applicant-By name)	99
Figure 150: Internal (Receptionist: Search Applicant-By mobile no).....	99
Figure 151: Internal (Receptionist: Payment-Application payment).....	100
Figure 152: Internal (Receptionist: Payment-Admission fees).....	101
Figure 153: Internal (Receptionist: Profile)	102

Figure 154: Internal (Scrutinizer: Home)	102
Figure 155: Internal (Scrutinizer: Search Applicant)	103
Figure 156: Internal (Scrutinizer: Profile)	103
Figure 157: Internal (Program chairperson: Home)	104
Figure 158: Internal (Program chairperson: Manage Course- Details)	104
Figure 159: Internal (Program chairperson: Manage Course- Add)	105
Figure 160: Internal (Program chairperson: Manage User- View)	105
Figure 161: Internal (Program chairperson: Manage User- View - Edit)	106
Figure 162: Internal (Program chairperson: Manage User- New User Request).....	107
Figure 163: Internal (Program chairperson: Manage User- Create Student Account)...	107
Figure 164: Internal (Program Chairman: Profile)	108
Figure 165: Internal (Exam controller: Home).....	108
Figure 166: Internal (Exam controller: Admission-Session)	109
Figure 167: Internal (Exam controller: Admission-Eligible Applicants)	109
Figure 168: Internal (Exam controller: Manage Schedule-Viva)	110
Figure 169: Internal (Exam controller: Manage Schedule-Update room)	110
Figure 170: Internal (Exam controller: Profile).....	111
Figure 171: Home	115
Figure 172: Notices	115
Figure 173: Options	116
Figure 174: Admission form	116
Figure 175: Dashboard of Teacher.....	117
Figure 176: Profile	117
Figure 177: Course details.....	118
Figure 178: Update marks.....	118
Figure 179: Supplementary exam list	119
Figure 180: Viva schedule.....	119
Figure 181: Dashboard of exam controller	120
Figure 182: Admission session	120
Figure 183: Viva schedule manage	121
Figure 184: Update room.....	121

CHAPTER 1: ARCHITECTURAL DESIGN

Software architectural design represents the structure of the data and program components that are required to build a computer-based system. There are 4 steps in architectural design:

- 1) Represent the system in context
- 2) Define archetypes
- 3) Refine the architecture into components
- 4) Describe instantiations of the system

1.1 Representing the system in context

Following is the ACD (Architectural Context Diagram) of "PGDIT automation system".

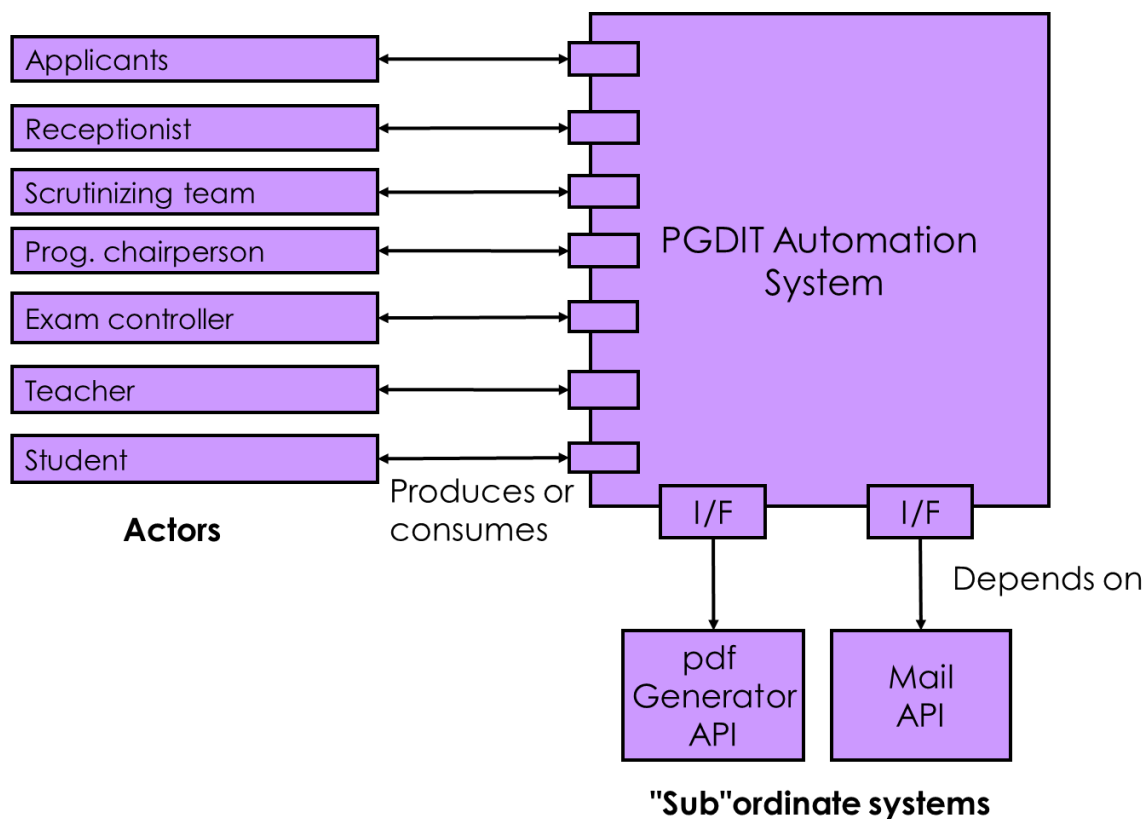


Figure 1: PGDIT Automation System in context

1.2 Define archetypes

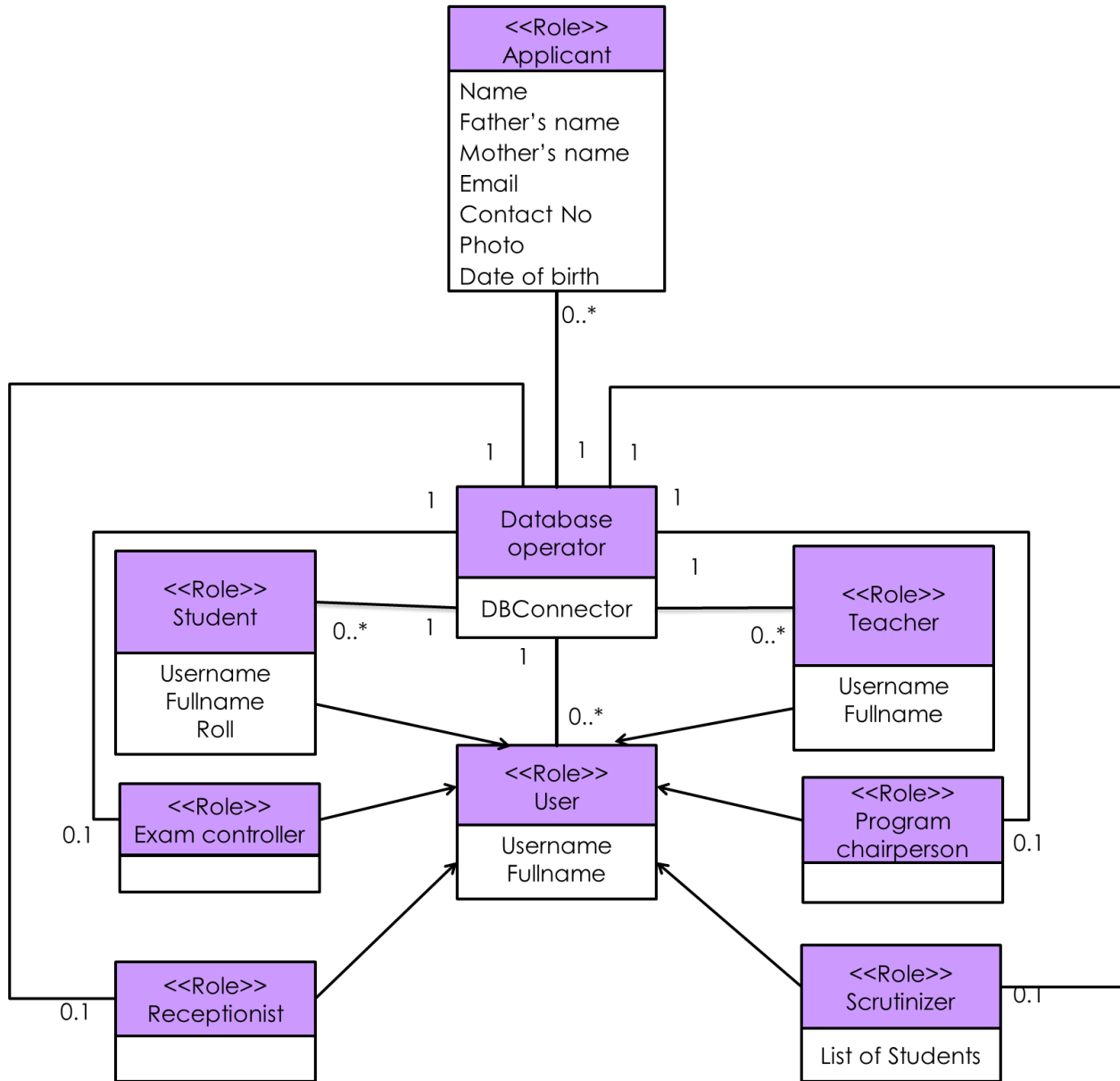


Figure 2: Archetypes - attributes

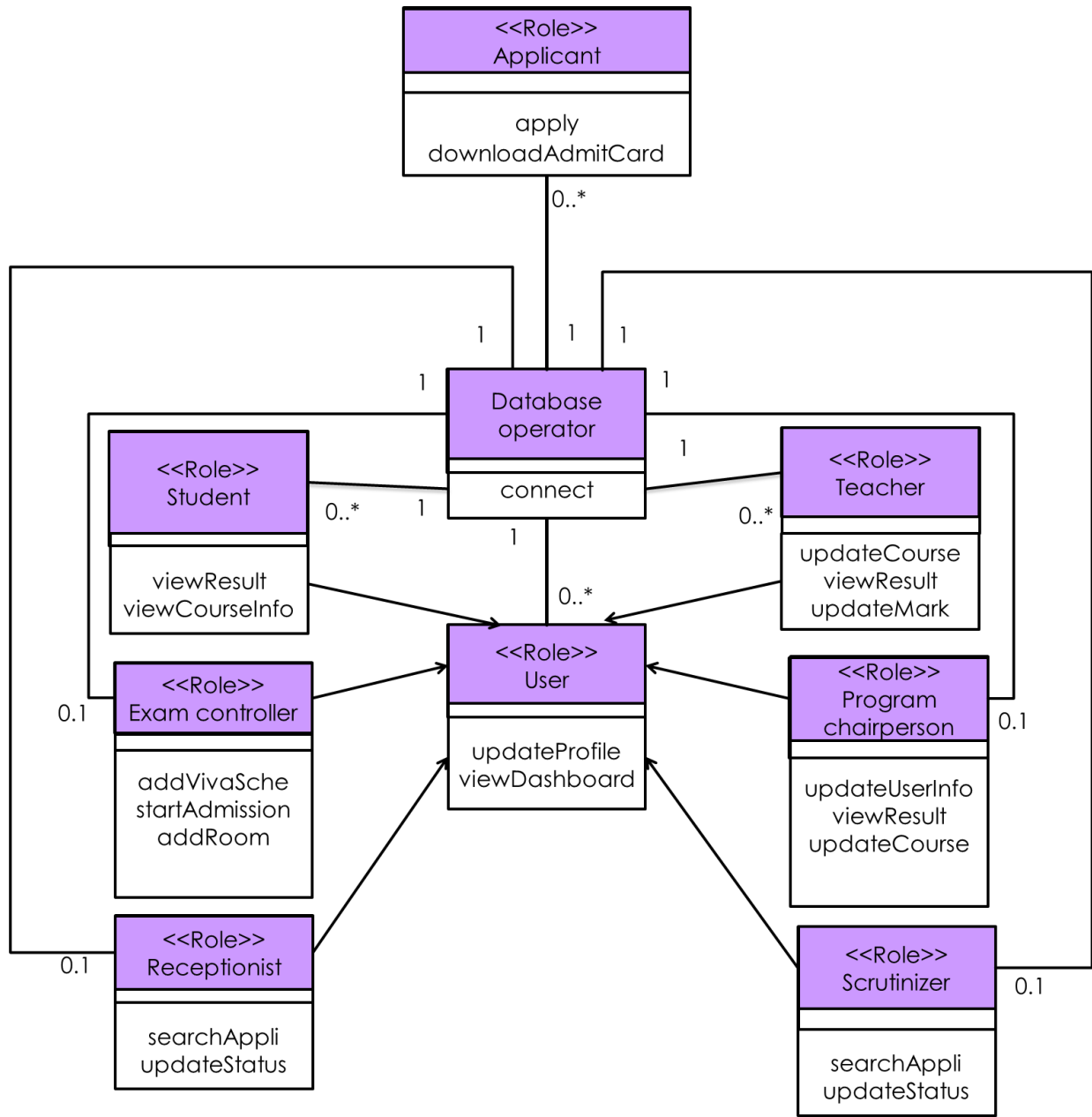


Figure 3: Archetypes - Methods

1.3 Refine the Architecture into Components

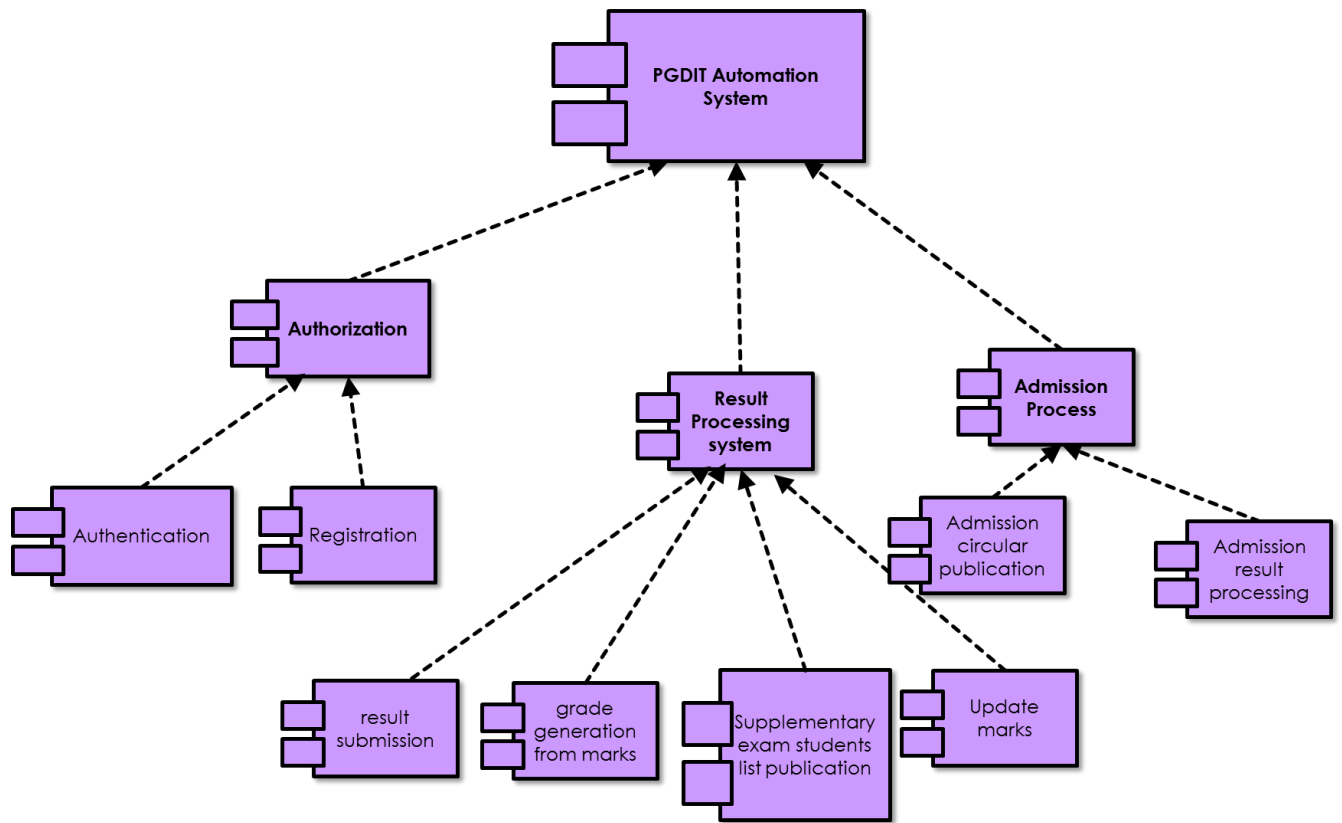


Figure 4: Components of PGDIT System

1.4 Describe Instantiations of the System

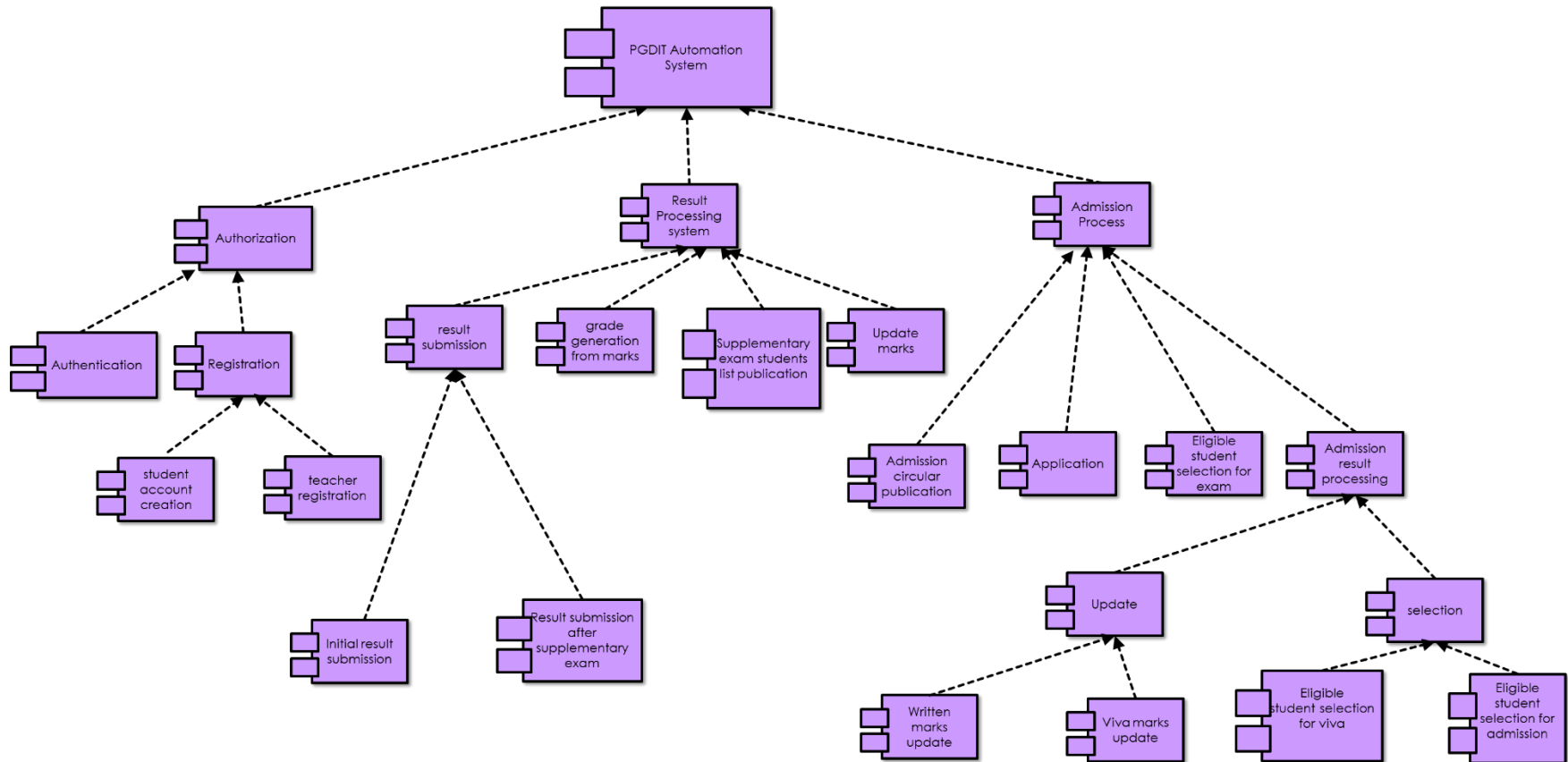


Figure 5: Instantiations of PDGIT system

1.5 Structured Design

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

Level-0 DFD

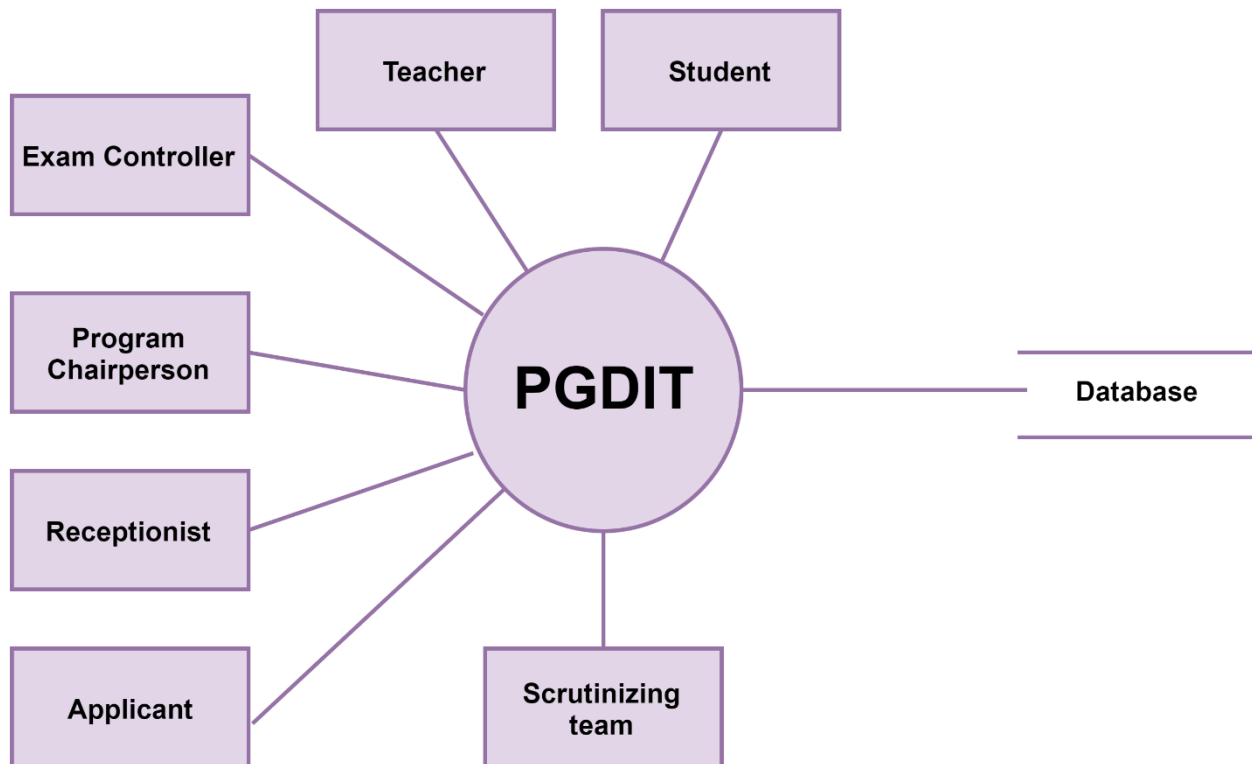


Figure 6: Level-0 DFD

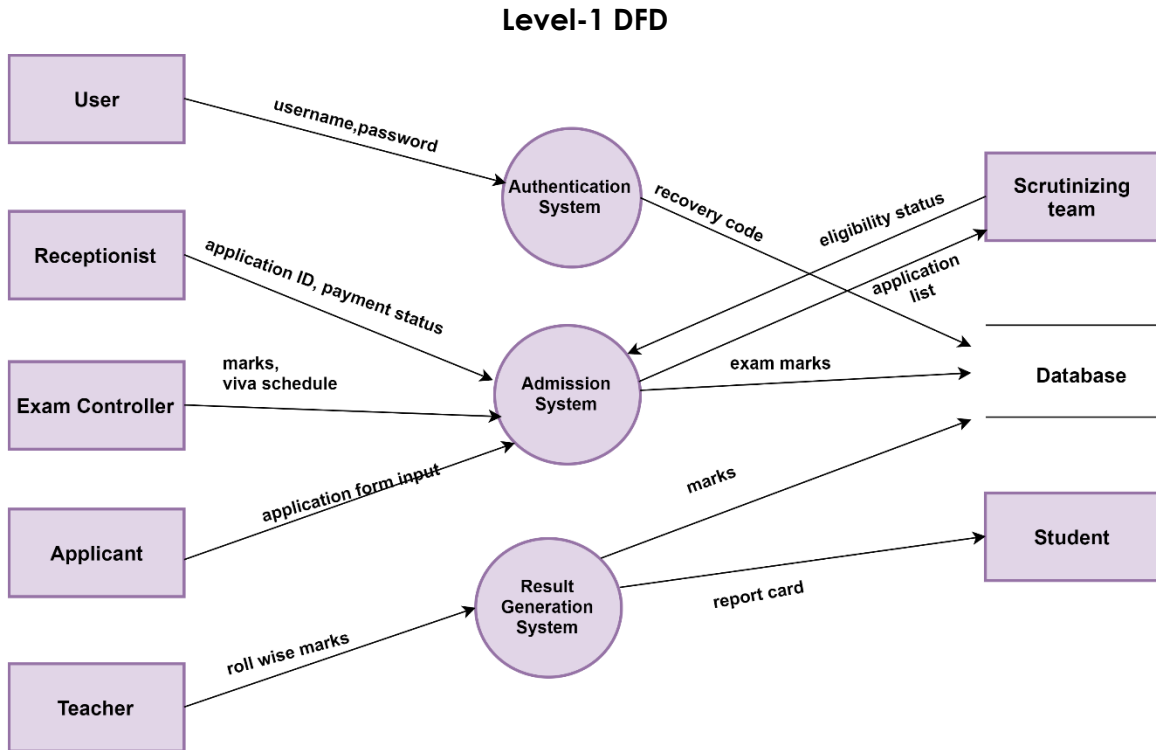


Figure 7: Level-1 DFD

Level-2 DFD (Authentication)

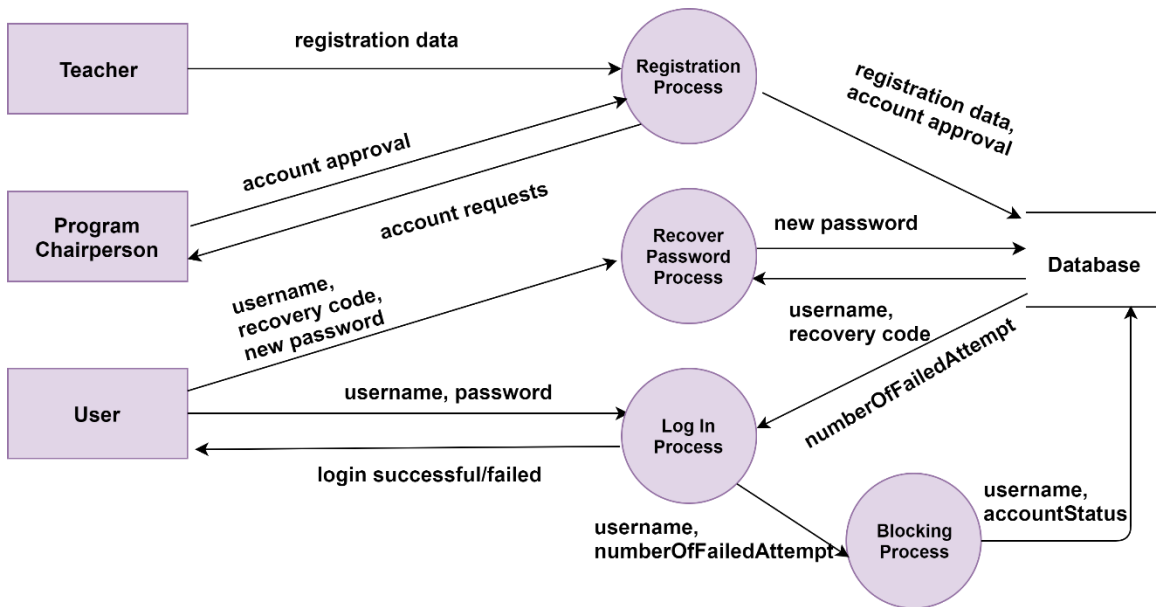


Figure 8: Level-2(Authentication) DFD

Level-2.1 Registration Process

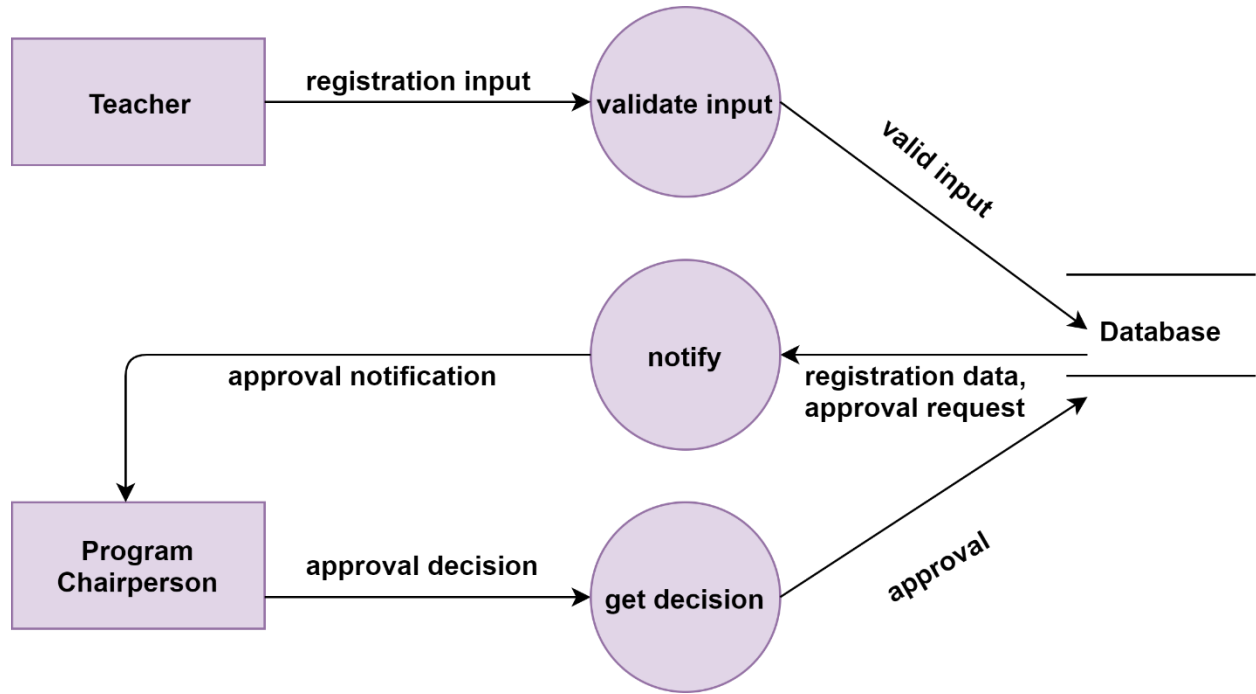


Figure 9: DFD of Registration process

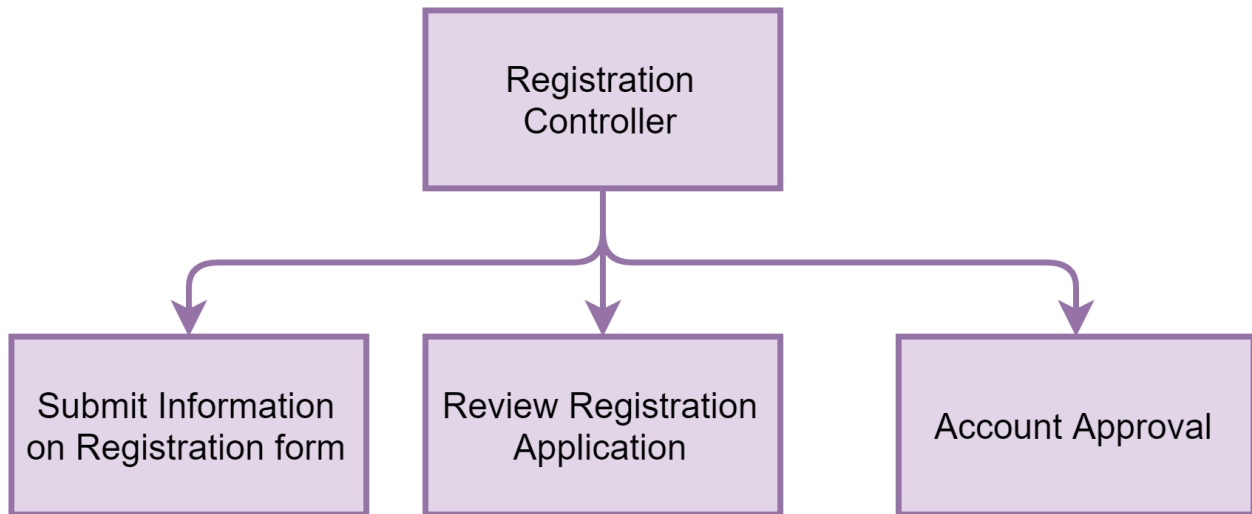


Figure 10: Mapping of registration process

Level 2.2 Password Recovery

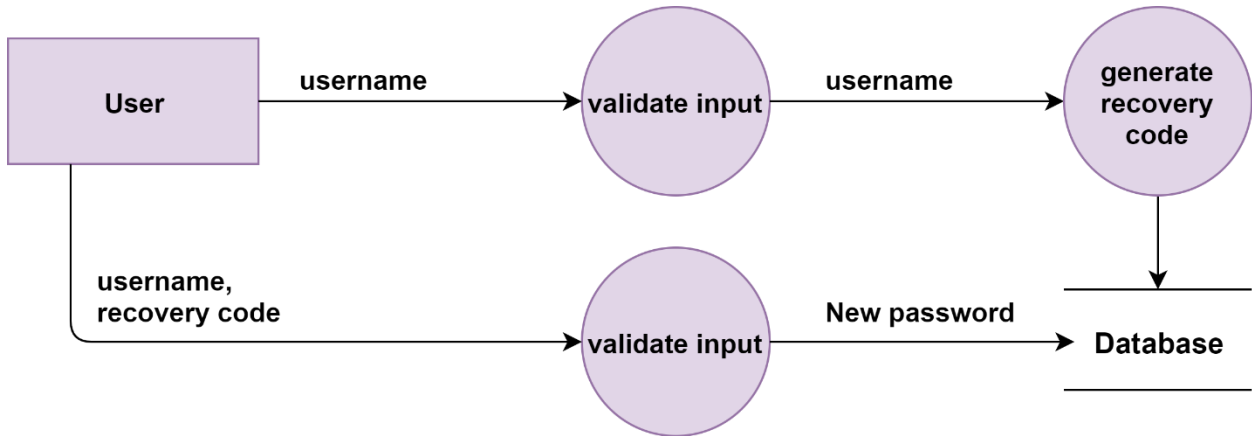


Figure 11: DFD of password recovery

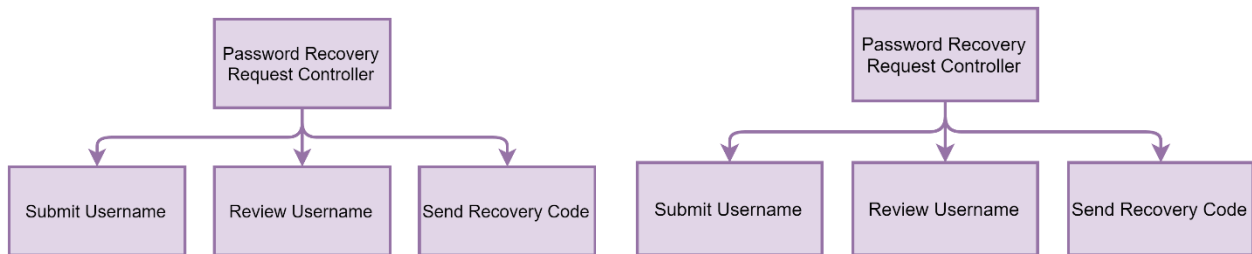


Figure 12: Mapping of password recovery

Level-2.3 Sign in Process

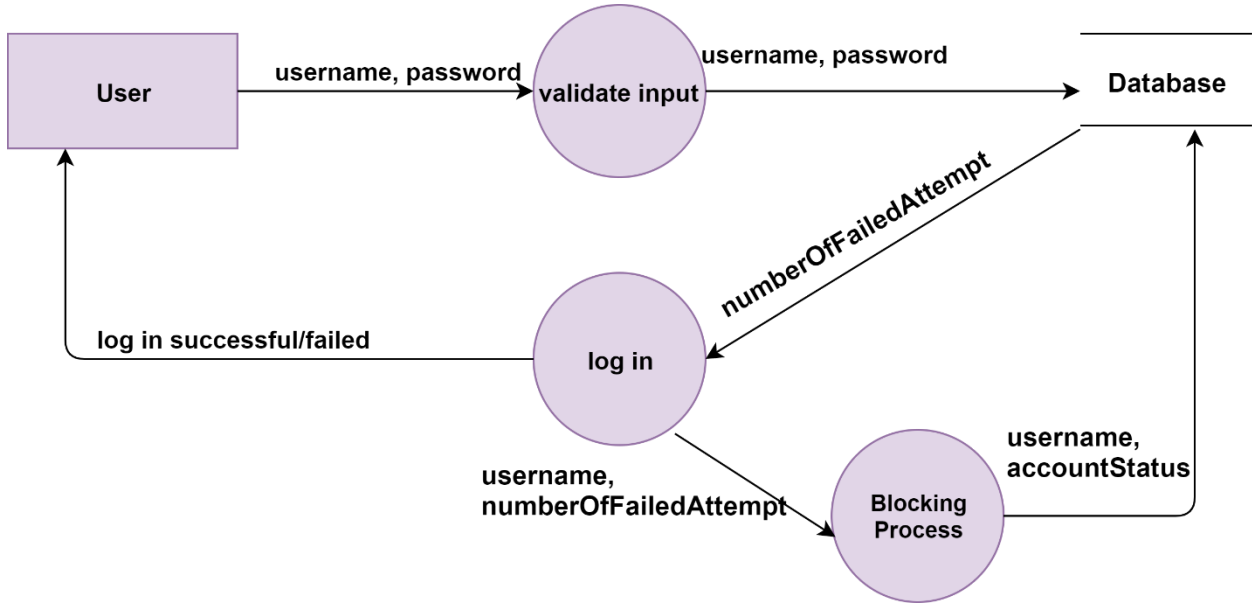


Figure 13: DFD of Sign in

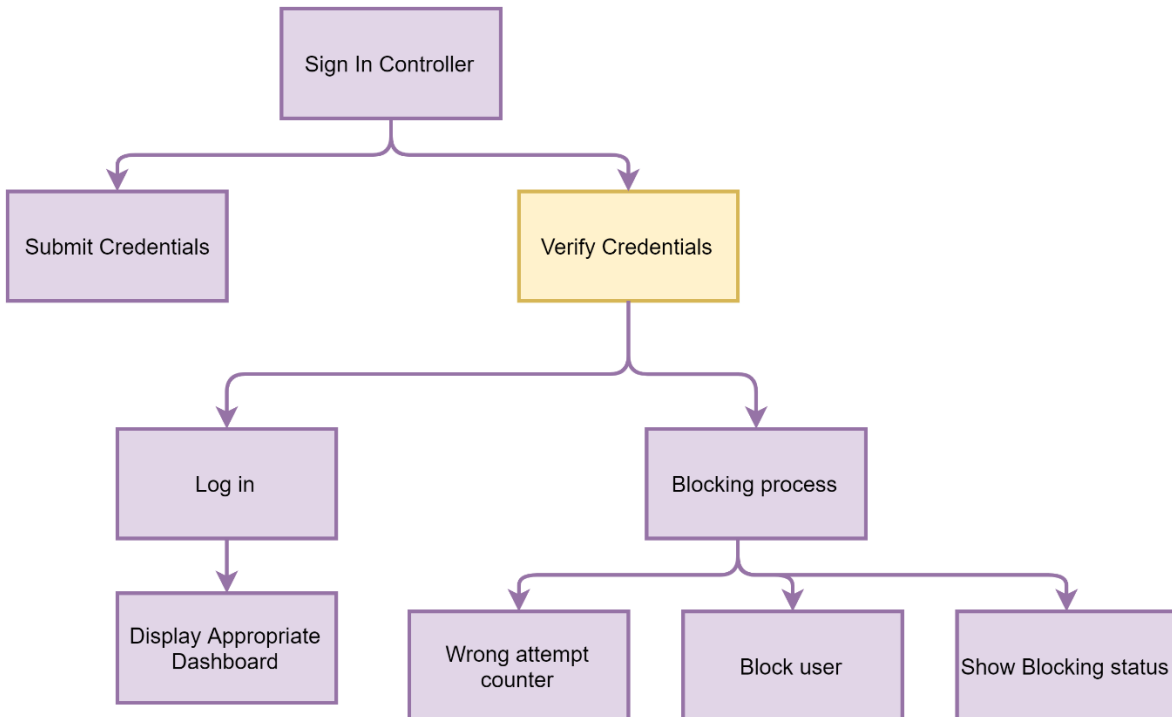


Figure 14: Transaction mapping of sign in

Level-2.4 Admission administration

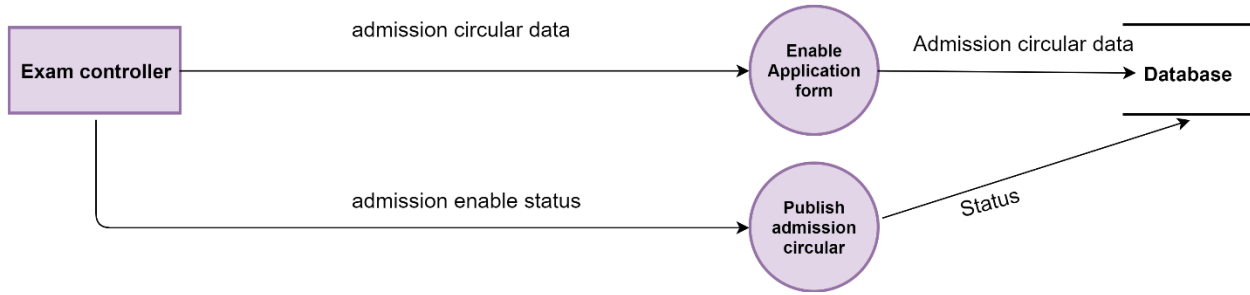


Figure 15: DFD of Admission administration

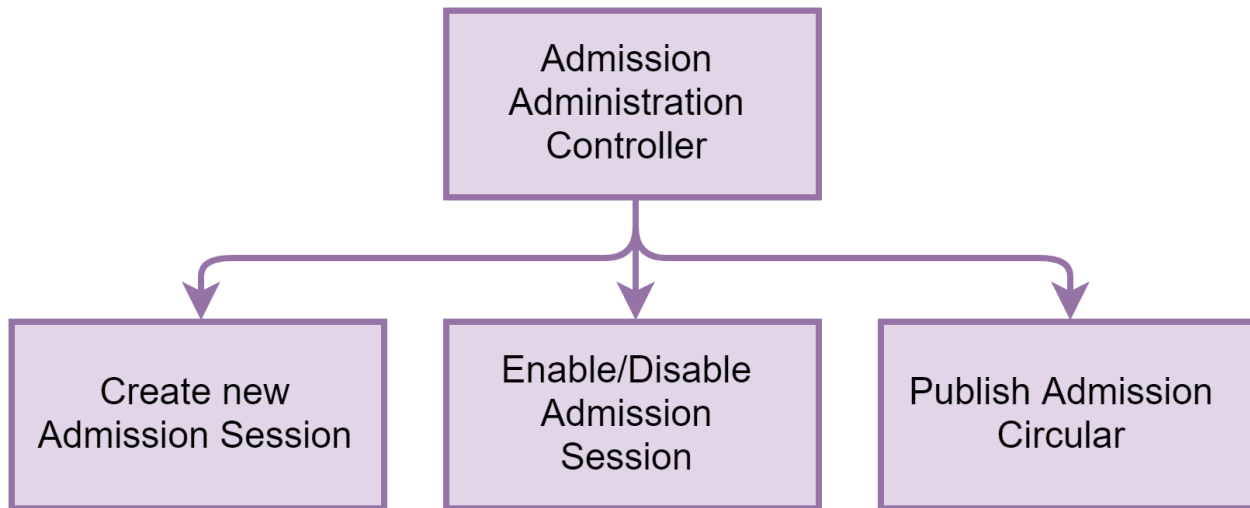


Figure 16: Mapping of admission administration

Level-2.5 Application payment

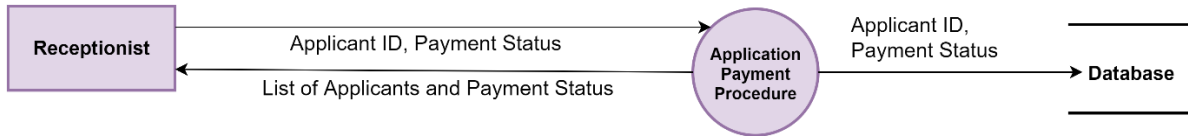


Figure 17: DFD of application payment

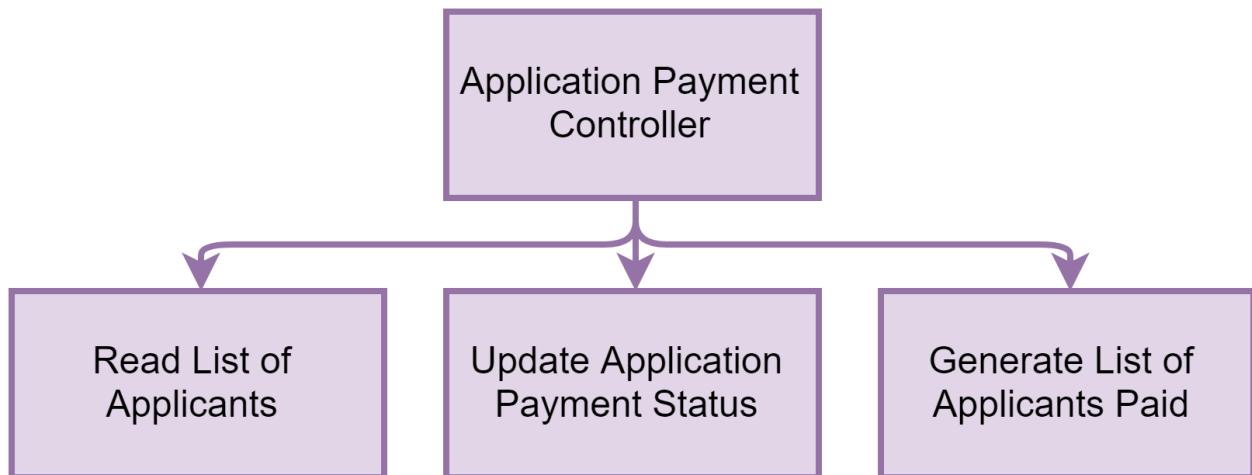


Figure 18: Mapping of application payment

Level-2.6 Scrutinizing process

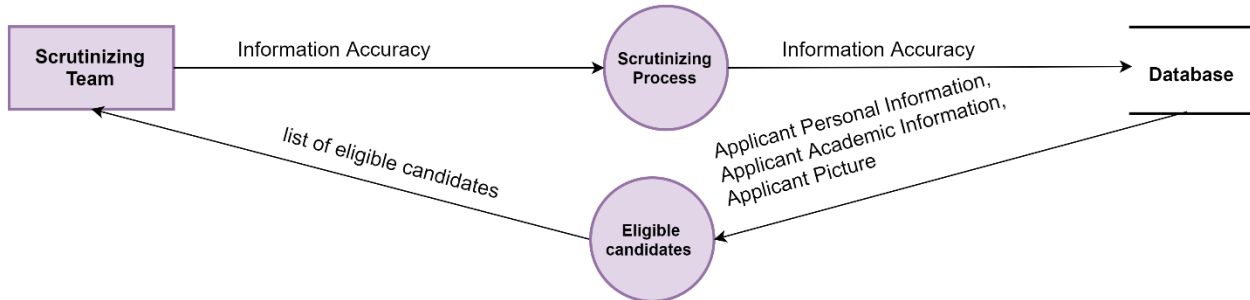


Figure 19: DFD of scrutinizing process

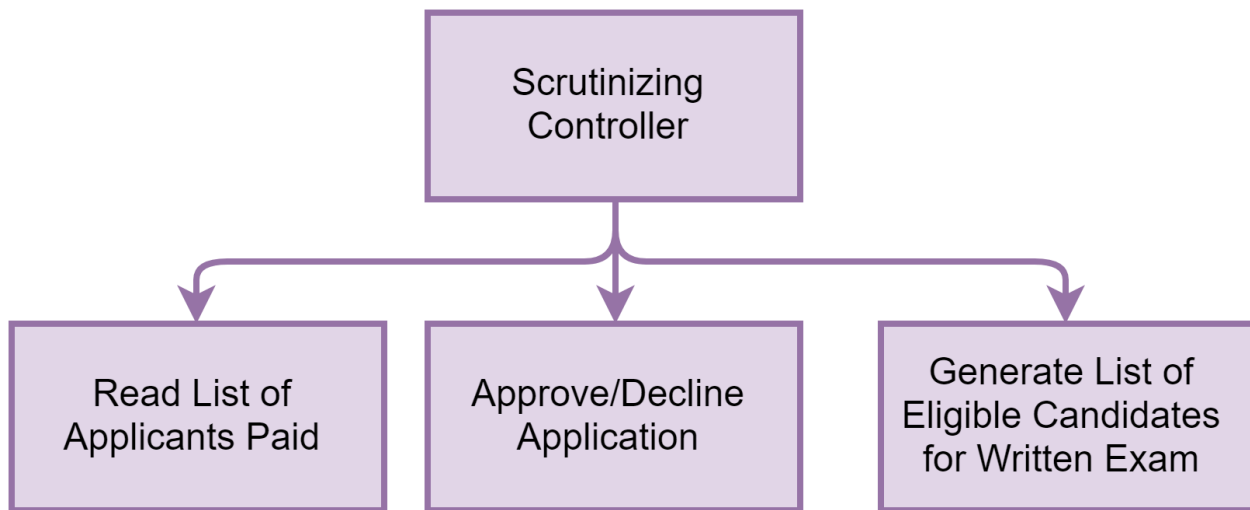


Figure 20: Mapping of scrutinizing process

Level-2.6 Seat plan generation process

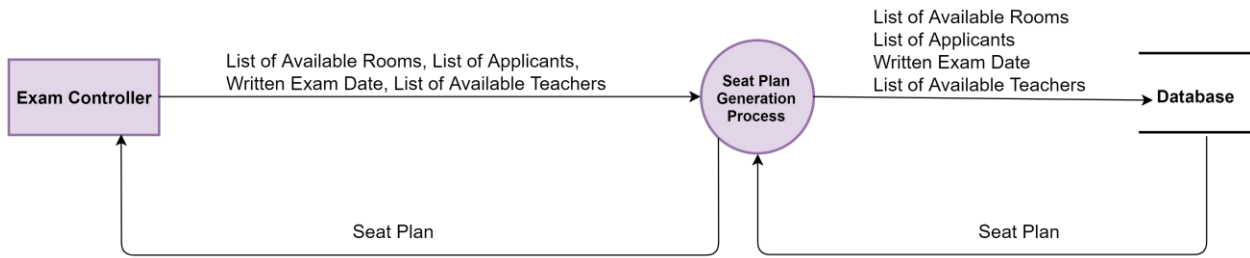


Figure 21: DFD of seat plan generation process

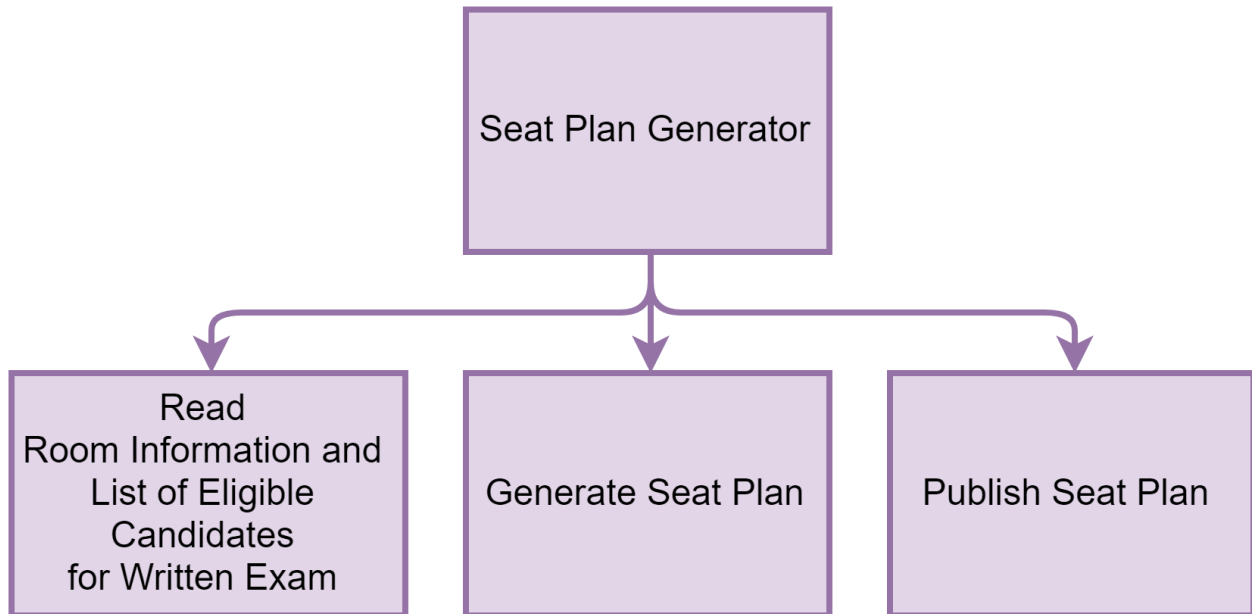


Figure 22: Mapping of seat plan generator

Level-2.7 Written mark process

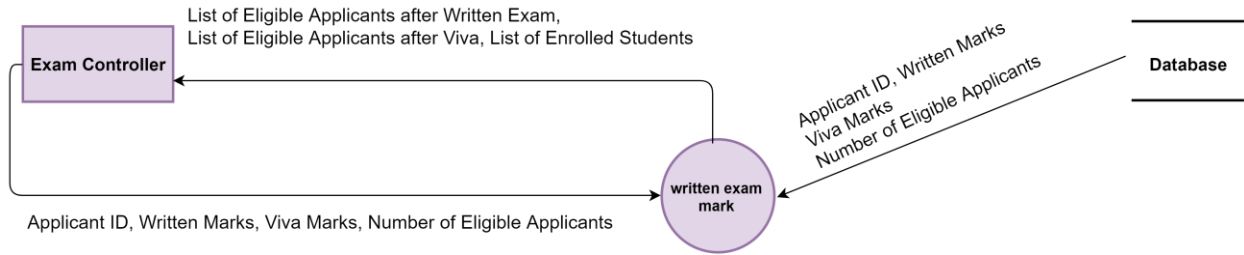


Figure 23: Written mark process

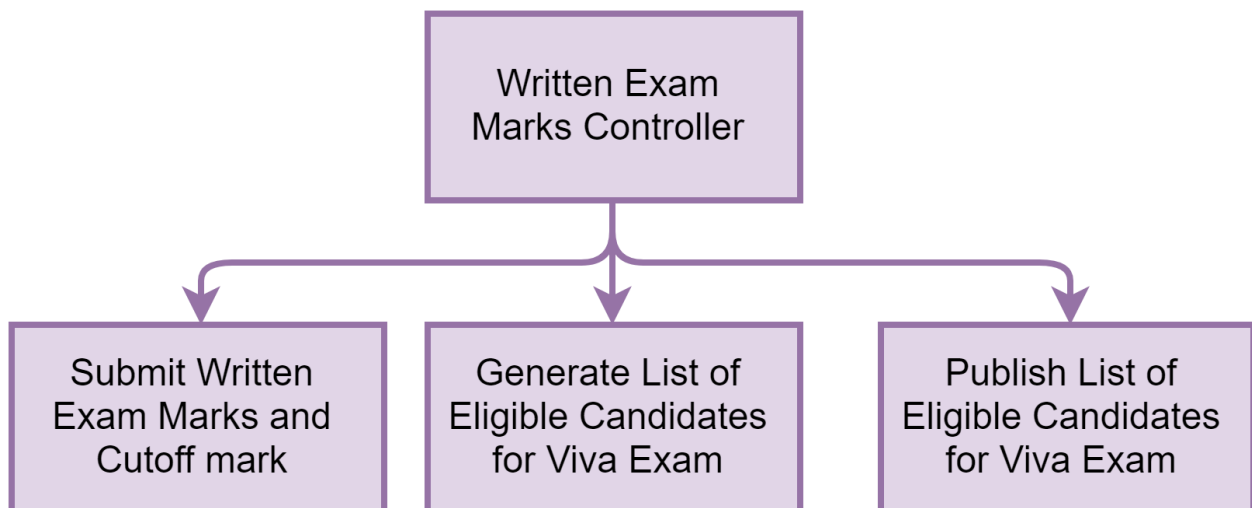


Figure 24: Mapping of written exam mark process

Level-2.7 Viva schedule generation

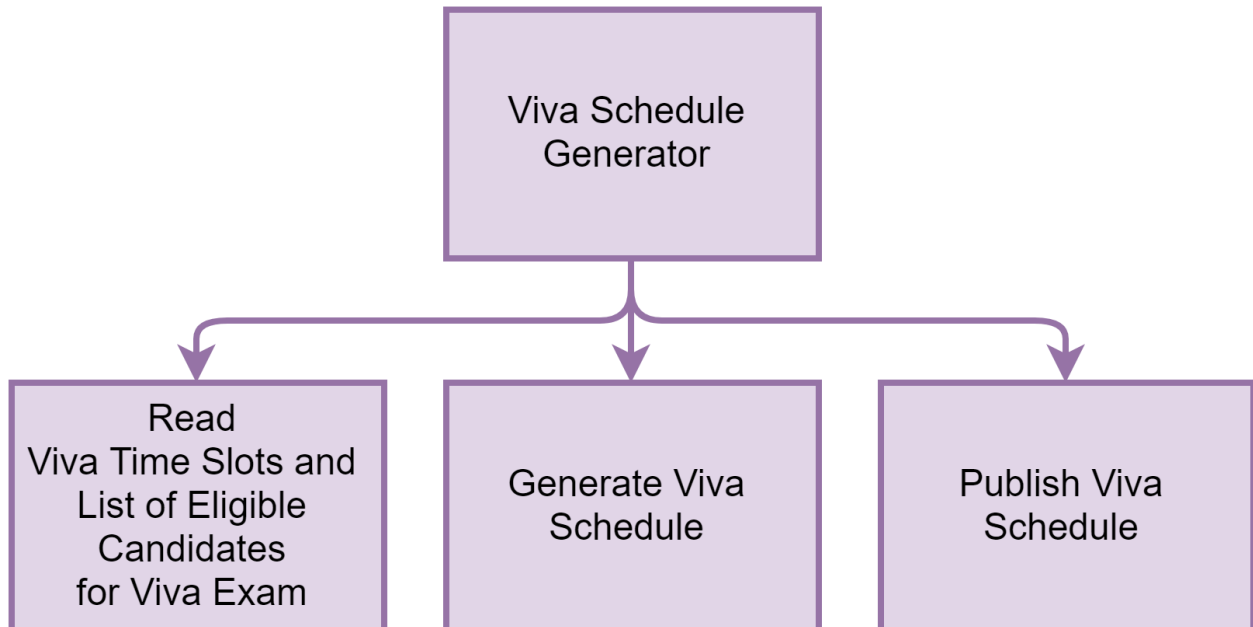
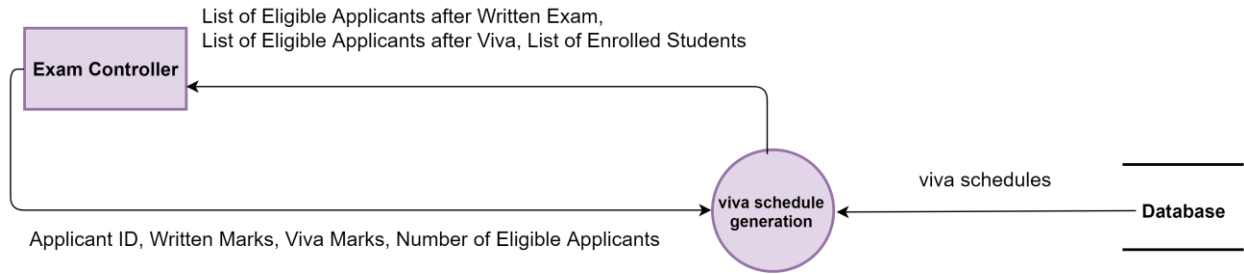


Figure 25: Mapping of viva schedule generator

Level-2.8 Admission payment process

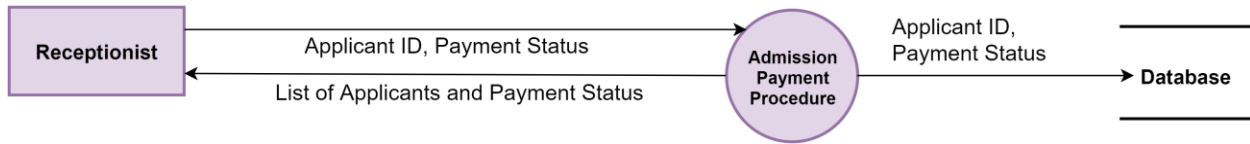


Figure 26: DFD of admission payment process

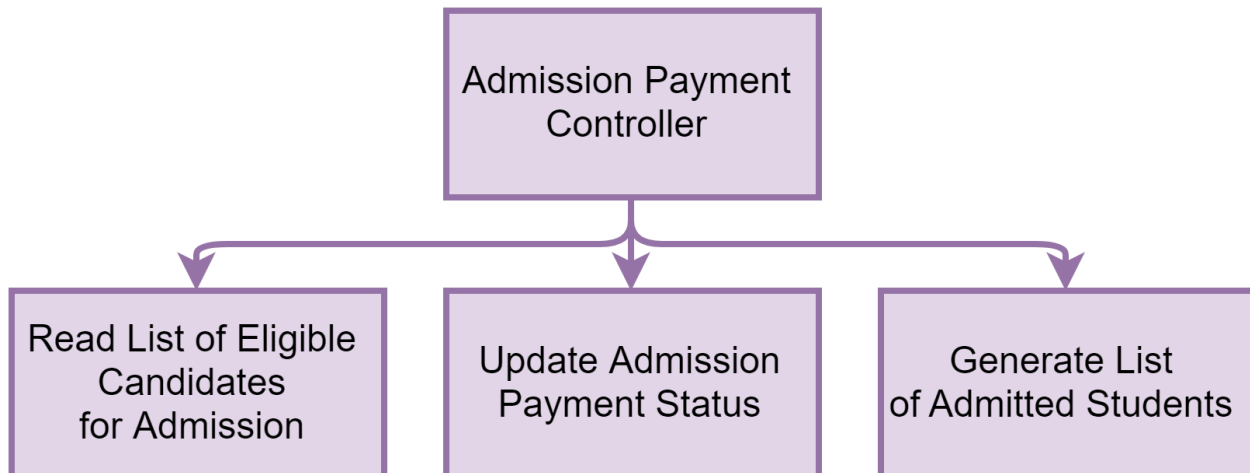


Figure 27: Mapping of admission payment

Level-2.9 Course result generation process

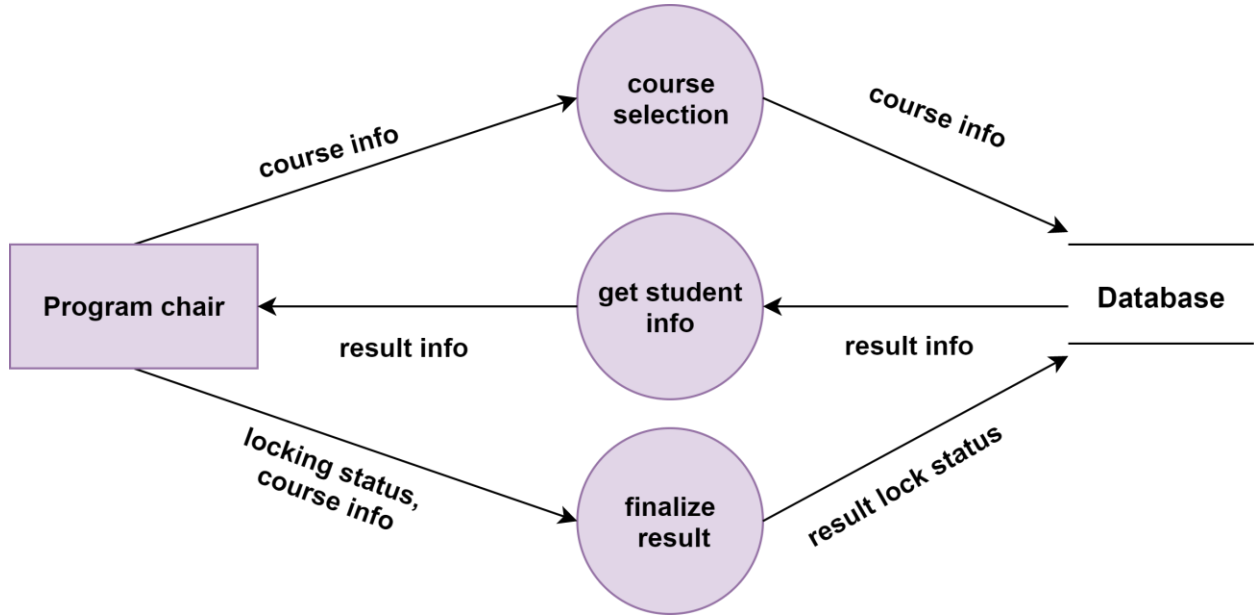


Figure 28: DFD of course result generation process

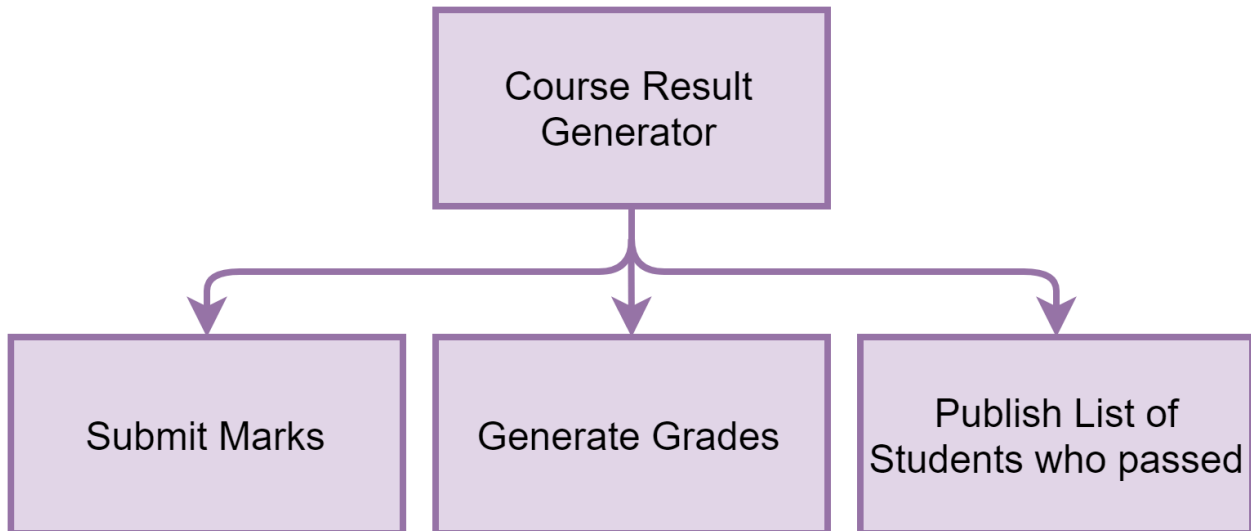


Figure 29: Mapping of result generation process

Level-2.10 Result publication process

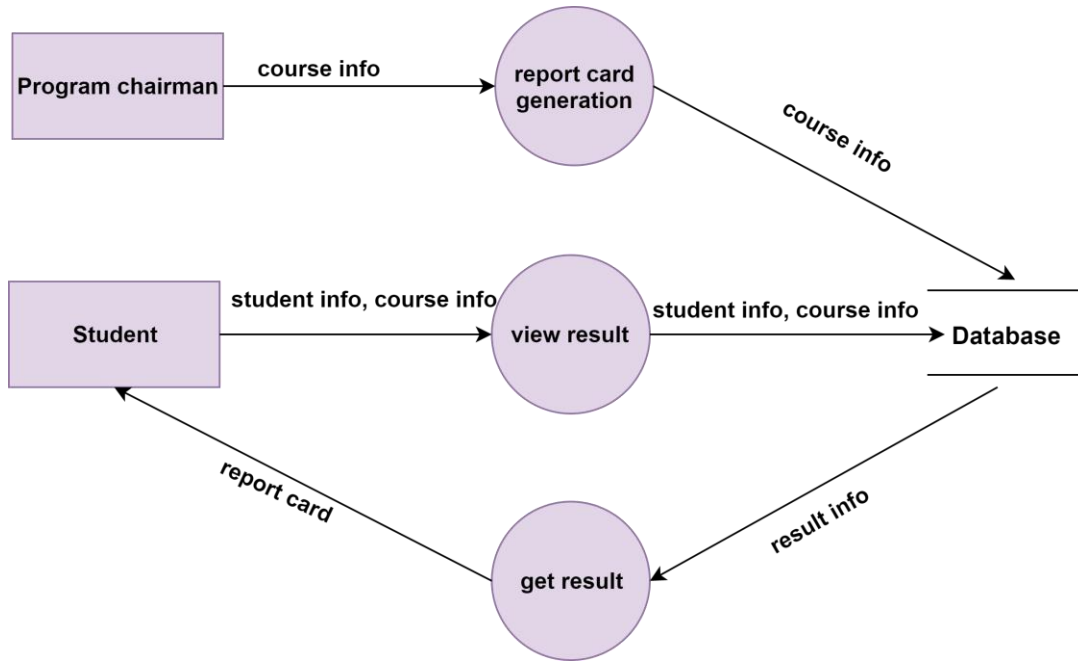


Figure 30: DFD of result publication process

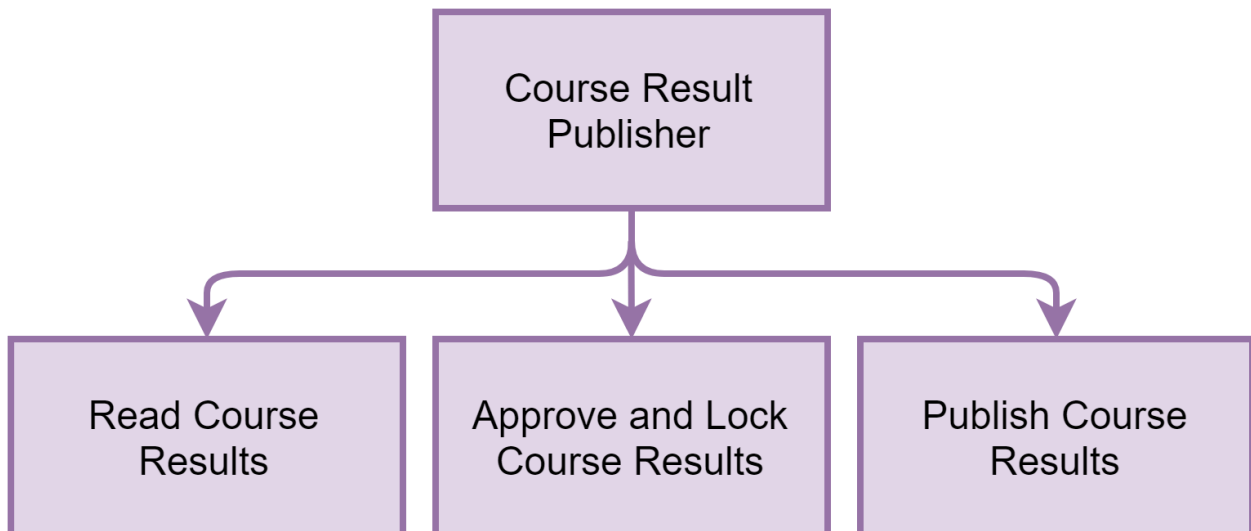


Figure 31: Mapping of course result publisher

CHAPTER 2: COMPONENT DESIGN

A component-level design can be represented using some intermediate representation (e.g. graphical, tabular, or text-based) that can be translated into source code.

2.1 Design classes that correspond to the problem domain



2.2 Design classes that correspond to the infrastructure domain

Database Operator

2.3 Class elaboration

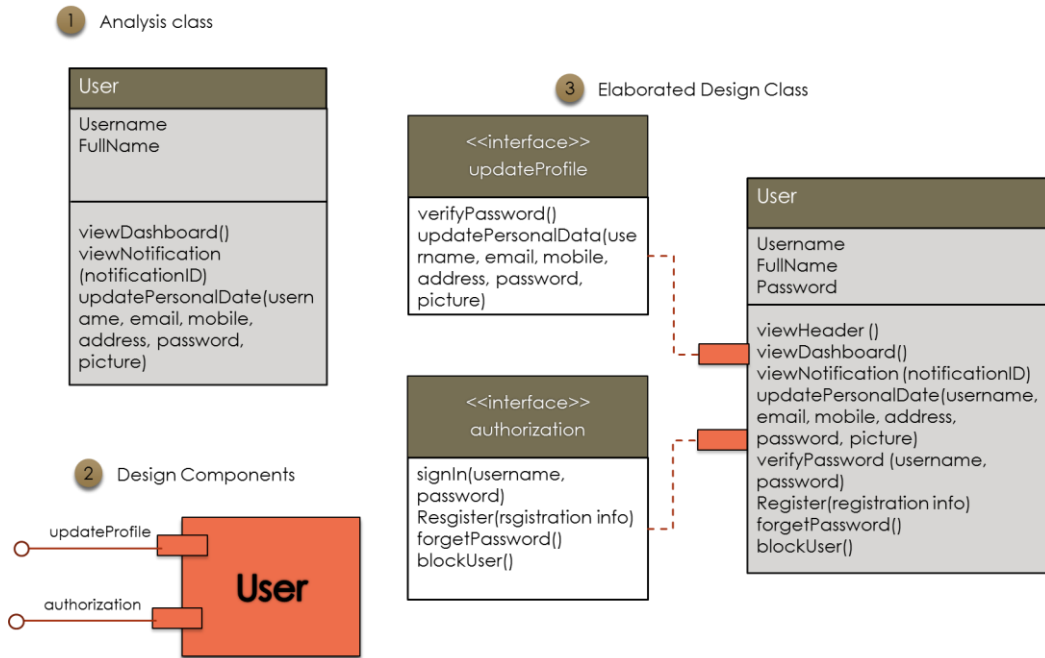


Figure 32: Class elaboration of User

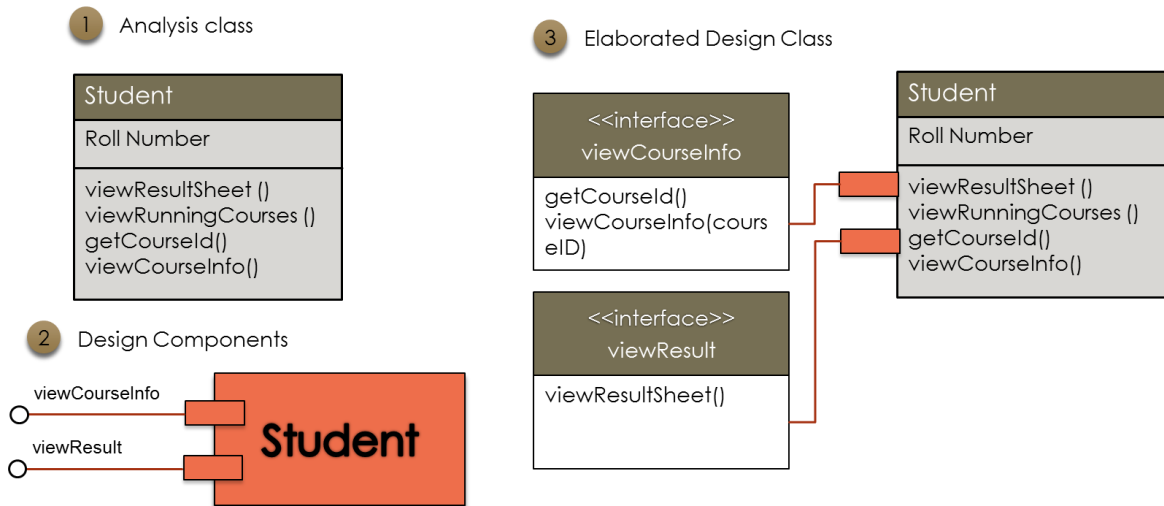
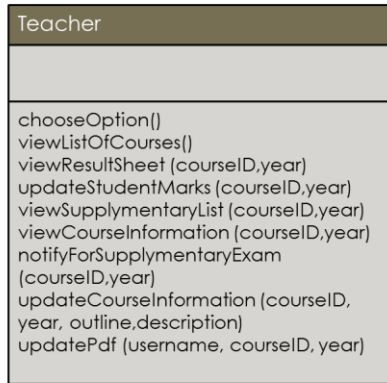
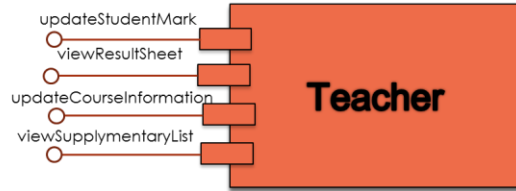


Figure 33: Class elaboration of Student

1 Analysis class



2 Design Components



3 Elaborated Design Class

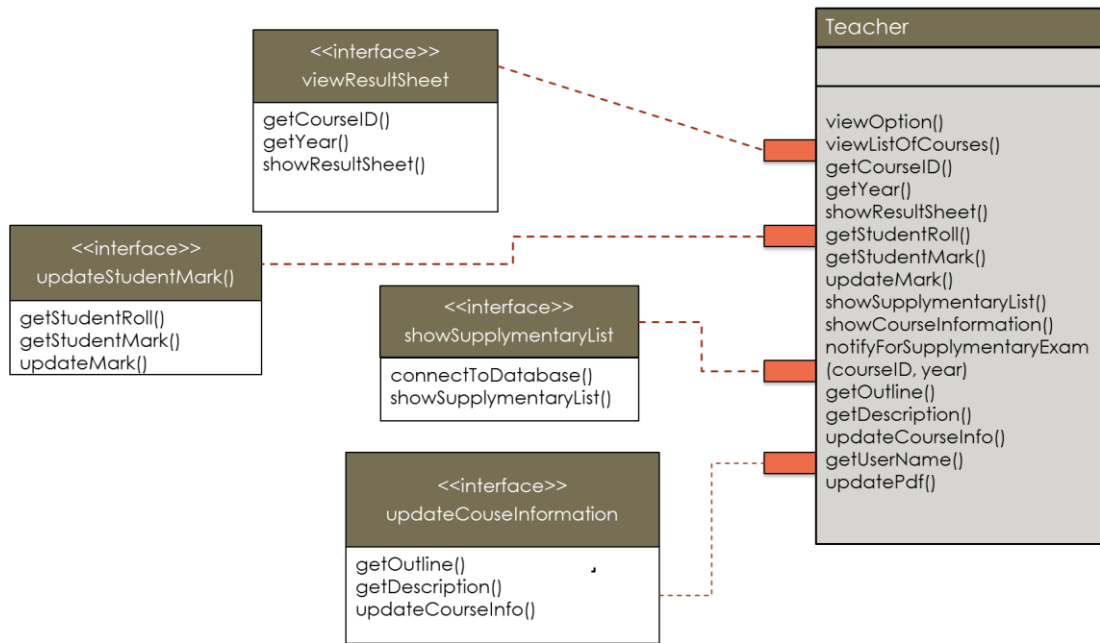


Figure 34: Class elaboration of Teacher

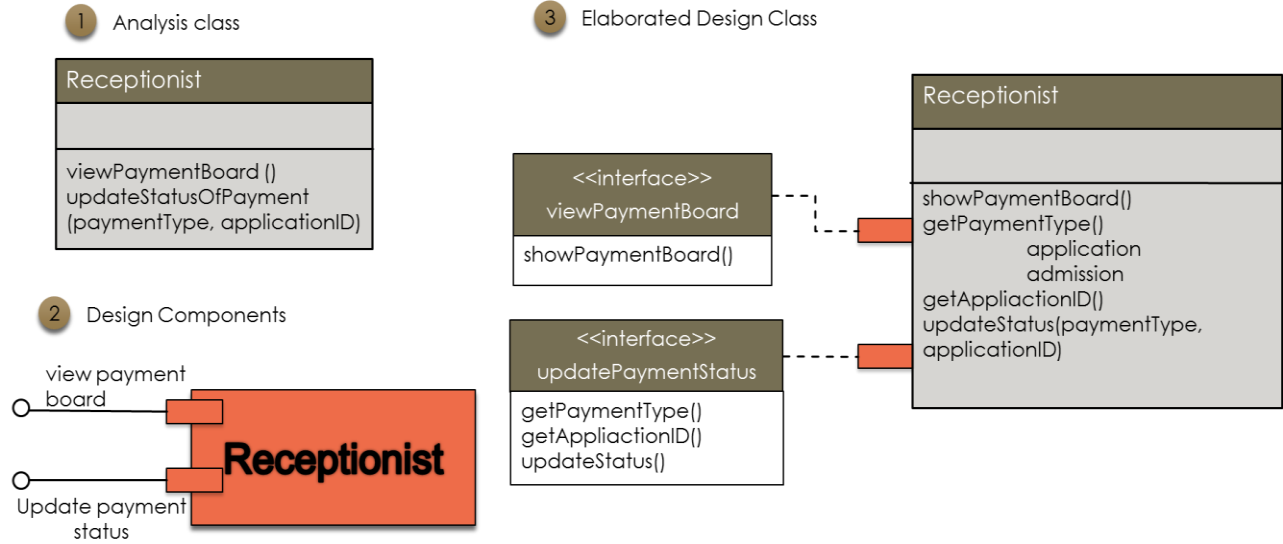


Figure 35: Class elaboration of Receptionist

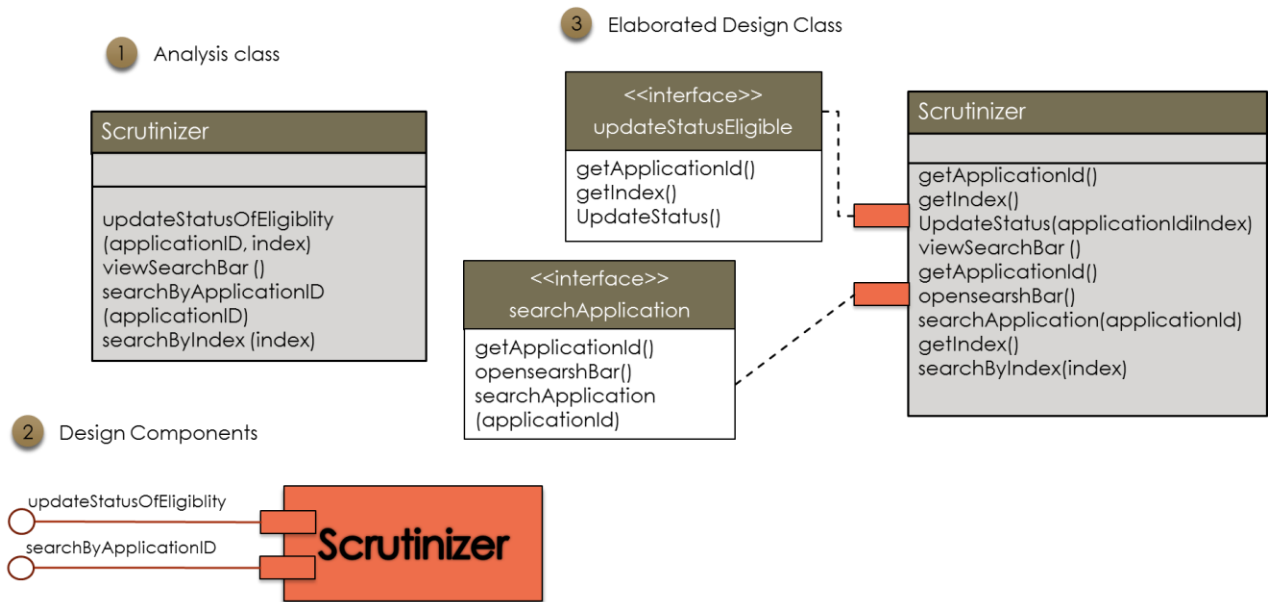
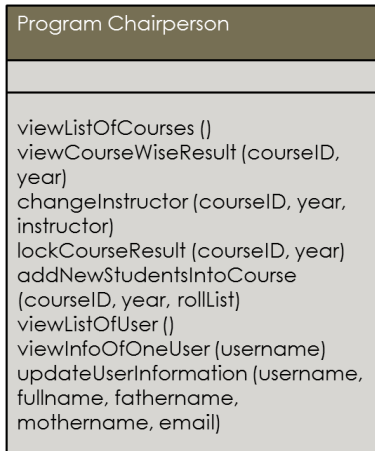
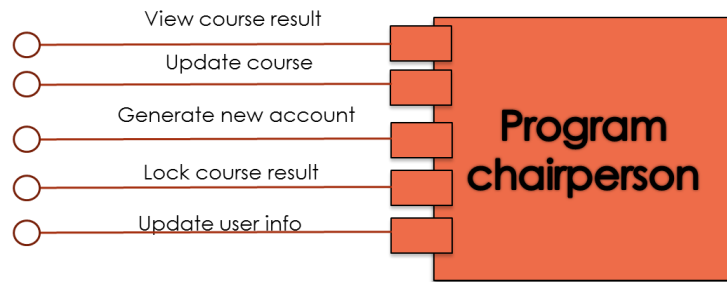


Figure 36: Class elaboration of Scrutinizer

1 Analysis class



2 Design Components



3 Elaborated Design Class

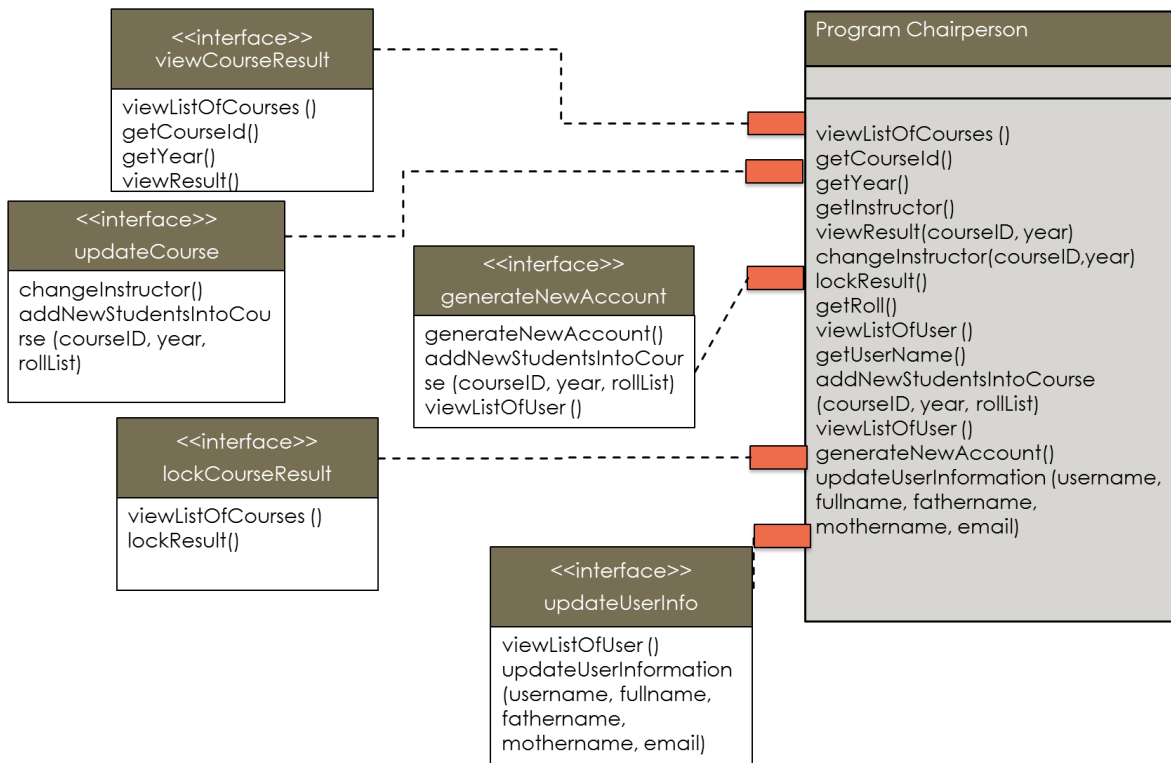
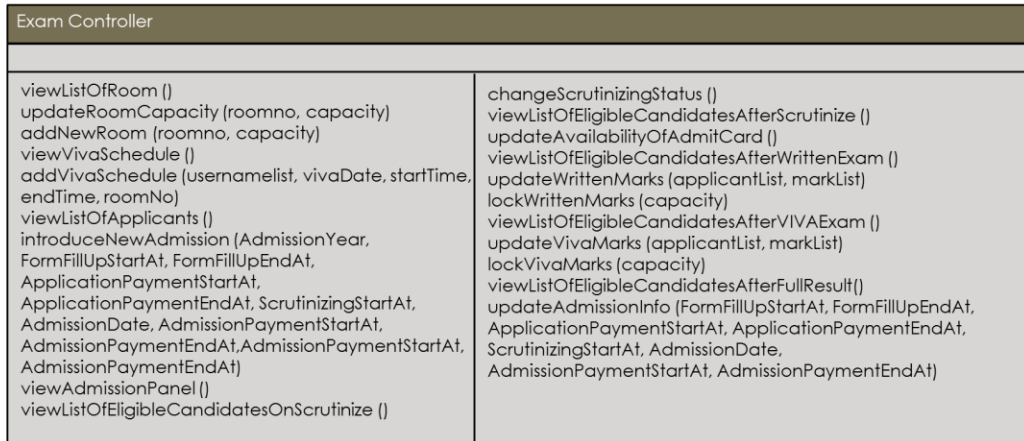
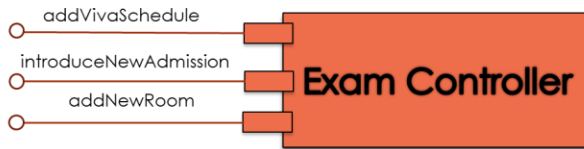


Figure 37: Program Chairperson

1 Analysis class



2 Design Components



3 Elaborated Design Class

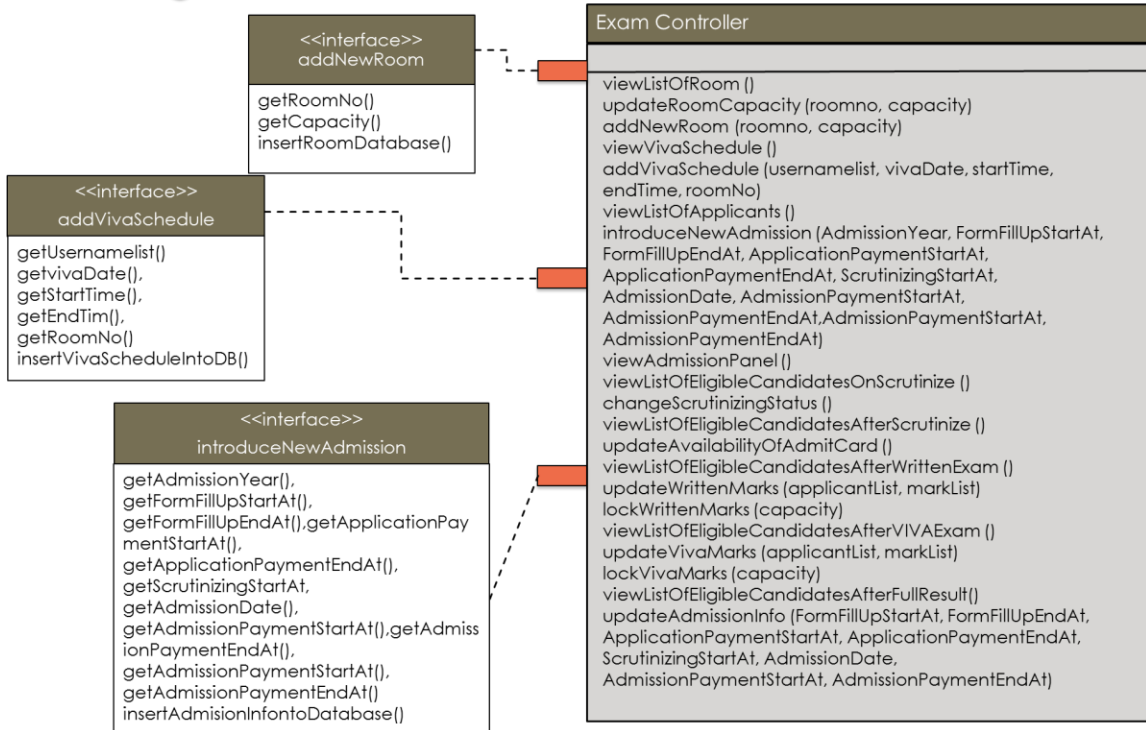
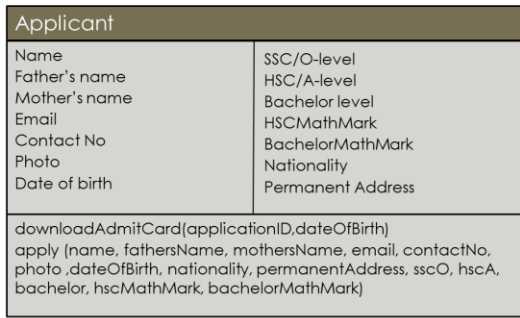
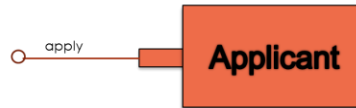


Figure 38: Class elaboration of Exam Controller

1 Analysis class



2 Design Components



3 Elaborated Design Class

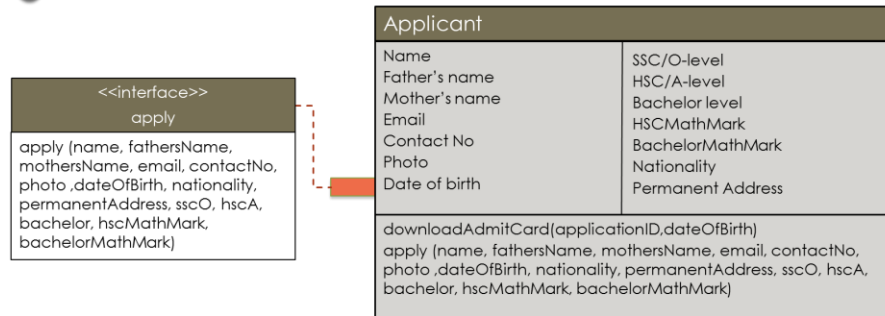
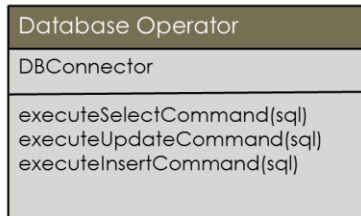
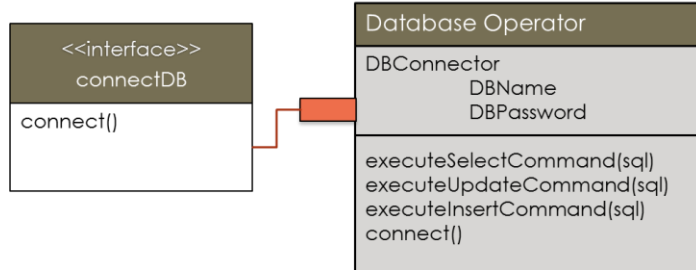


Figure 39: Class elaboration of Applicant

1 Analysis class



3 Elaborated Design Class



2 Design Components

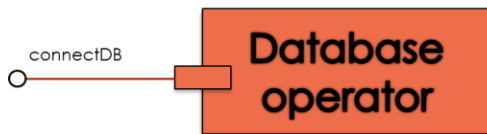


Figure 40: Class elaboration of Database operator

2.3 a) Class collaboration

Sign in

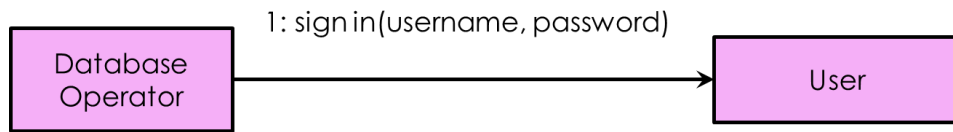


Figure 41: Class collaboration of authentication

Update Profile

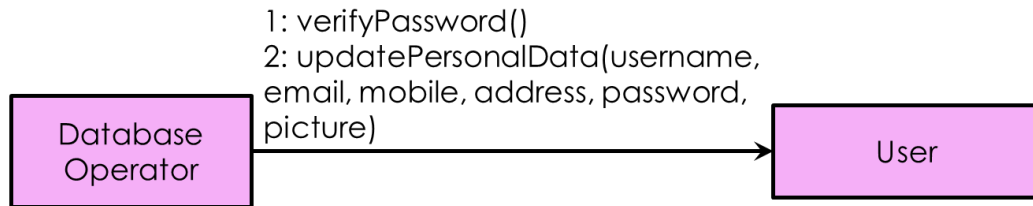


Figure 42: Class collaboration of update Profile

New Admission Session

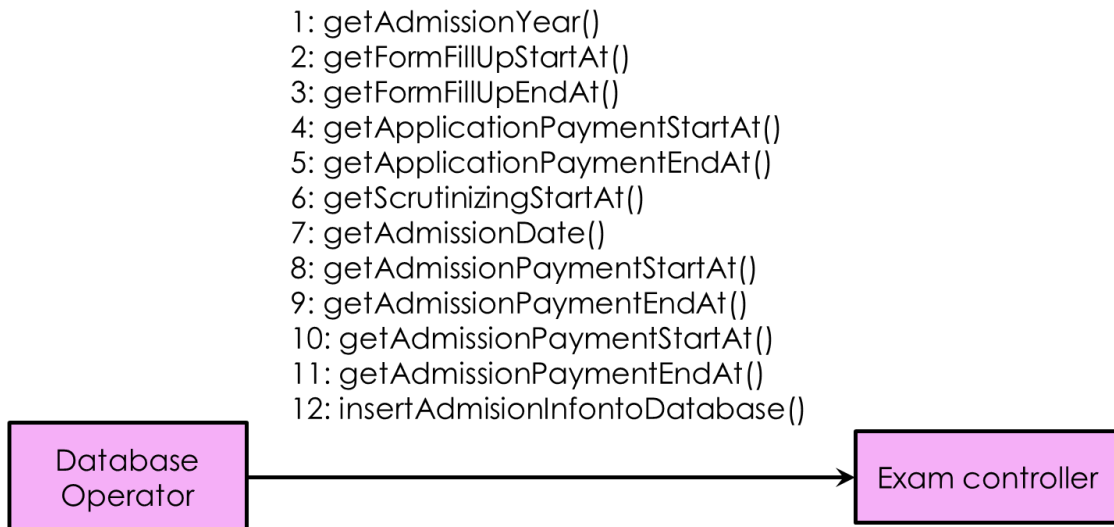


Figure 43: Class collaboration of new admission session

Apply

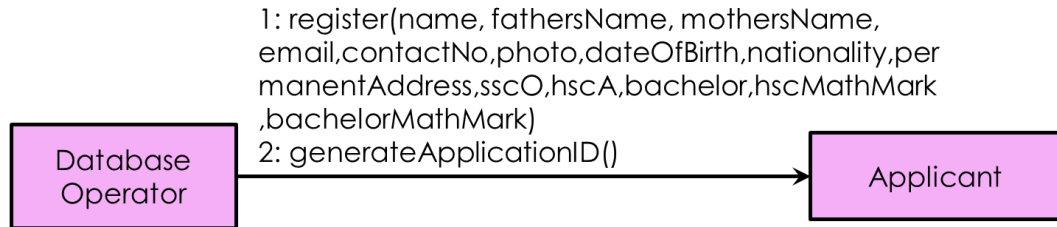


Figure 44: Class collaboration of Apply

View Payment Board



Figure 45: Class collaboration of view payment board

Update Payment Status

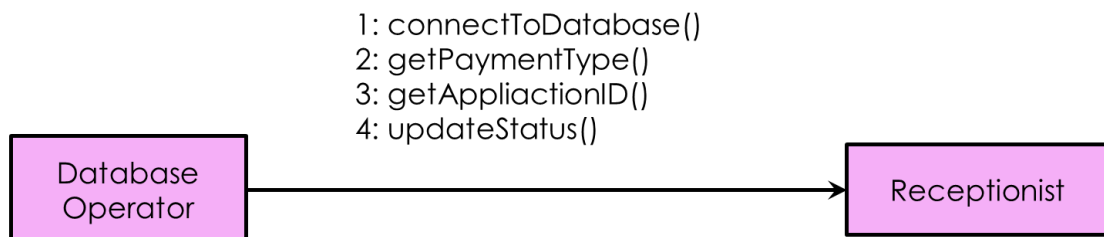


Figure 46: Class collaboration of update payment status

Update Status of Eligibility

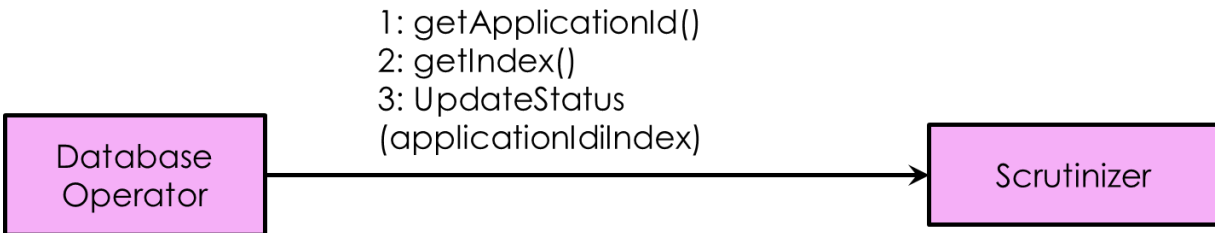


Figure 47: Class collaboration of update status of eligibility

Add Viva Schedule

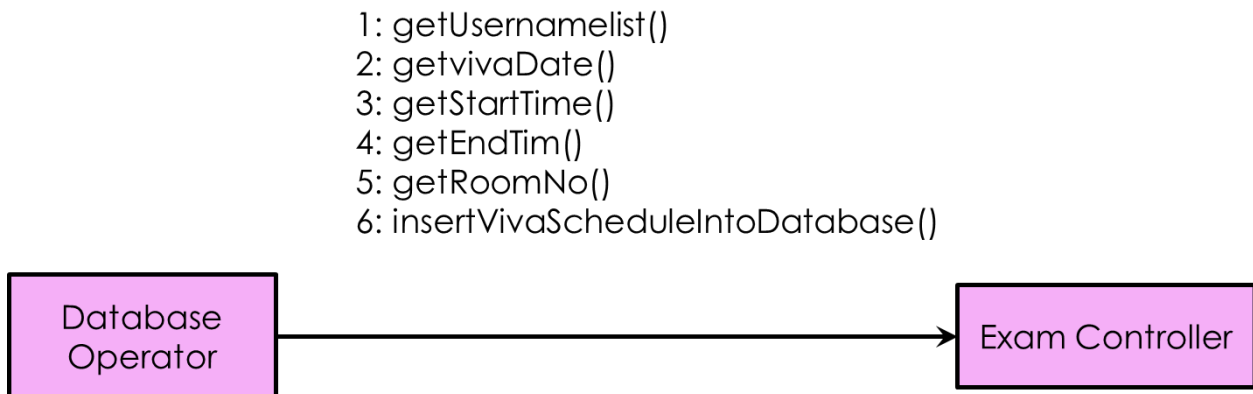


Figure 48: Class collaboration of add viva schedule

Add new room

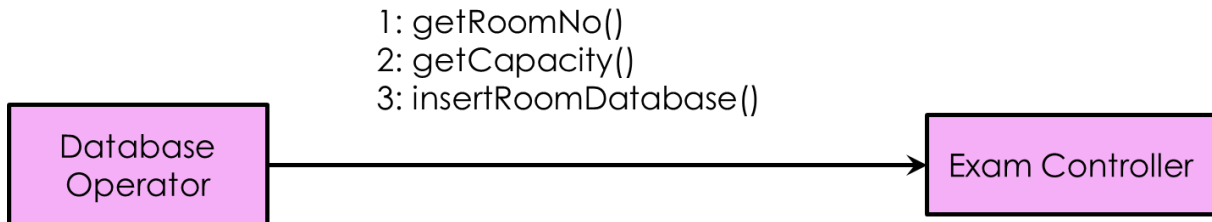


Figure 49: Class collaboration of add new room

Result submission

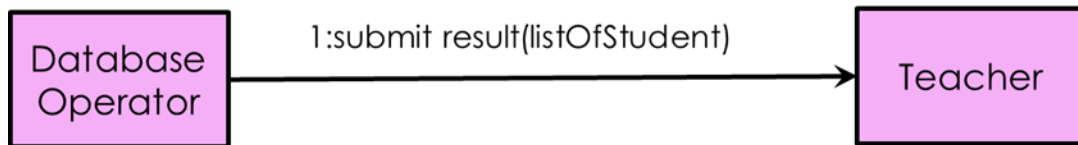


Figure 50: Class collaboration of result submission

View Result

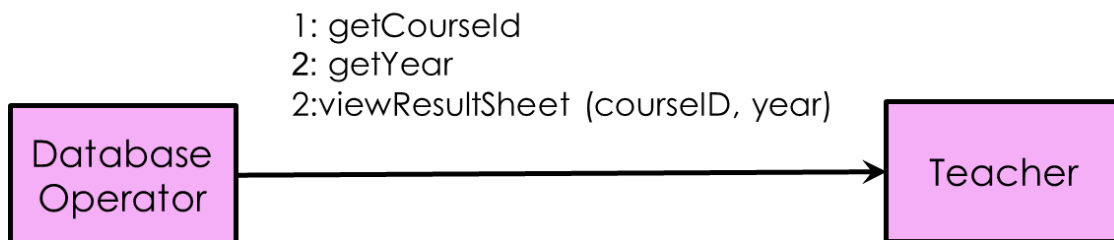


Figure 51: Class collaboration of view result

Update Result

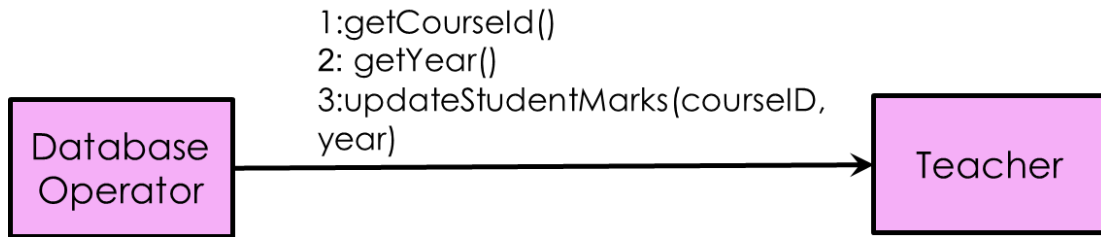


Figure 52: Class collaboration of update result

View SupplementaryList

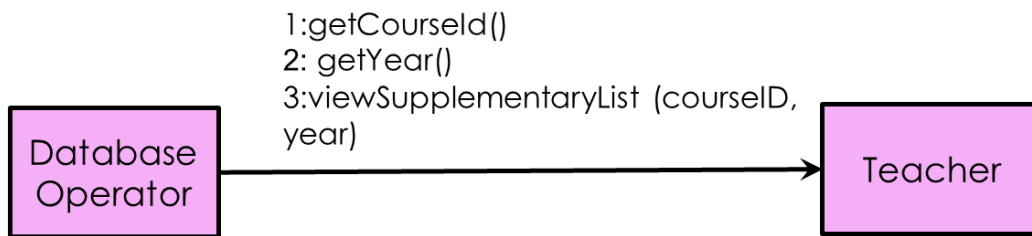


Figure 53: Class collaboration of view supplementary list

View Course Result

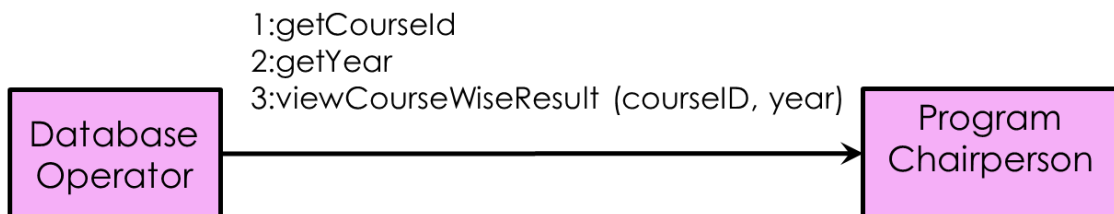


Figure 54: Class collaboration f view course result

2.3 b) Method call sequence for individual modules

Authentication

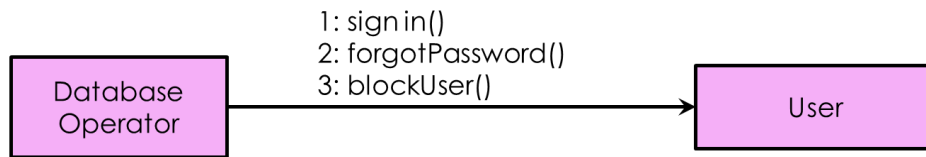


Figure 55: Method call sequence of Authentication

Teacher Registration

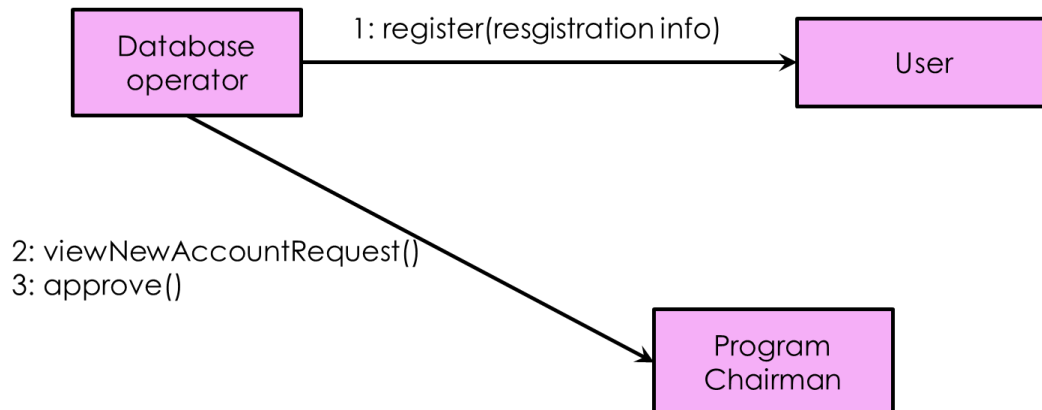


Figure 56: Method call sequence of Teacher registration

Student Account Creation



Figure 57: Method call sequence of Student account creation

Candidate Management

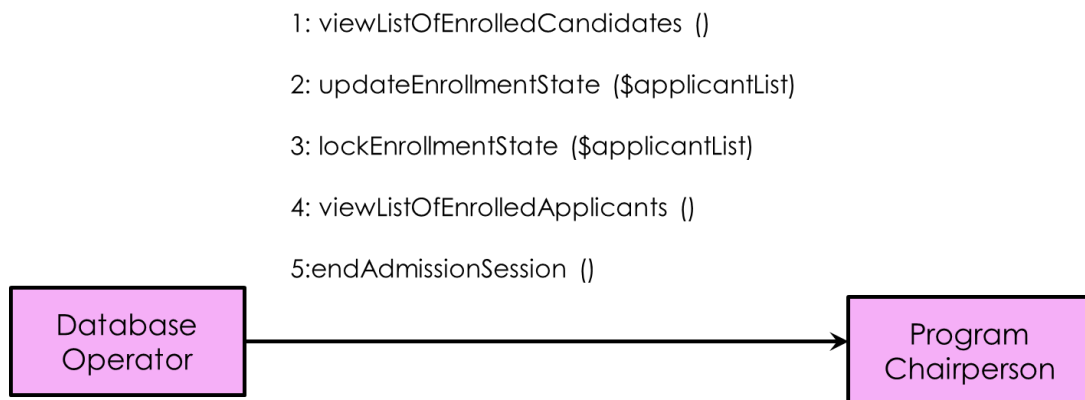


Figure 58: Method call sequence of Candidate management

Course Management

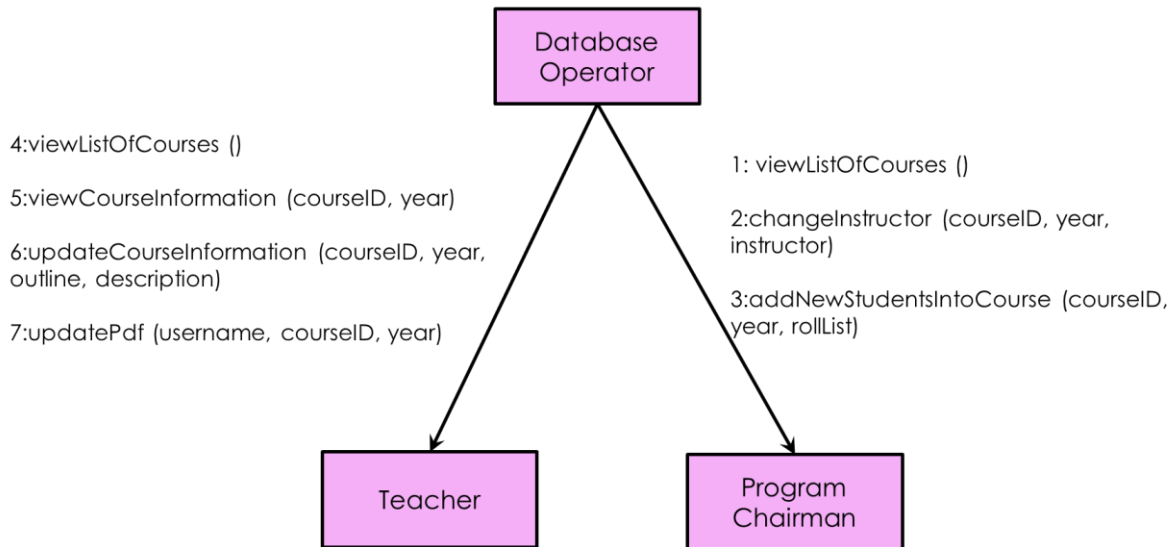


Figure 59: Method call sequence of Course management

Result Management

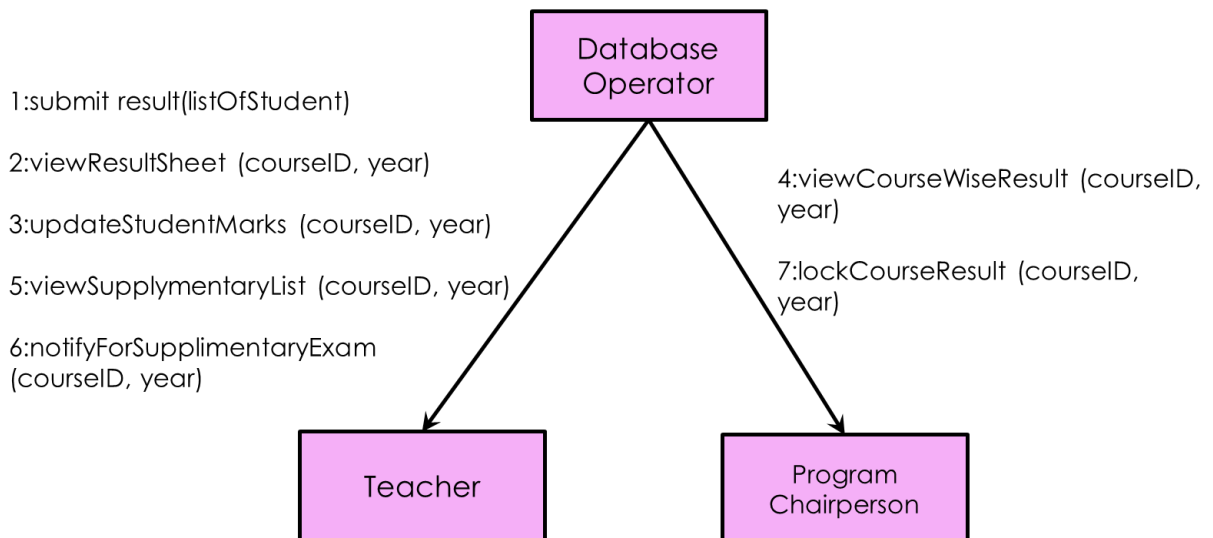


Figure 60: Method call sequence of Result management

User Management

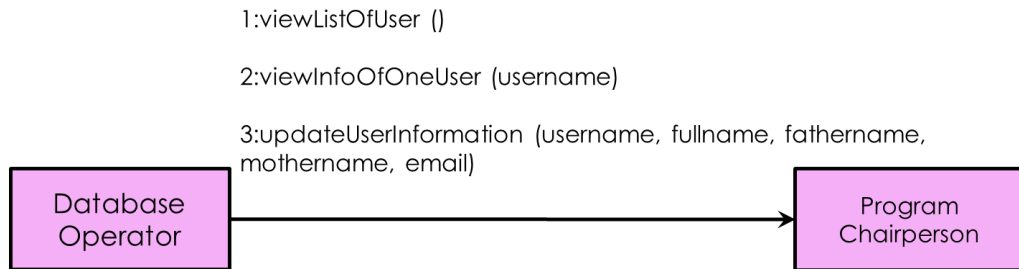


Figure 61: Method call sequence of User management

Admission Session

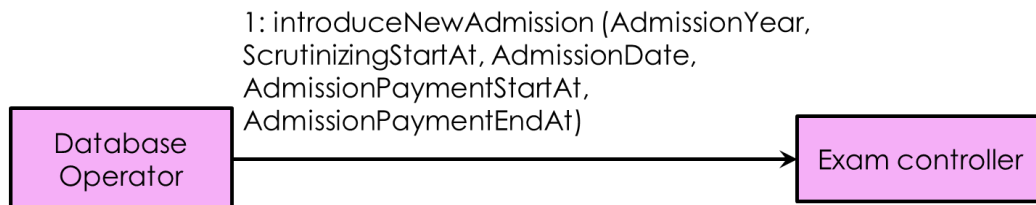


Figure 62: Method call sequence of Admission session

Exam Schedule Manage

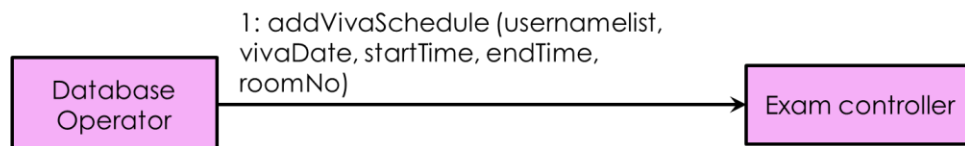


Figure 63: Method call sequence of Exam schedule manage

Result Manage

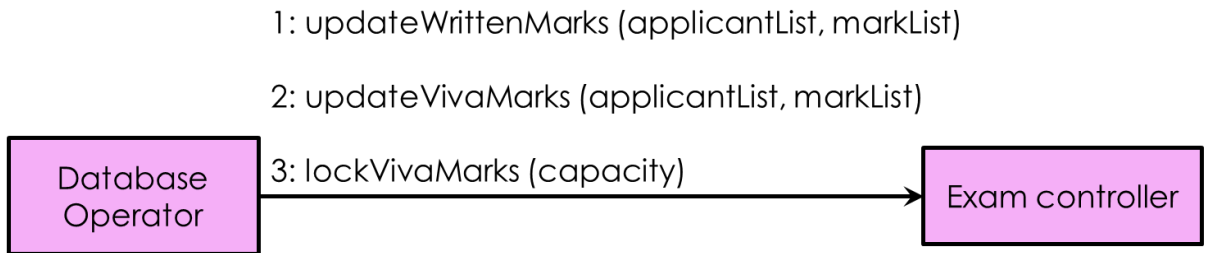


Figure 64: Method call sequence of Result manage

Application Processing

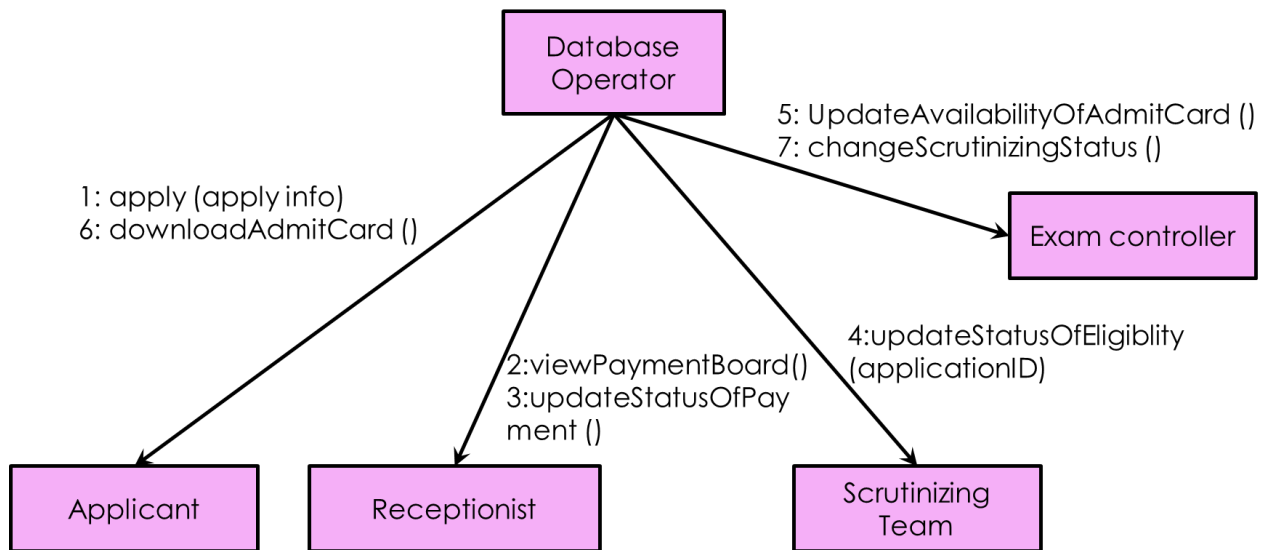


Figure 65: Method call sequence of Application processing

2.3 c) Appropriate Interfaces

There is no necessity to divide the classes in subclasses as they exhibit sufficient cohesion. So there is no need to define appropriate interfaces.

2.3 d) Elaborate Attributes

1. Username: string = null { contains name value 20 character max}
2. EmailAddress: string = null { contains email – xyz@gmail.com}
3. Password: string = null {4<=number of character<=20}
4. MobileNumber: string = null { contains number value 11 character max}
5. Name: string = null { contains name value 50 character max}
6. RollNumber: int = null {10 character max}
7. DBName: string = null {contains database name value 10 character max}
8. DBPassword: string = null {5<=number of characters<=10}
9. CourseID: int = null {5 digit}
10. Year: int = null {4 digit}
11. Outline: string = null {number of characters<=500}
12. Description:String=null {number of characters<=1000}
13. Username:String=null {number of characters<=15}
14. FullName:String=null {number of characters<=40}
15. FathersName:String=null {number of characters<=40}
16. MothersName : String=null {number of characters<=40}
17. Instructor : String=null {number of characters<=40}
18. Email : String=null {number of characters<=40}
19. RollList : int =null {List of students roll number}
20. Photo : String=null{filepath}
21. Nationality : String=null{number of characters<=15}
22. ContactNumber : int =null {15 digit}
23. PermanentAddress : String =null {number of characters<=100}
24. HscMathMark : int=null {3 digits}
25. BatchelorMathMark : int=null {3 digit}
26. schoolName: String =null {number of characters<=100}

27. schoolBoard: String =null {number of characters<=100}
28. schoolGroup: String =null {number of characters<=100}
29. schoolYear: int = null {4 digit}
30. schoolGPA: String =null {number of characters<=100}
31. collegeName: String =null {number of characters<=100}
32. collegeBoard: String =null {number of characters<=100}
33. collegeGroup: String =null {number of characters<=100}
34. collegeYear: int = null {4 digit}
35. collegeGPA: String =null {number of characters<=100}
36. universityName: String =null {number of characters<=100}
37. departmentName: String =null {number of characters<=100}
38. universityYear: int = null {4 digit}
39. universityGPA: String =null {number of characters<=100}

2.3 e) Describe Processing Flow

Processing flow of the methods: Teacher class

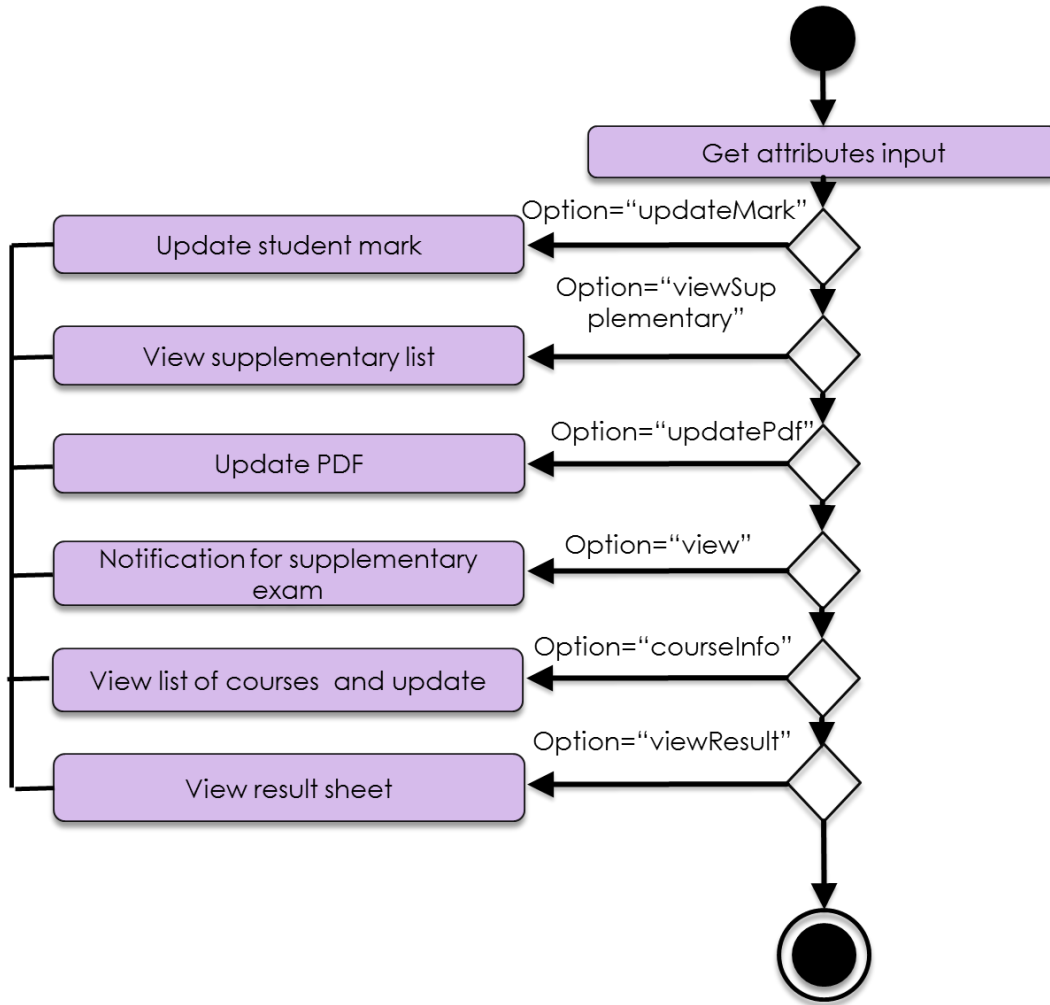


Figure 66: Processing flow of chooseOption

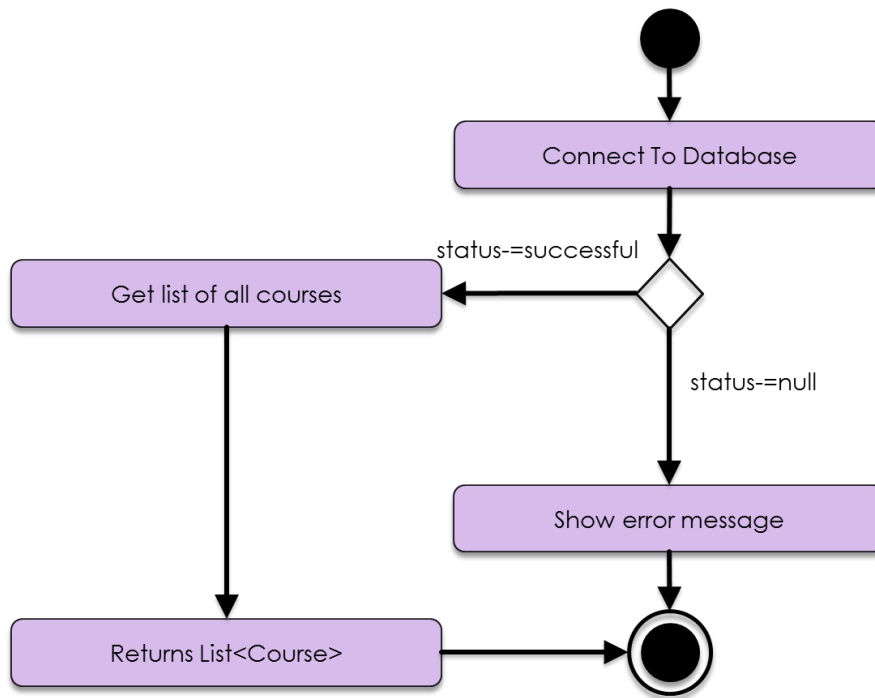


Figure 67: Processing flow of viewListOfCourse()

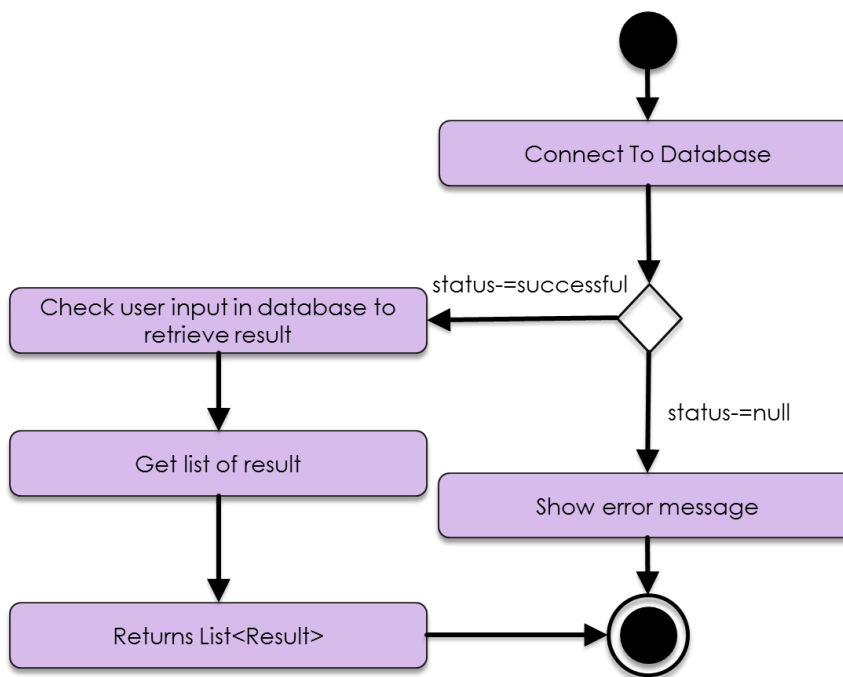


Figure 68: Processing flow of viewResultSheet(courseID,year)

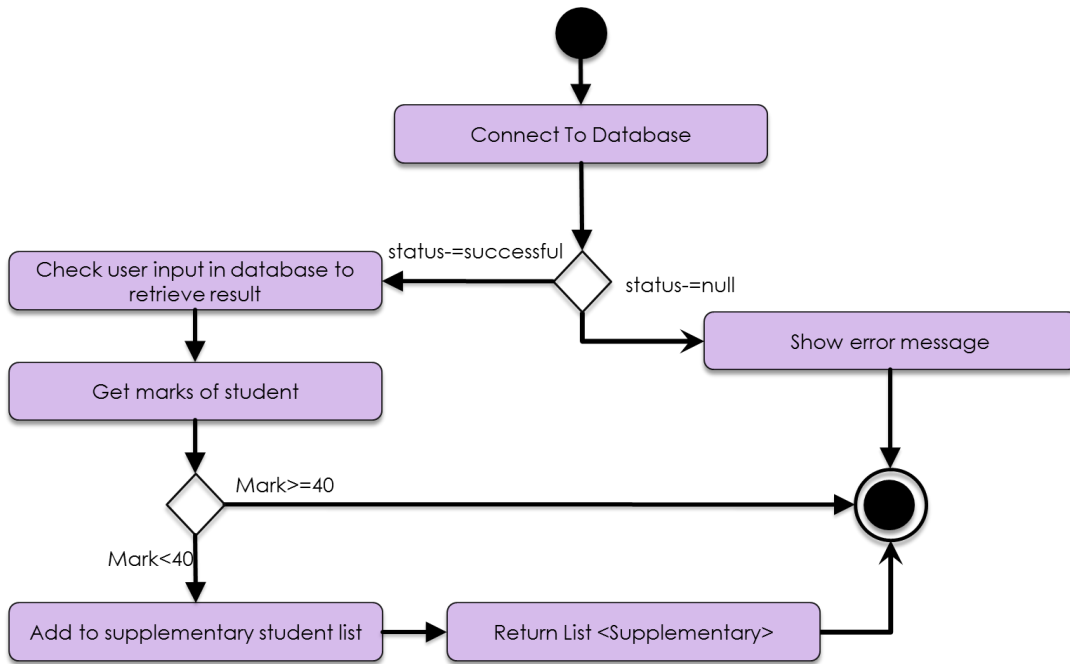


Figure 69: Processing flow of viewSupplementaryList(courseID, year)

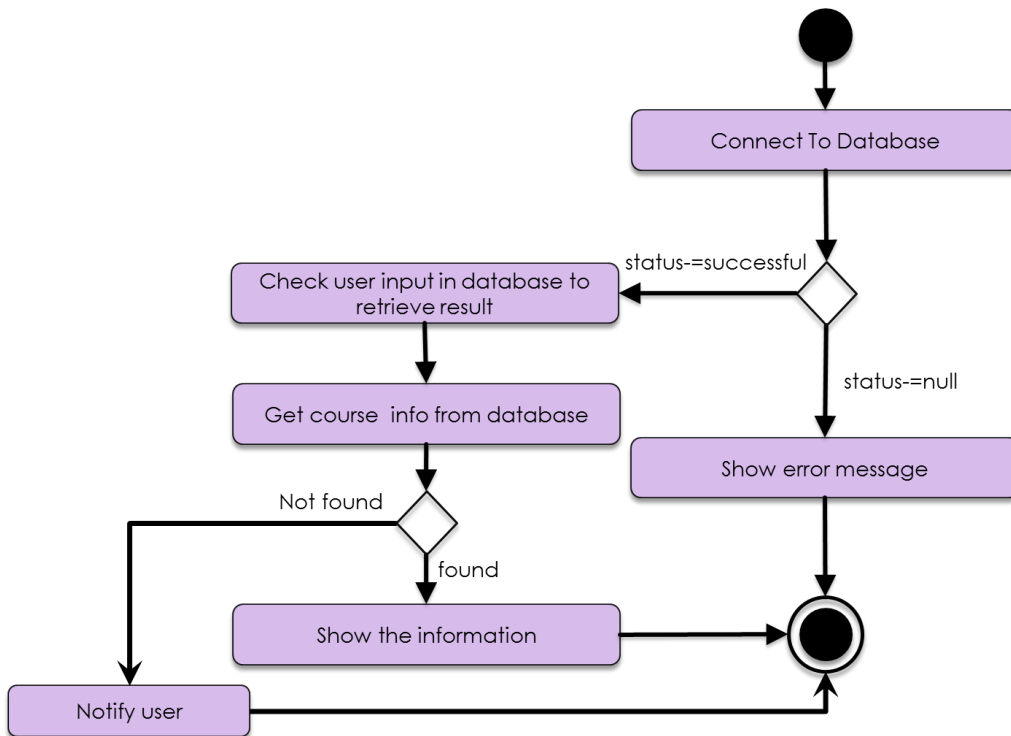


Figure 70: Processing flow of viewCourseInformation(courseID, year)

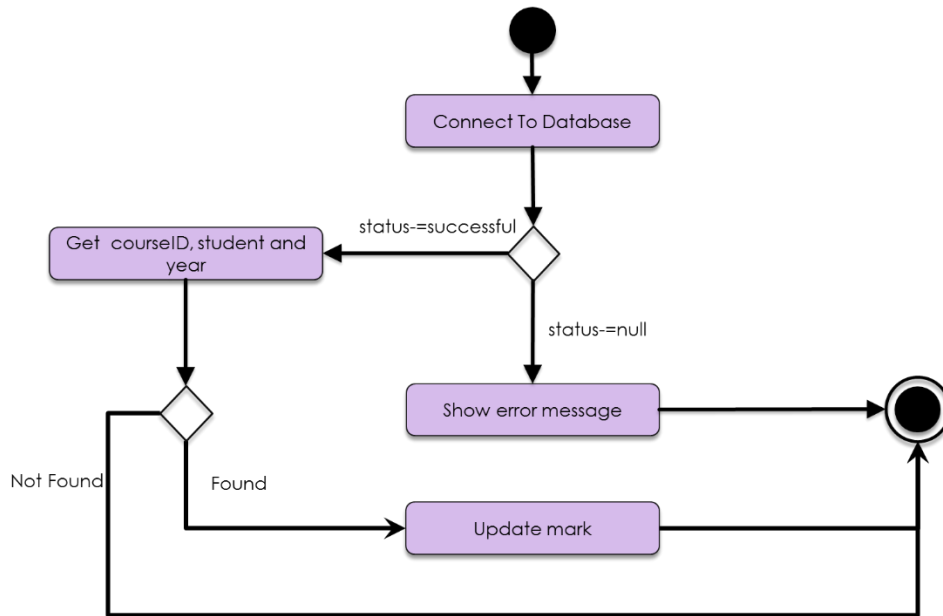


Figure 71: Processing flow of `updateStudentMark(courseID, year, mark, studentID)`

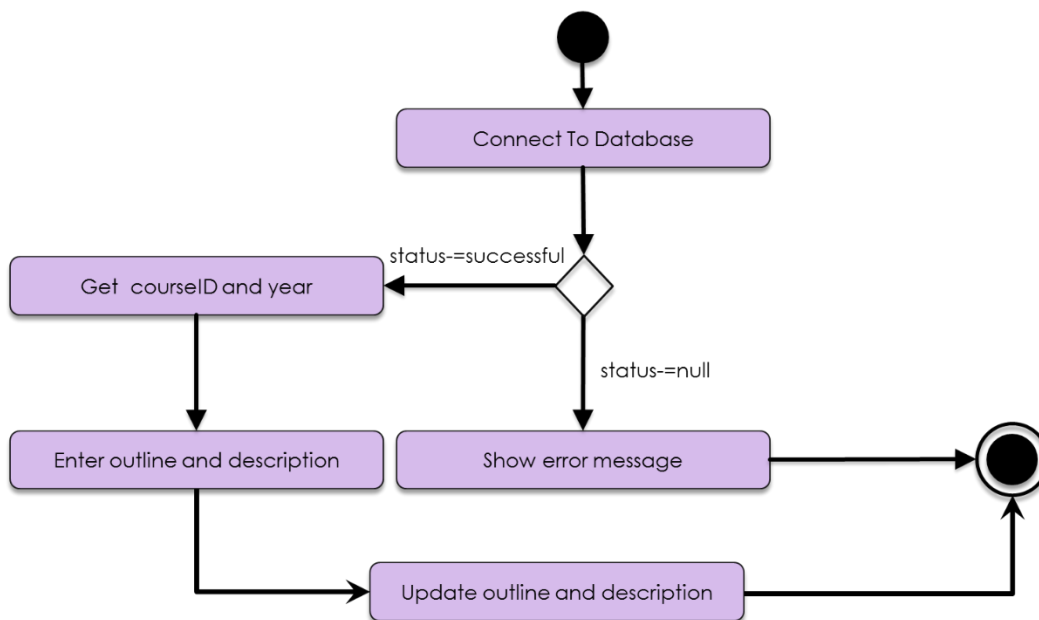


Figure 72: Processing flow of `updateCourseInformation(courseID, year, outline, description)`

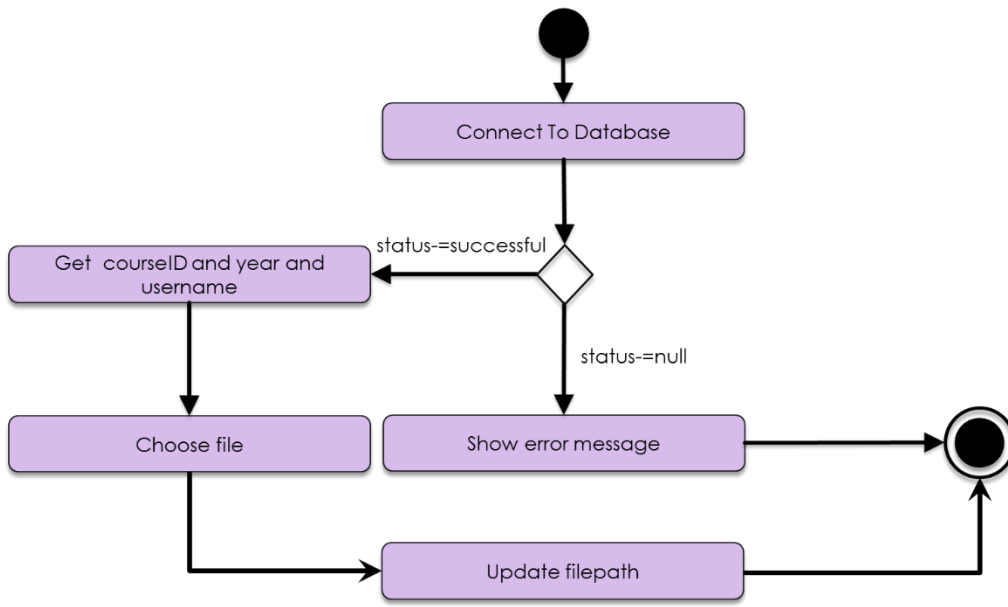


Figure 73: Processing flow of `updatePDF(courseID, year, username)`

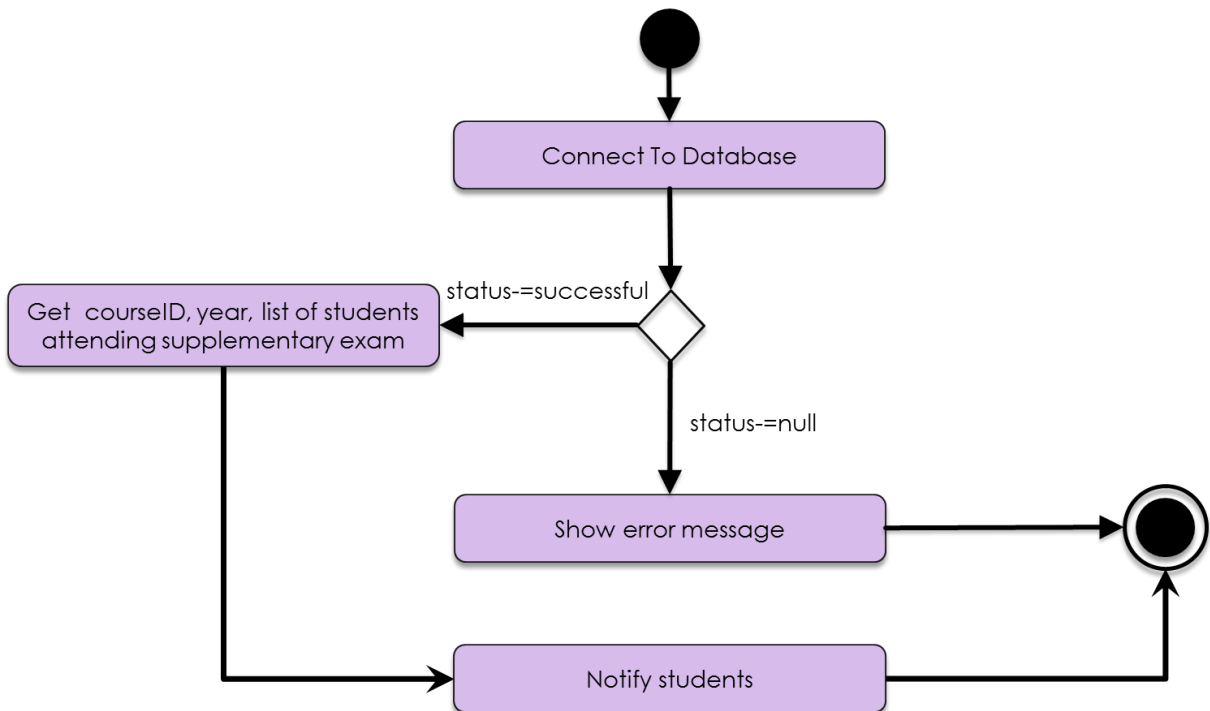


Figure 74: Processing flow of `notifyForSupplementaryExam(courseID, year)`

Processing flow of the methods: Database Operator class

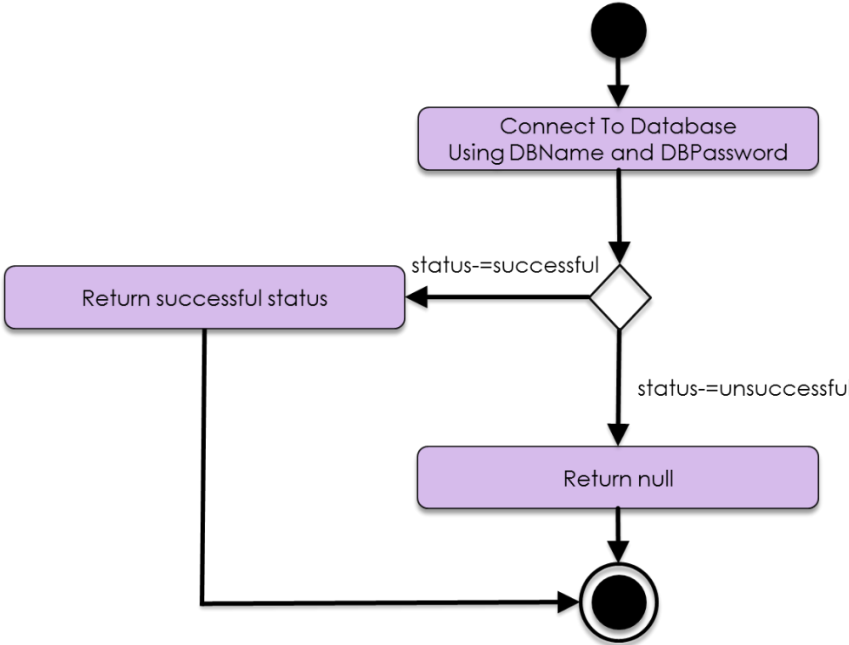


Figure 75: Processing flow of Connect()

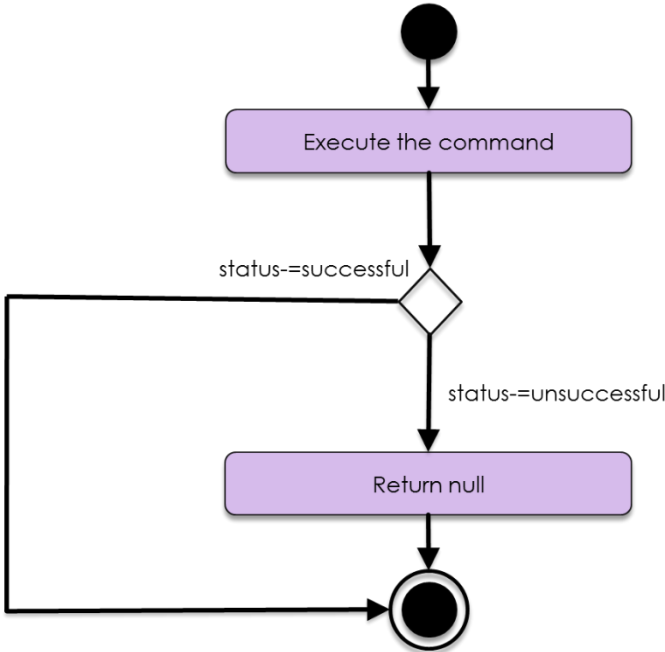


Figure 76: Processing flow of executeSelectCommand(sql)

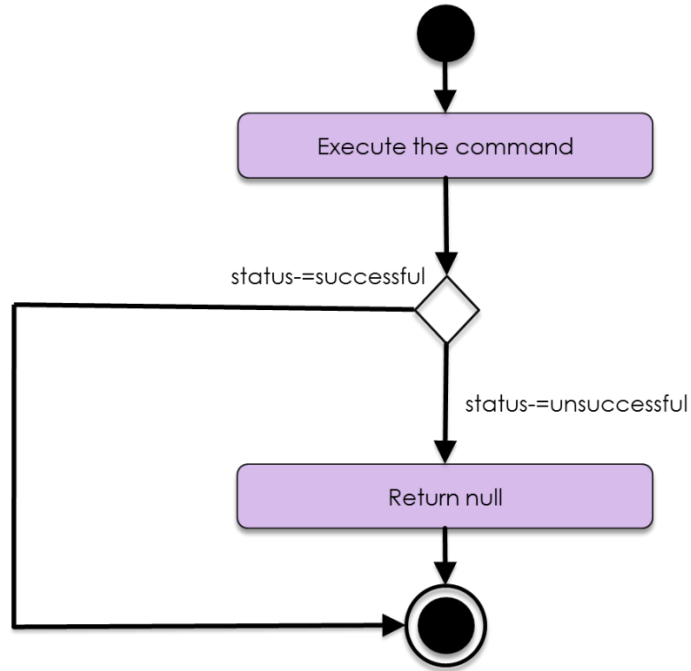


Figure 77: Processing flow of `executeUpdateCommand(sql)`

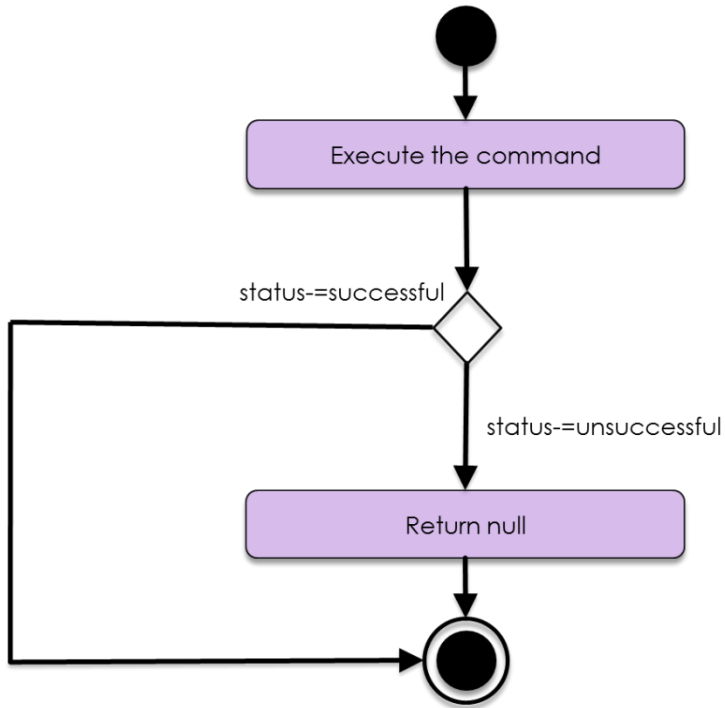


Figure 78: Processing flow of `executeInsertCommand(sql)`

Processing flow of the methods: Receptionist class

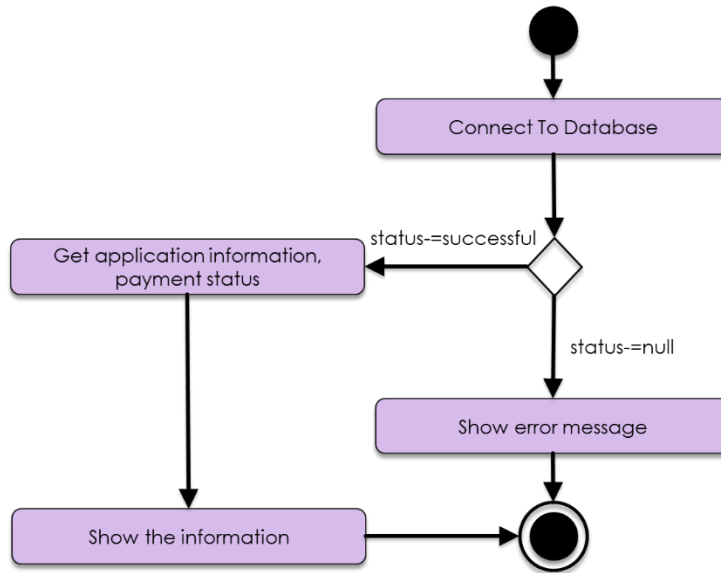


Figure 79: Processing flow of `viewPaymentBoard()`

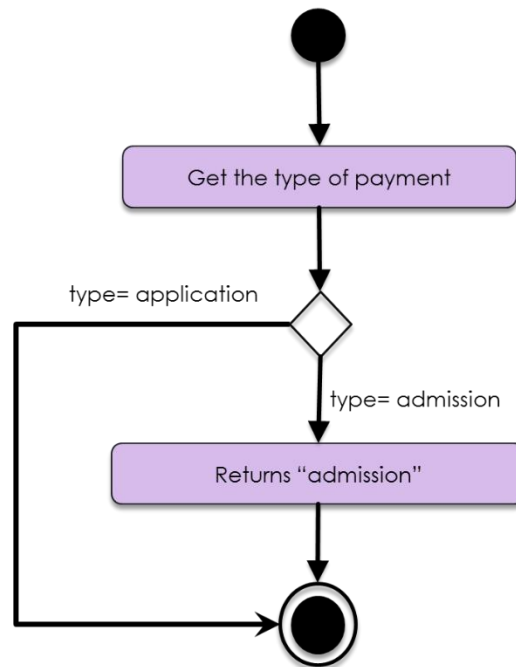


Figure 80: Processing flow of `getPaymentType ()`

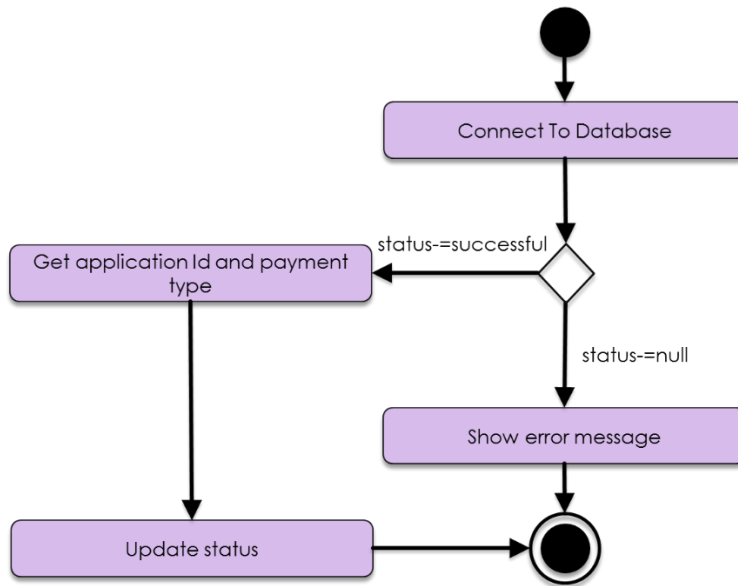


Figure 81: Processing flow of `updatePaymentStatus()`

Processing flow of the methods: Program Chairperson Class

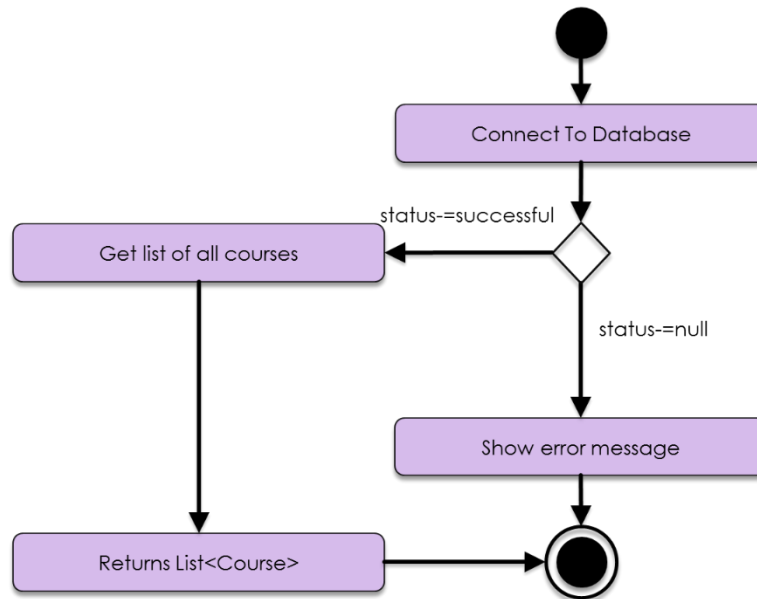


Figure 82: Processing flow of `viewListOfCourses()`

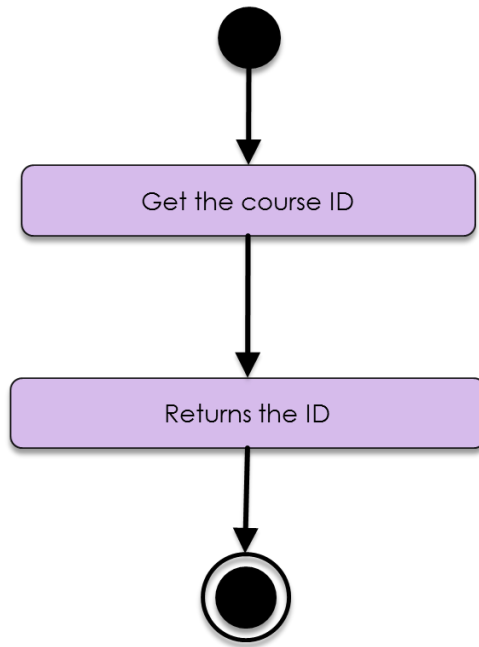


Figure 83: Processing flow of getCourseID ()

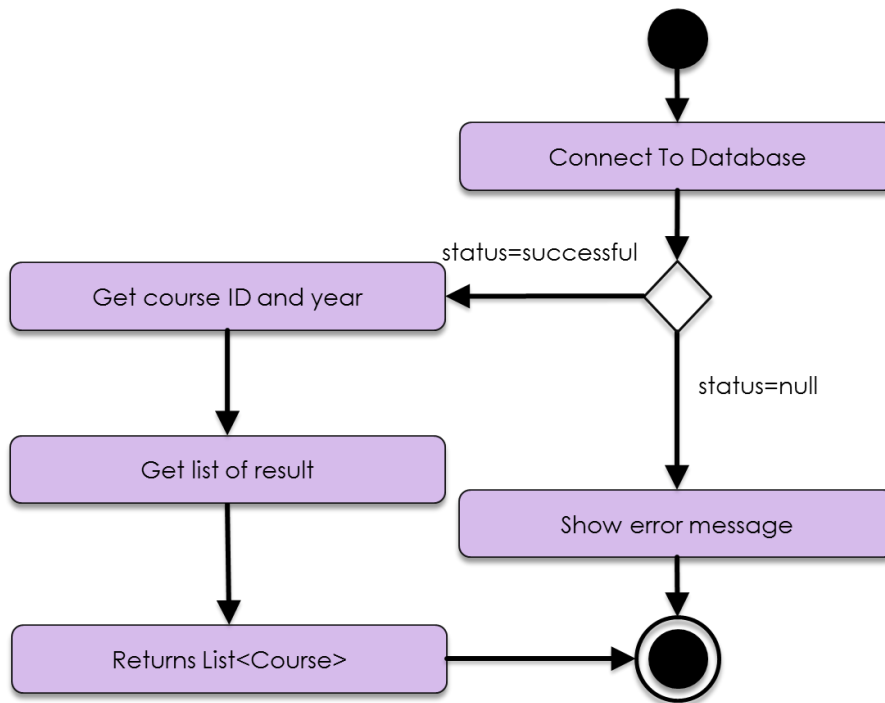


Figure 84: Processing flow of viewResult()

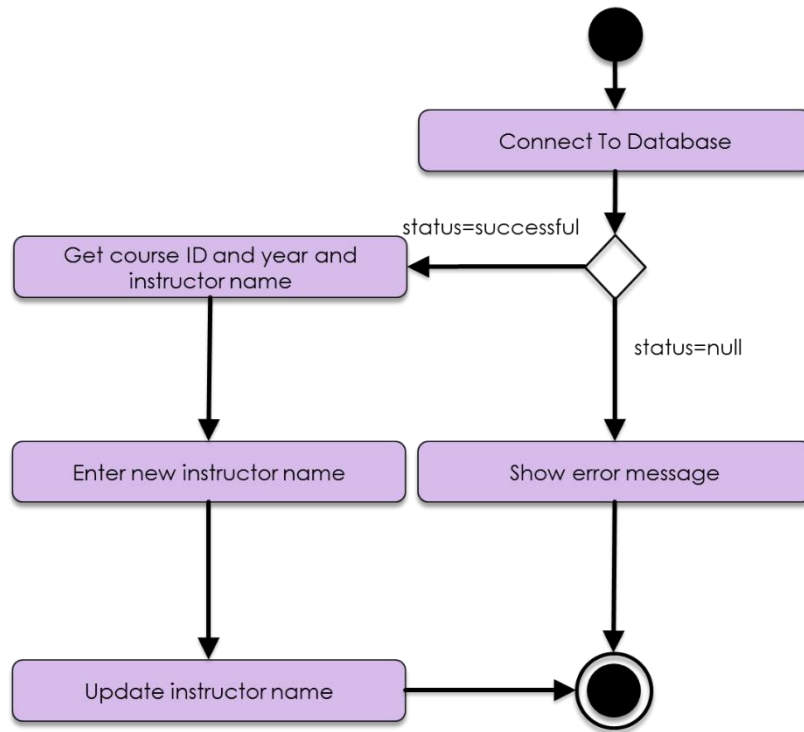


Figure 85: Processing flow of `changeInstructor()`

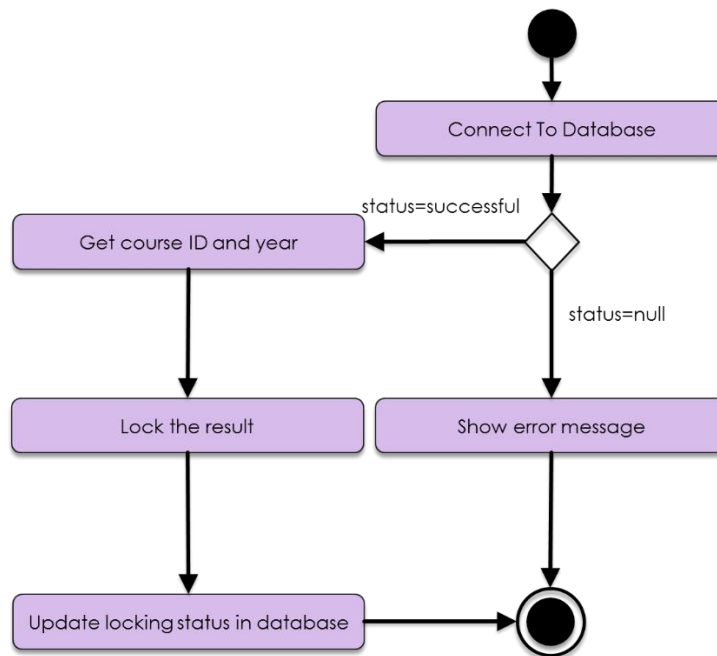


Figure 86: Processing flow of `lockResult()`

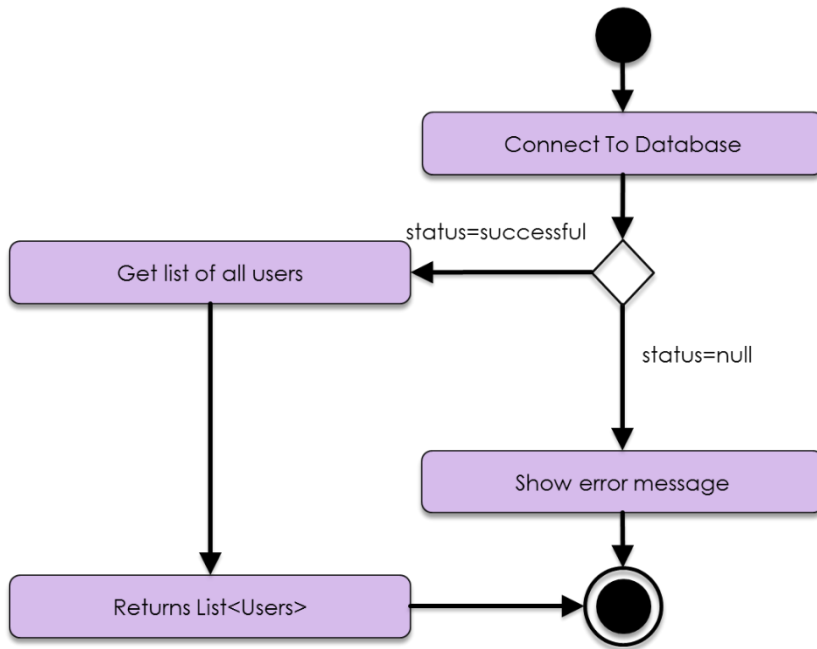


Figure 87: Processing flow of viewListOfUsers()

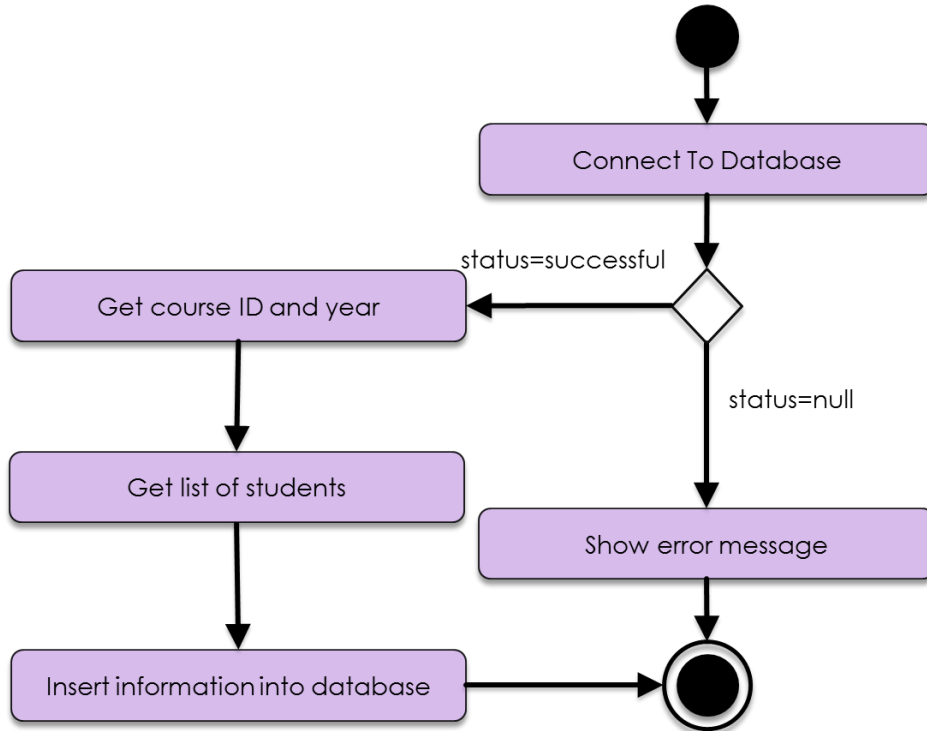


Figure 88: Processing flow of addStudents()

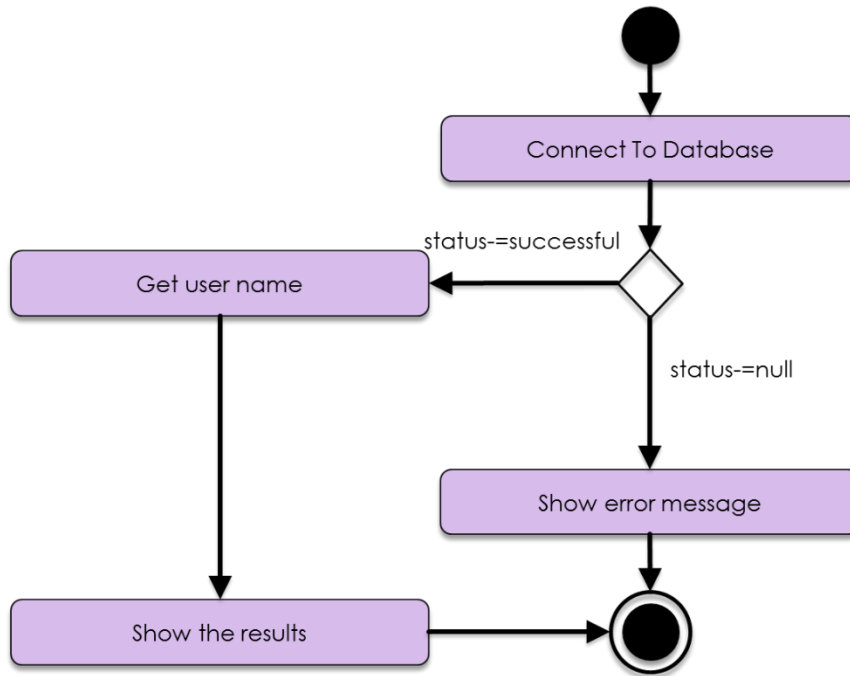


Figure 89: Processing flow of viewUserInfo ()

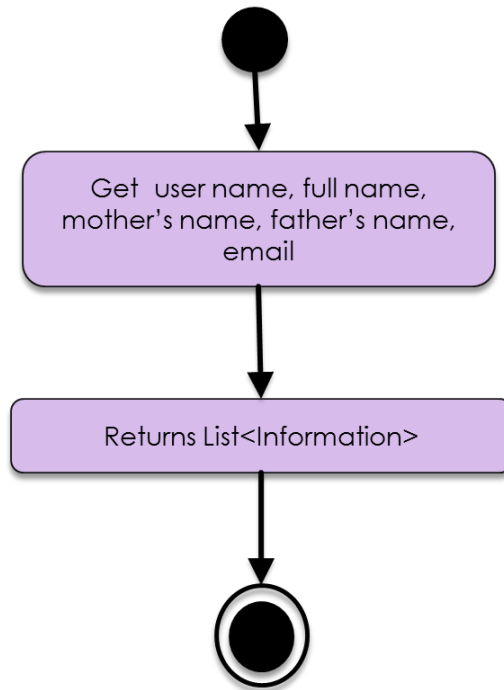


Figure 90: Processing flow of getUserInfo()

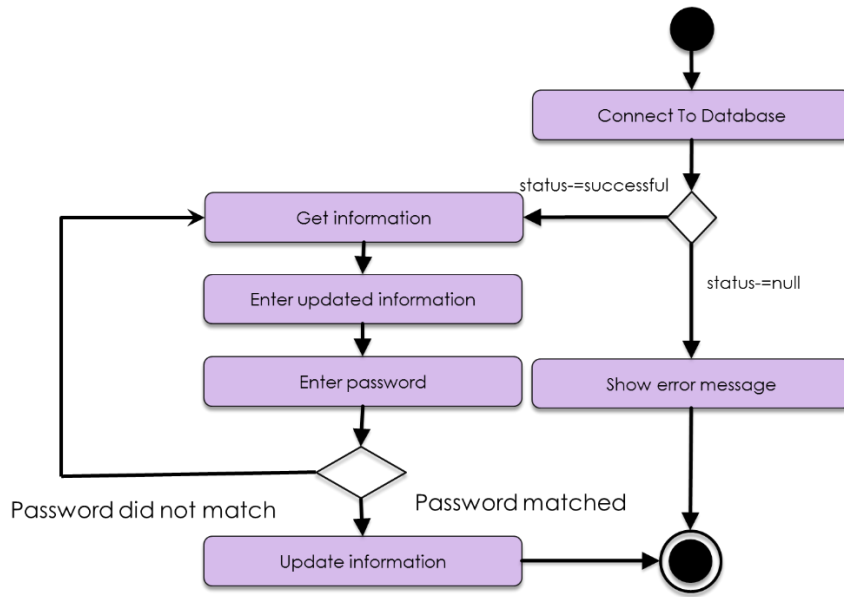
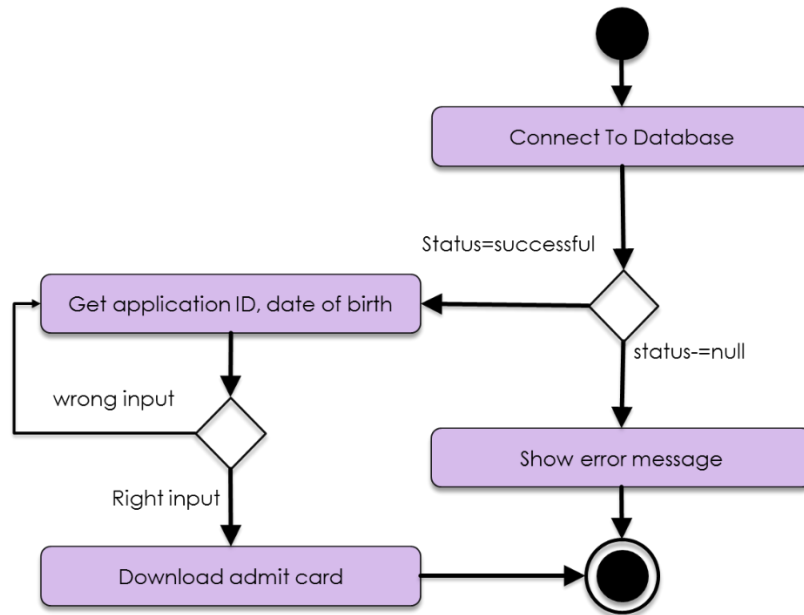


Figure 91: Processing flow of updateInfo()

Processing flow of the methods: Applicant class



**Figure 92: Processing flow of downloadAdmitCard
(applicationID,dateOfBarth)**

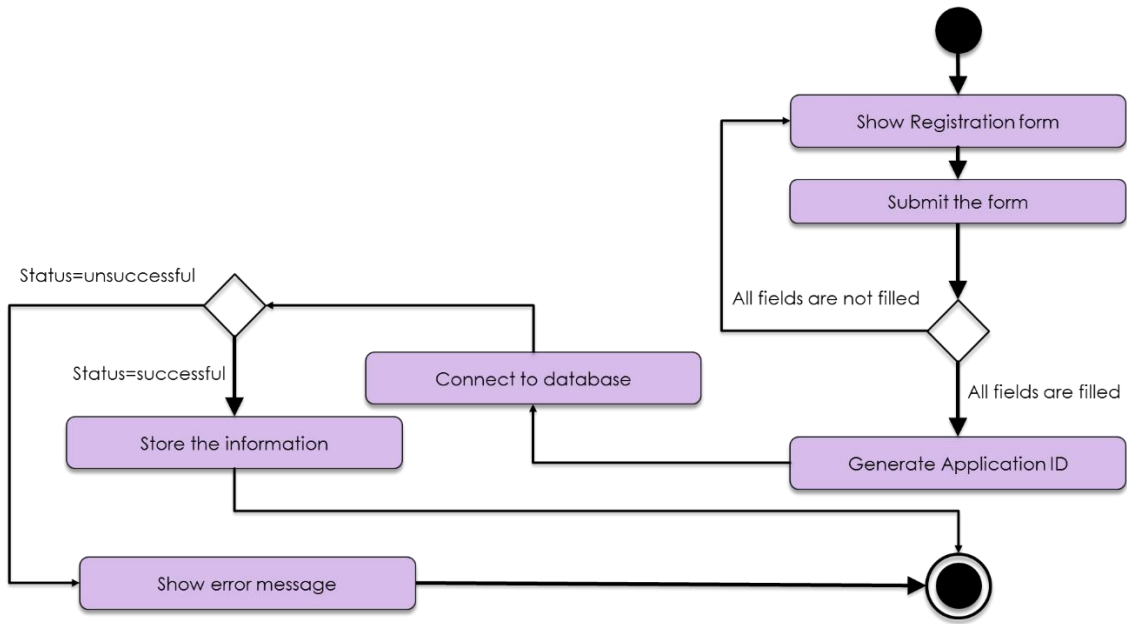


Figure 93: Processing flow of apply(name, fathersName, mothersName, email, contactNo, photo, dateOfBirth, nationality, permanentAddress, sscO, hscA, bachelor, hscMathMark, bachelorMathMark)

Processing flow of the methods: User class

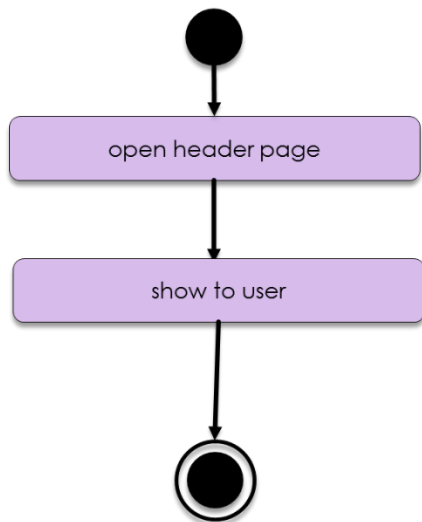


Figure 94: Processing flow of veiwHeader()

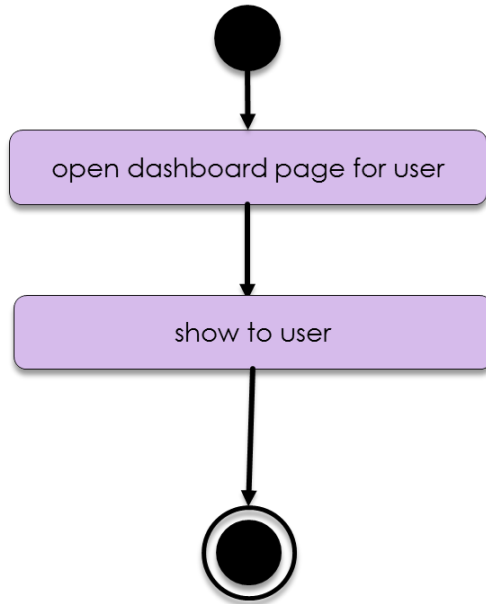


Figure 95: Processing flow of `viewDashboard()`

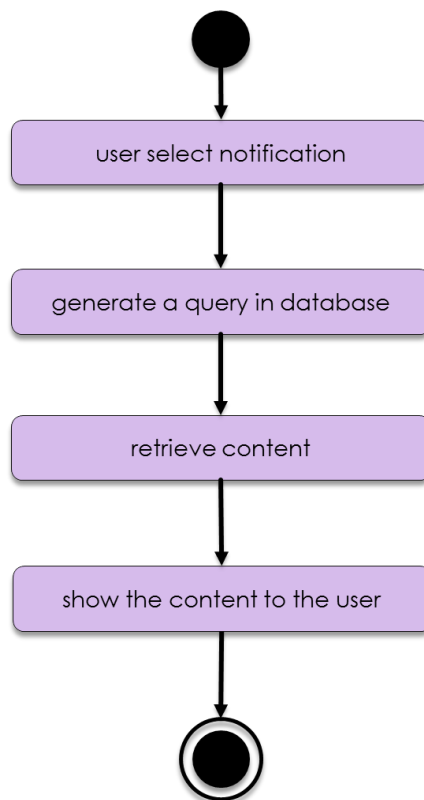


Figure 96: Processing flow of `viewNotification(notificationID)`

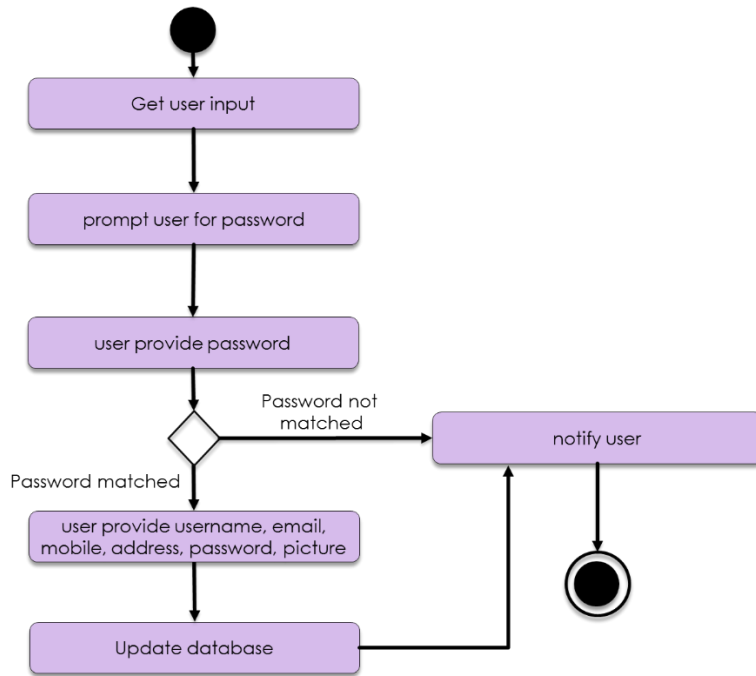
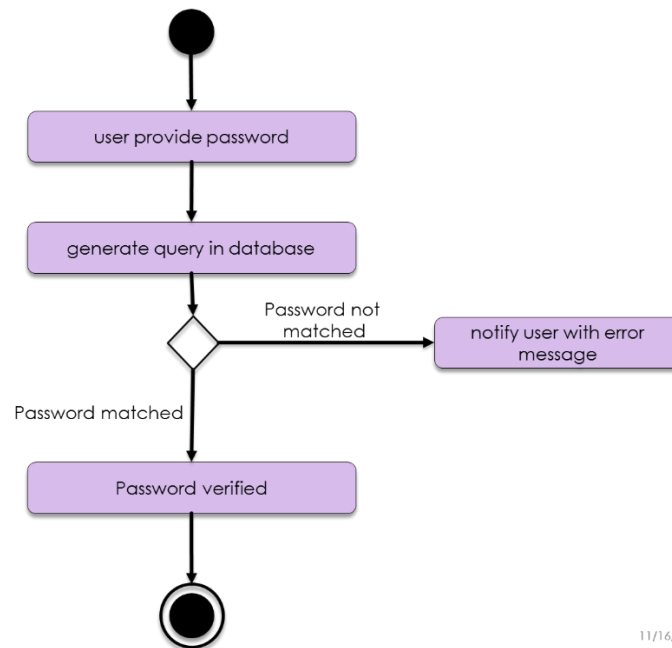


Figure 97: Processing flow of `updatePersonalDate(username, email, mobile, address, password, picture)`



11/16/2017

Figure 98: Processing flow of `verifyPassword(username, password)`

Processing flow of the methods: Exam Controller Class

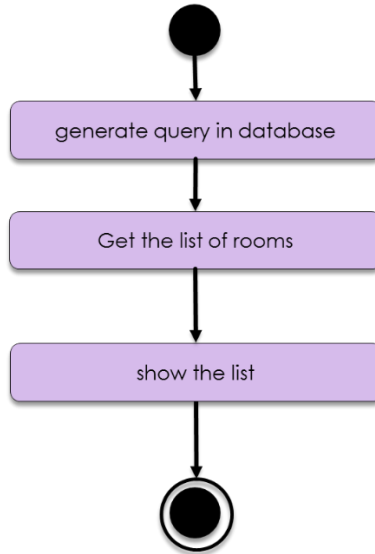


Figure 99: Processing flow of `viewListOfRoom()`

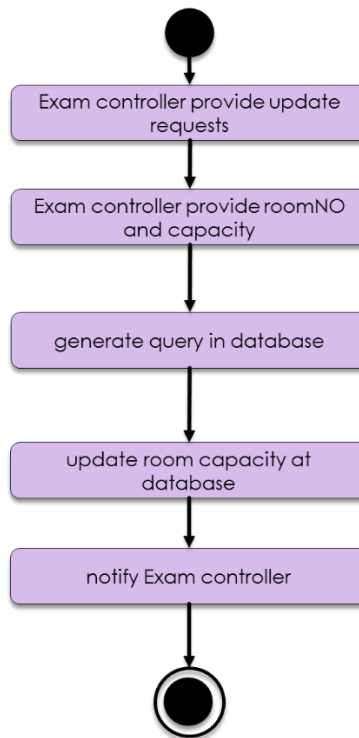


Figure 100: Processing flow of `updateRoomCapacity()`

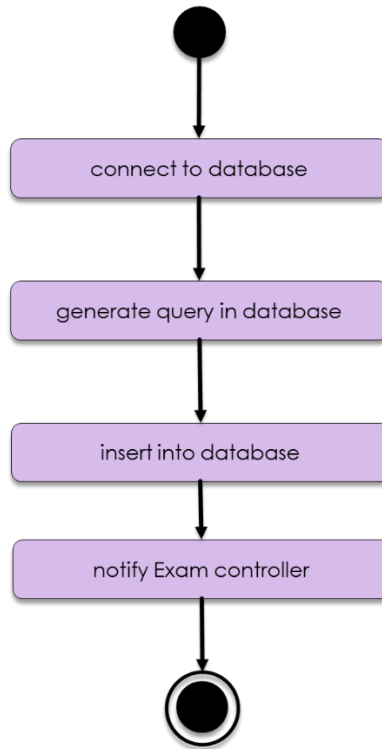


Figure 101: Processing flow of `insertRoomDatabase()`

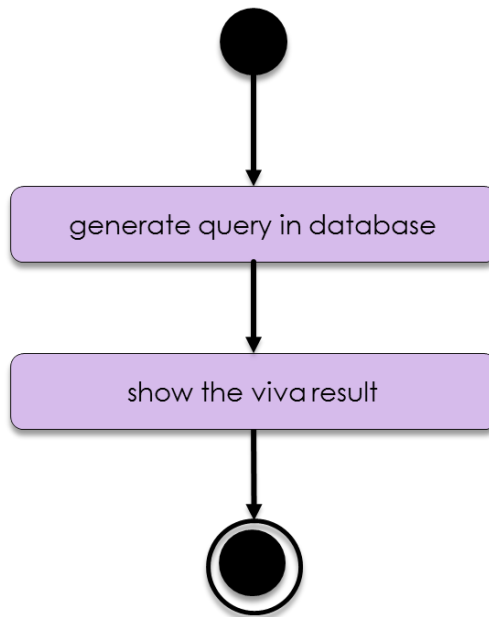


Figure 102: Processing flow of `viewVivaSchedule()`

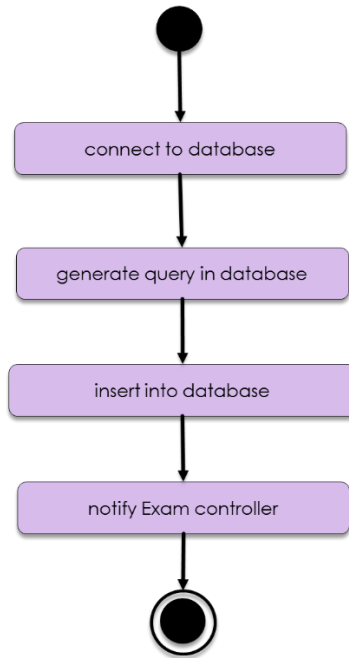


Figure 103: Processing flow of insertScheduleIntoDatabase()

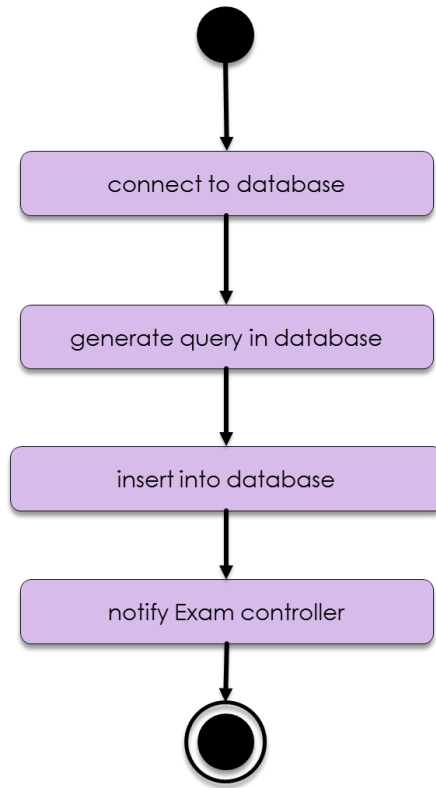


Figure 104: Processing flow of insertAdmissionInfointoDatabase()

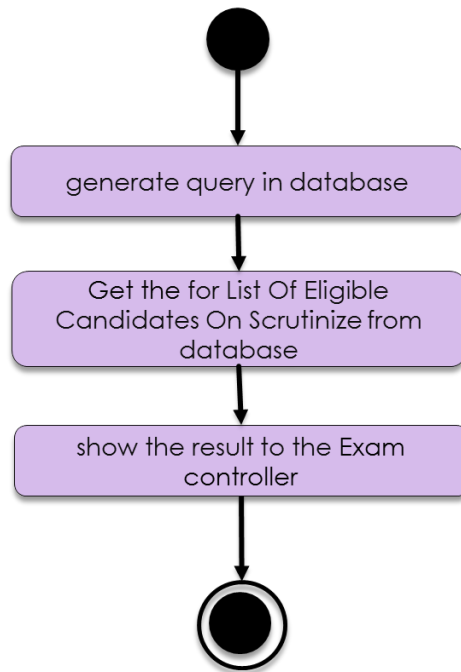


Figure 105: Processing flow of `viewListOfEligibleCandidatesOnScrutinize()`

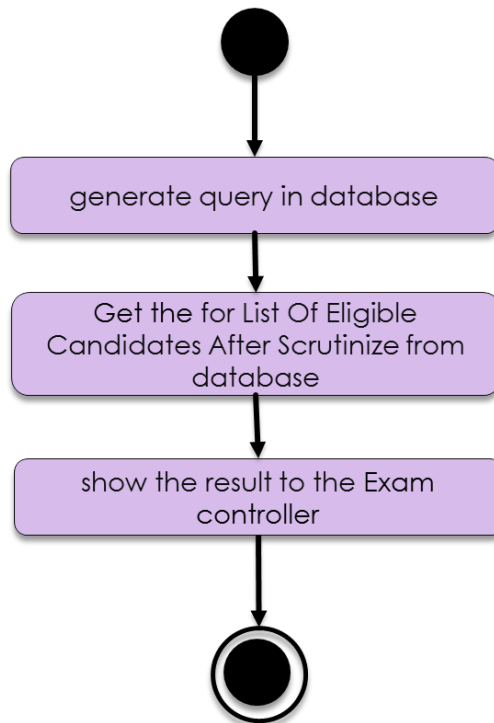


Figure 106: Processing flow of `viewListOfEligibleCandidatesAfterScrutinize()`

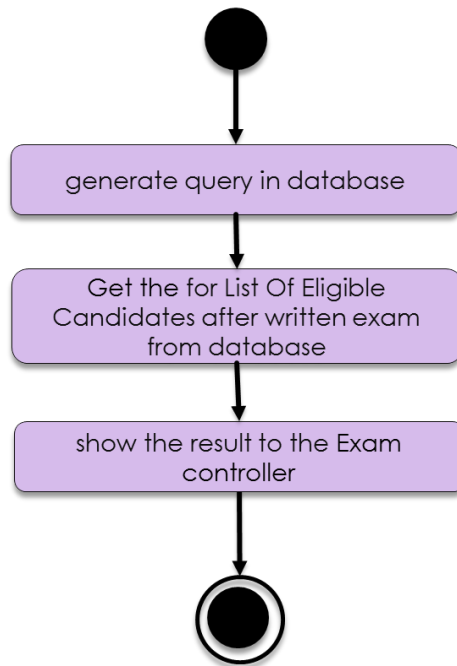


Figure 107: Processing flow of `viewListOfEligibleCandidatesAfterWrittenExam()`

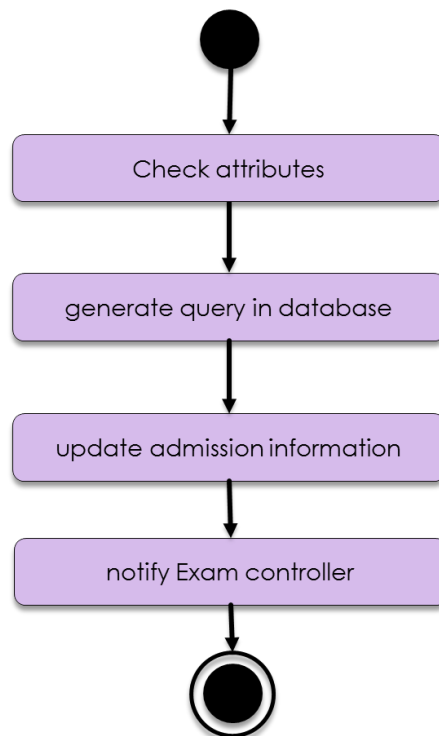


Figure 108: Processing flow of `updateAdmissionInfo`

Processing flow of the methods: Student Class

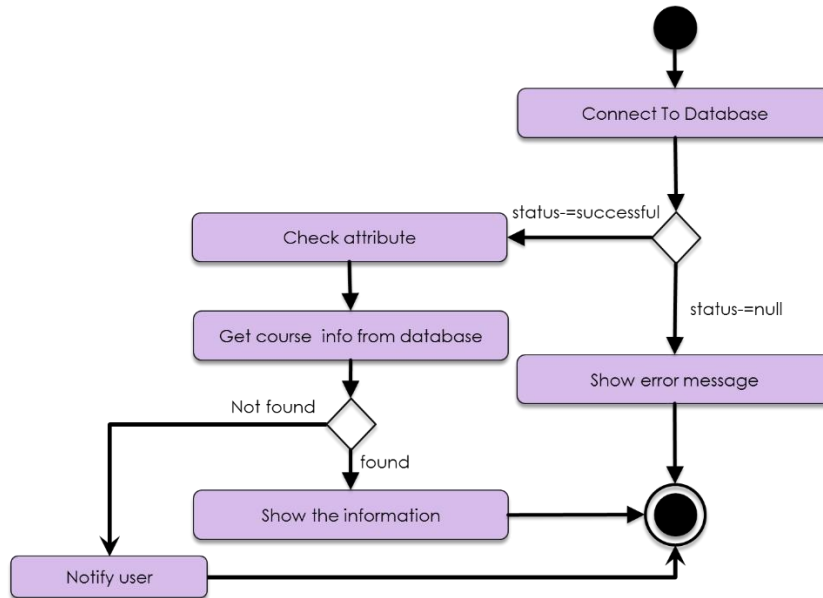


Figure 109: Processing flow of `viewCourseInformation(courseID, year)`

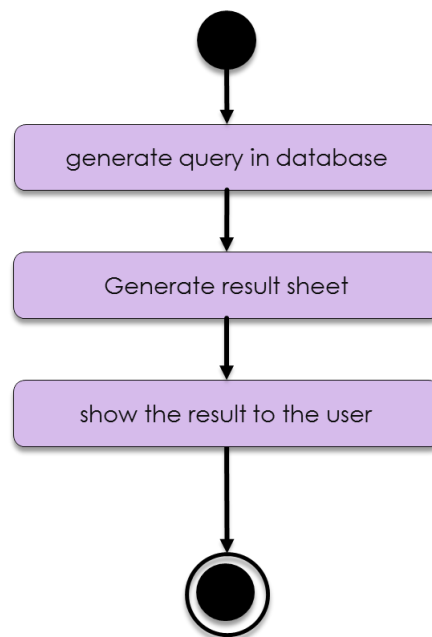


Figure 110: Processing flow of `viewResultsheet()`

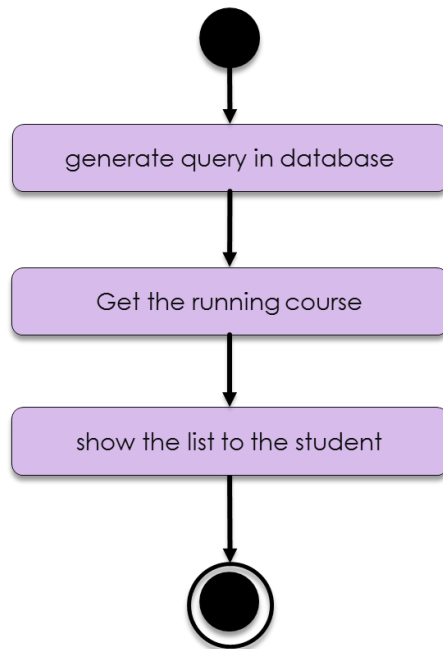


Figure 111: Processing flow of `viewRunningCourses()`

Processing flow of the methods: `Scrfinizer Class`

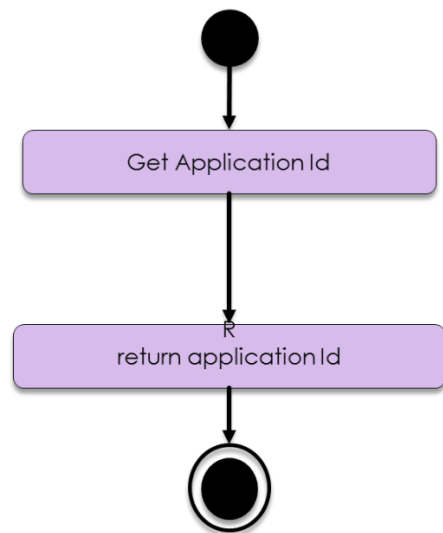


Figure 112: Processing flow of `getapplicationId()`

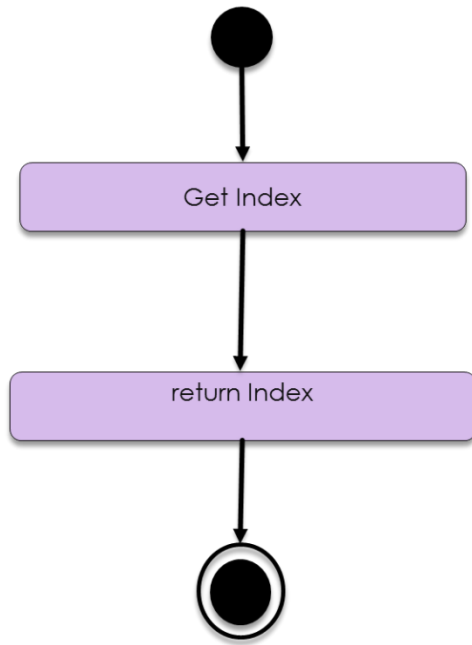
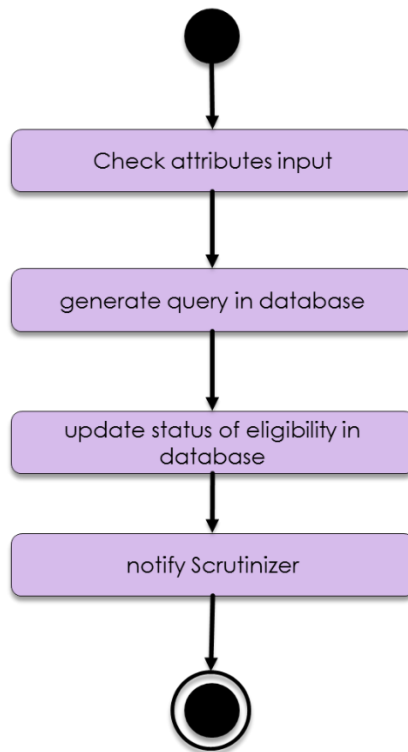


Figure 113: Processing flow of `getIndex()`



11/16/2017

Figure 114: Processing flow of `UpdateStatus(applicationId,index)`

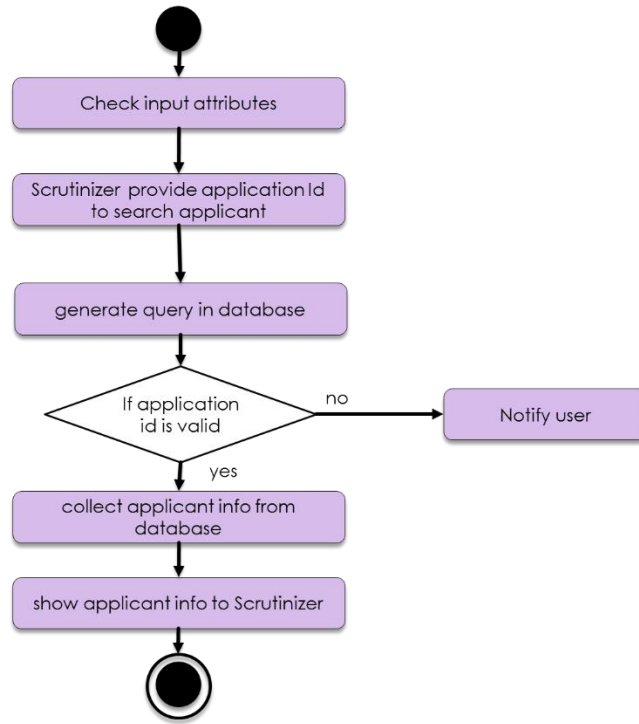


Figure 115: Processing flow of searchByApplicationID(applicationId)

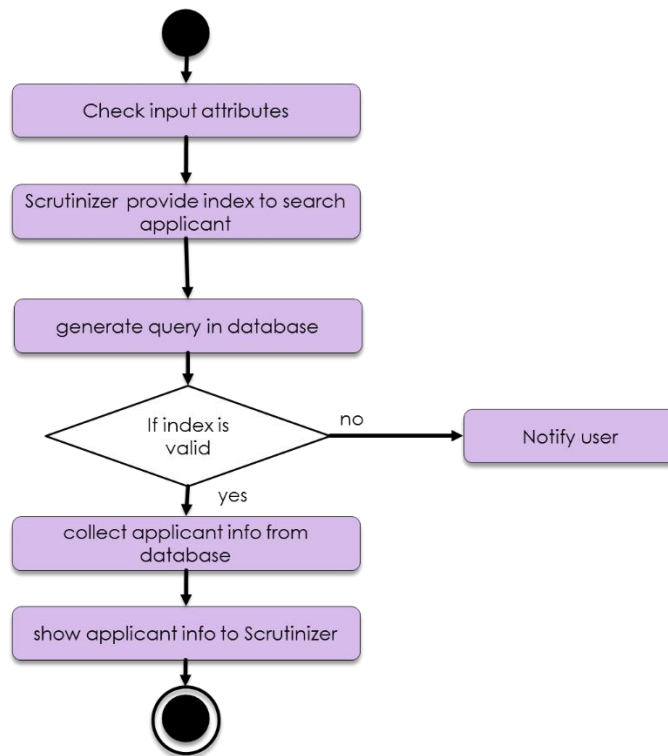


Figure 116: Processing flow of searchByIndex(index)

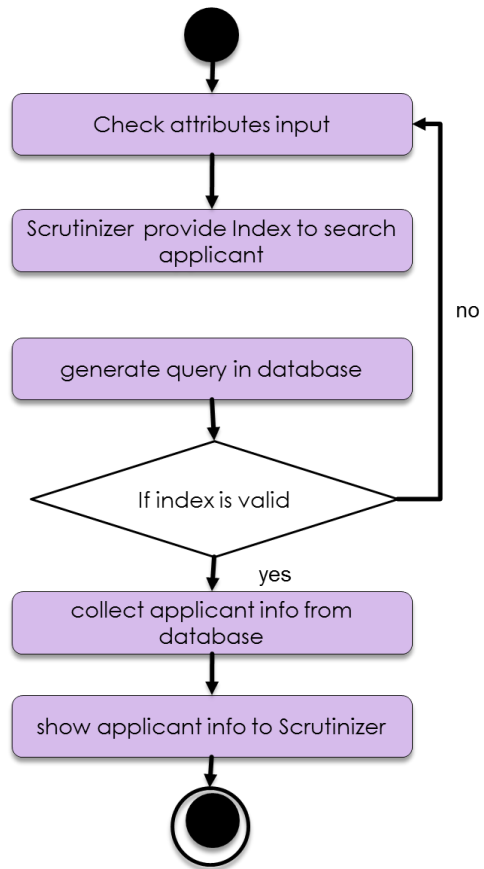


Figure 117: Processing flow of searchByIndex

2.3 f) State Diagram

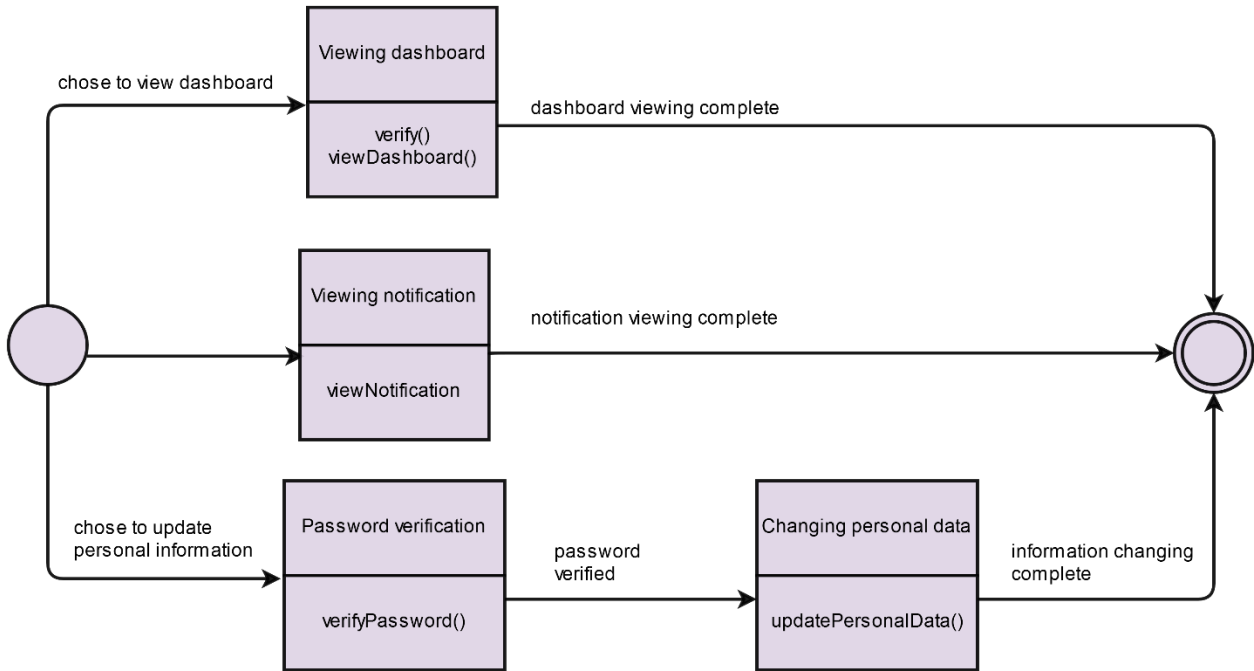


Figure 118: State diagram of User class

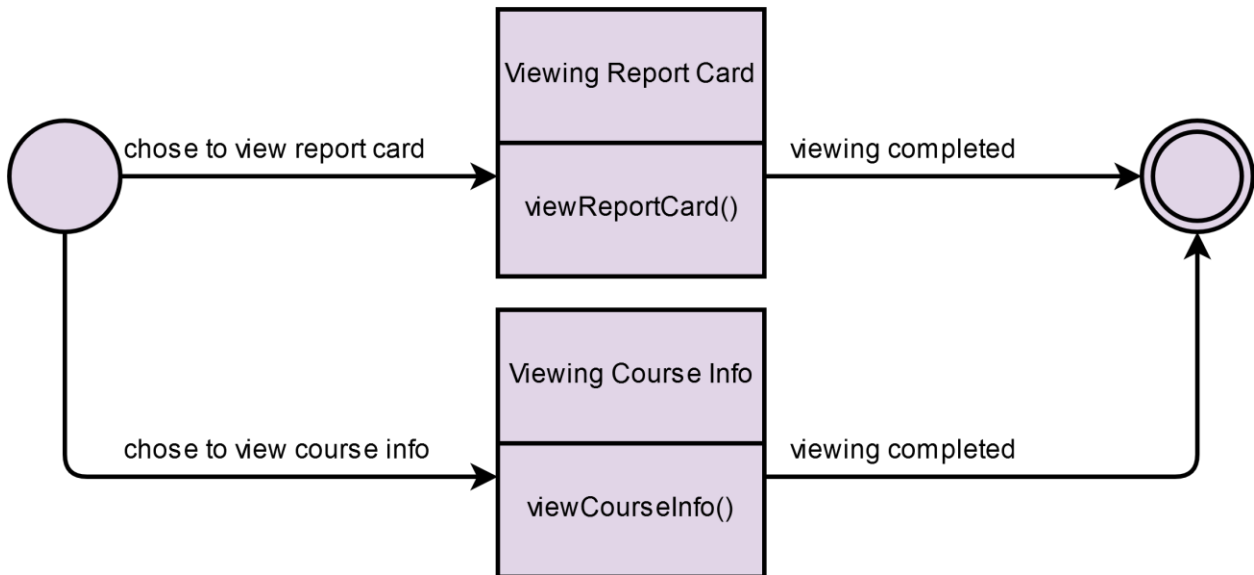


Figure 119: State diagram of Student class

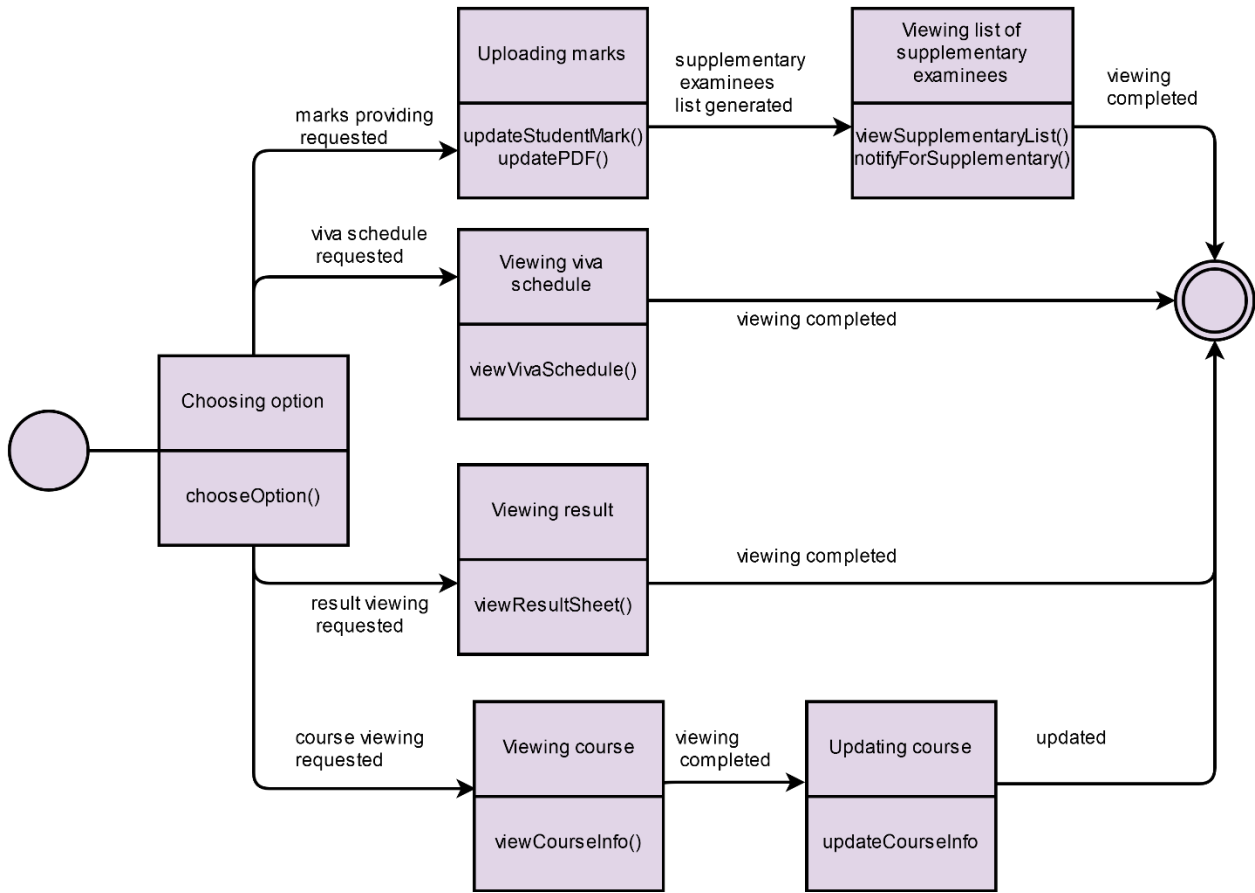


Figure 120: State diagram of Teacher class

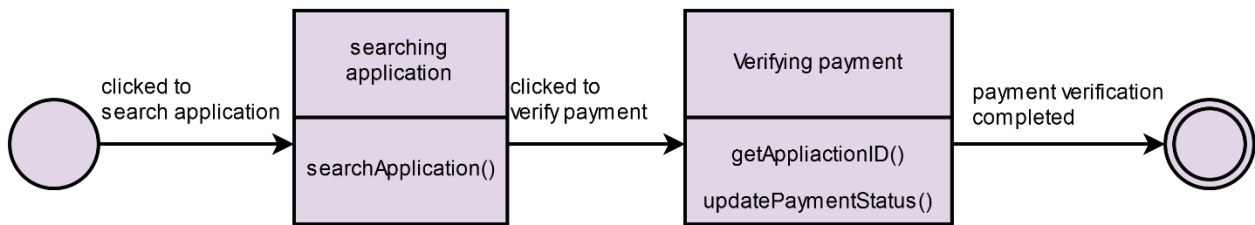


Figure 121: State diagram of Receptionist class

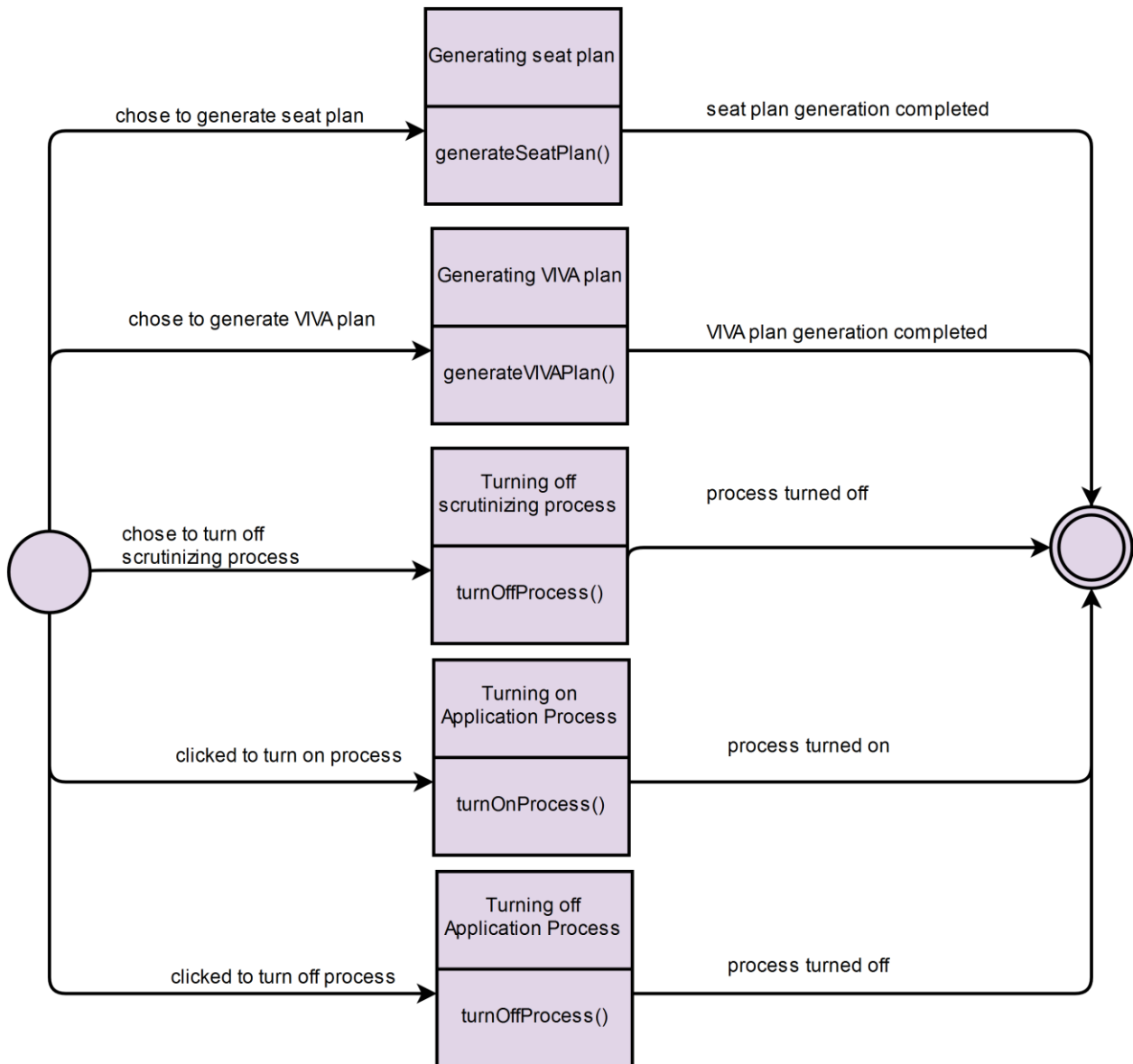


Figure 122: State diagram of Exam Controller class

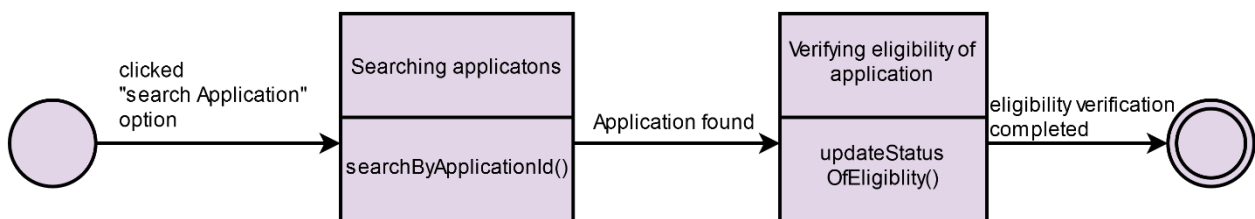


Figure 123: State diagram of Scrutinizer class

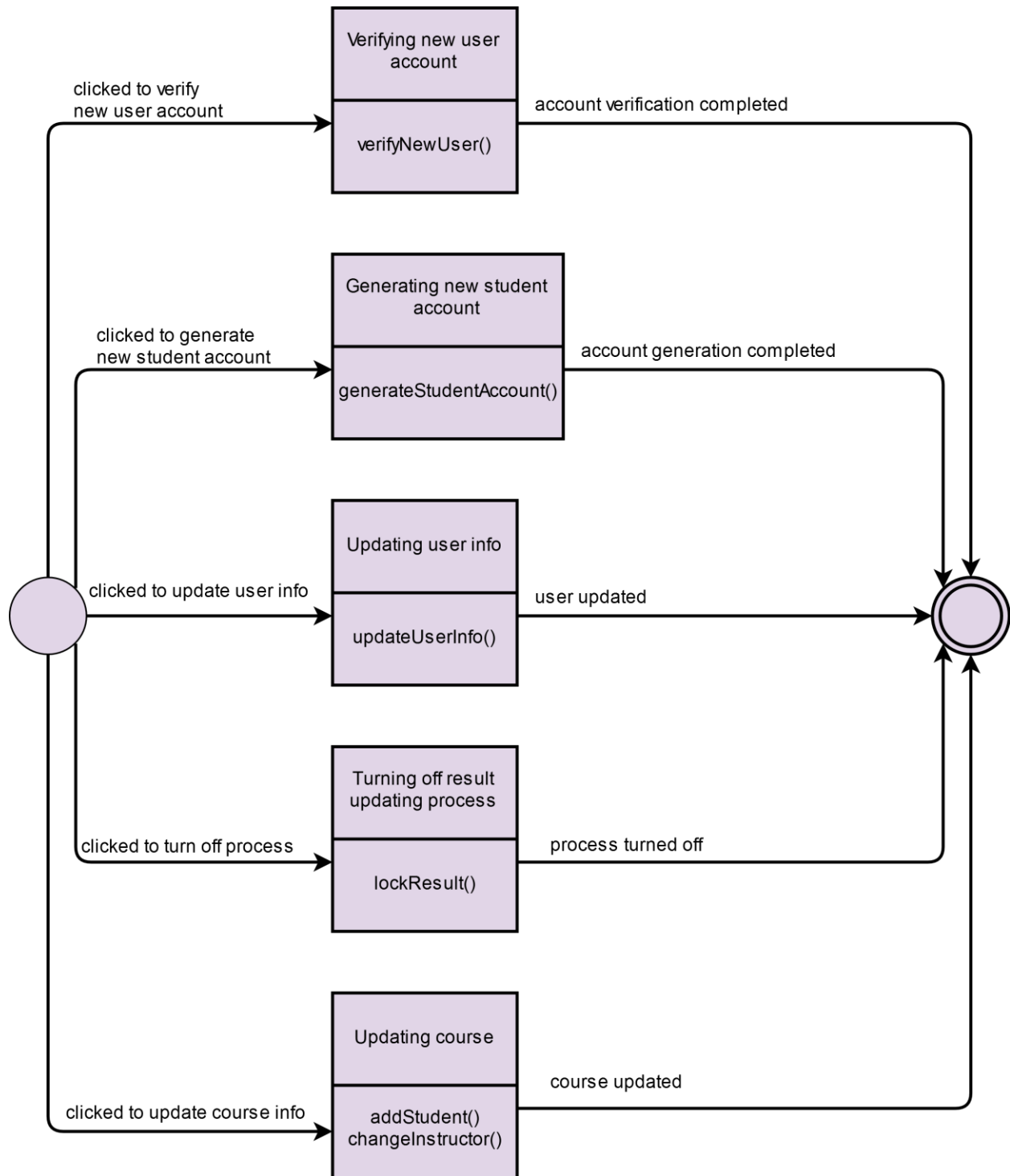


Figure 124: State diagram of Program Chairman class

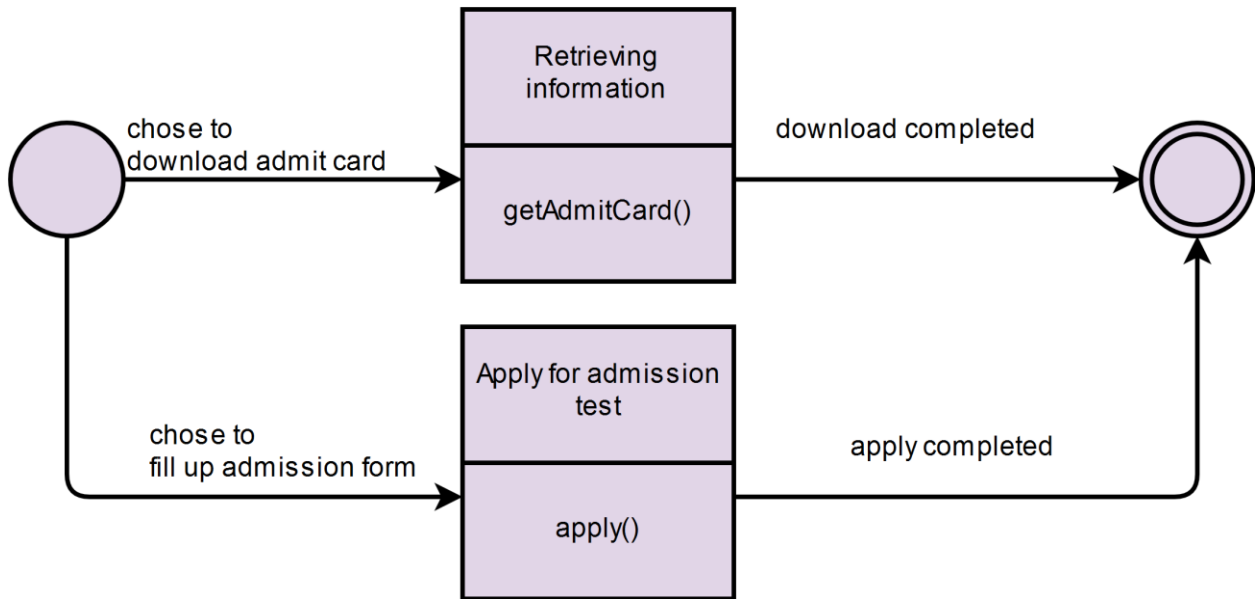


Figure 125: State diagram of Applicant class

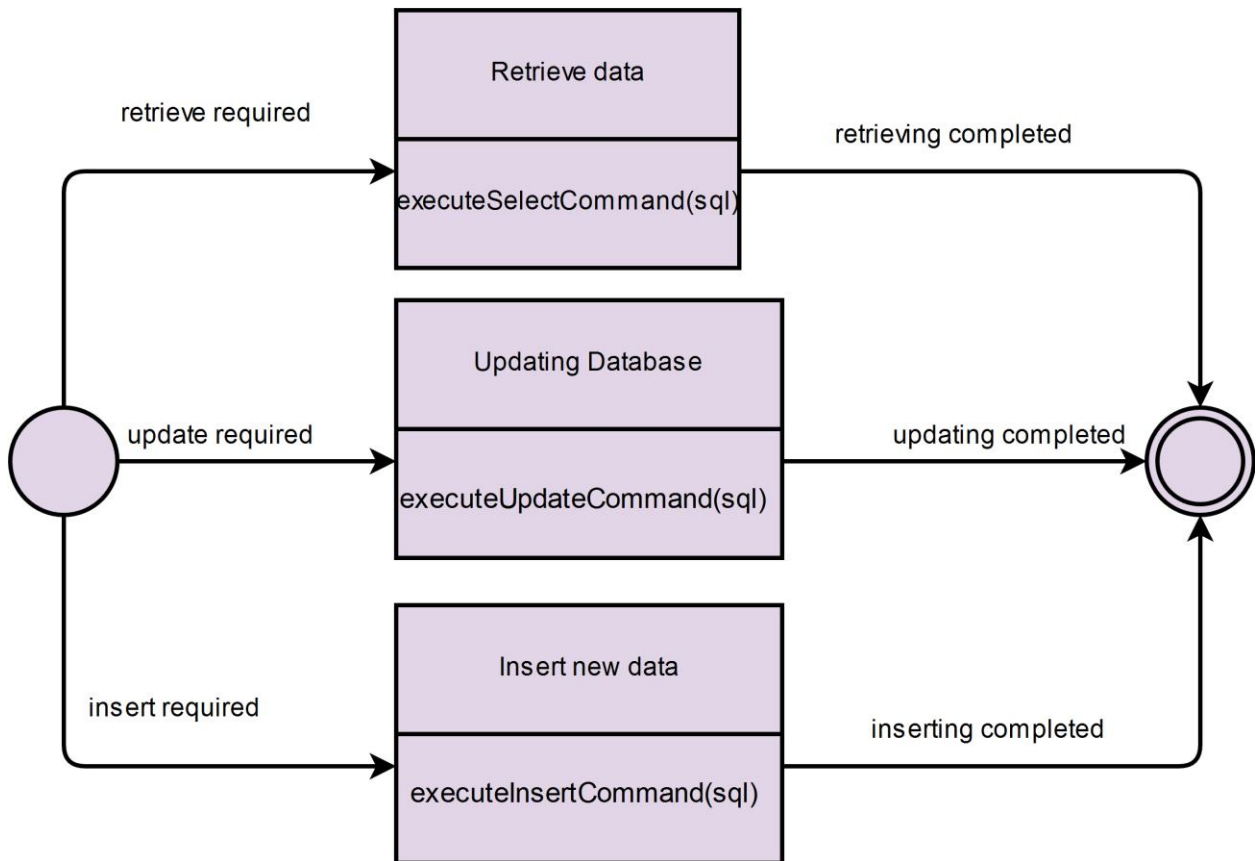


Figure 126: State diagram of Database Operator

2.4 Persistent Data

MySQL database: An open source relational database management system which is powerful and free that delivers a rich and reliable data store for lightweight Web Sites and desktop applications.

Class required to manage DB: Database Operator

2.5 Elaborated Deployment

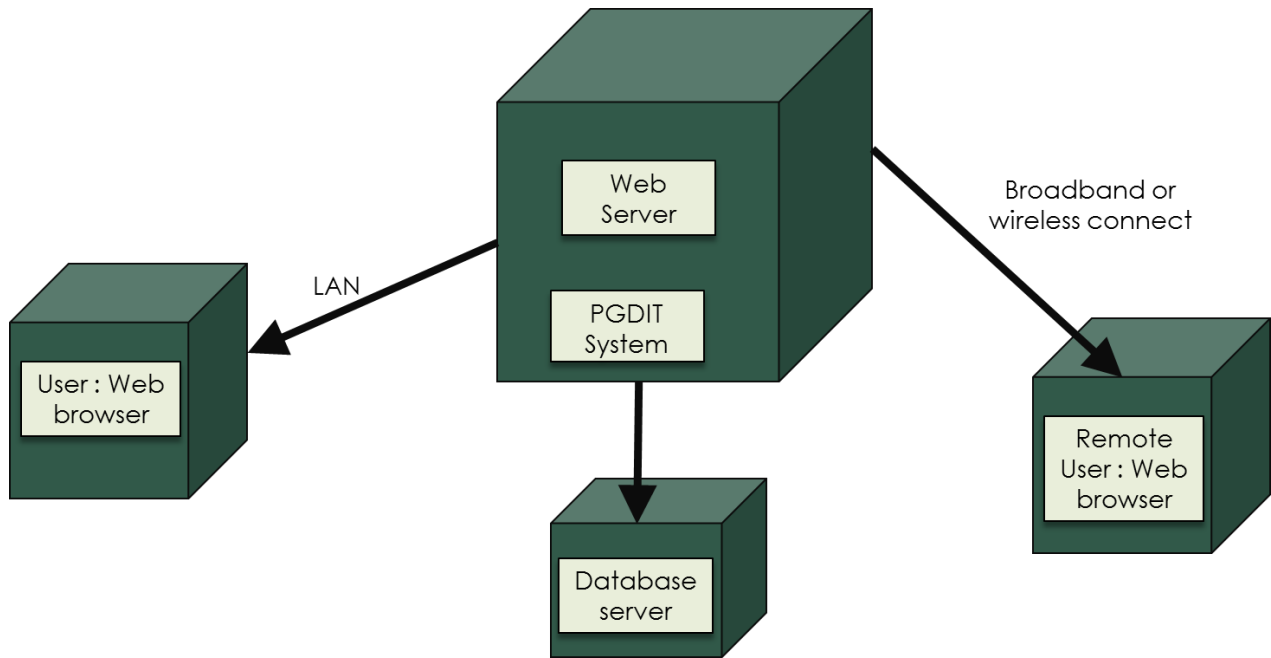


Figure 127: Elaborated deployment of PGDIT Automation System

CHAPTER 3: INTERFACE DESIGN

User Interface Design is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with the focus on the user's experience and interaction.

3.1 Interface Analysis

We divide interface analysis into the following part-

1. User analysis
2. Task analysis

USER ANALYSIS

There are two steps in this part-

1. Identify user
2. Know user

Identify User

From the requirement and specification document, we have found six types of user-

1. Teacher
2. Student
3. Exam Controller
4. Program Chairperson
5. Scrutinizing team
6. Receptionist
7. Applicant

Know User

We collect the following information about the users.

Teacher

- Age: 25-60
- Skills: Above Average
- Domain expert: Yes

- Application expert: No
- Office hour: Normal
- Frequency of use: Occasionally
- Consequence of a mistake: Low
- General computer experience: Yes

Student

- Age: 20-50
- Skills: Average
- Frequency of use: Occasionally
- Consequence of a mistake: Low
- General computer experience: Yes

Program Chairperson

- Age: 30-60
- Skills: Above Average
- Domain expert: Yes
- Application expert: No
- Office hour: Normal
- Frequency of use: Occasionally
- Consequence of a mistake: High
- General computer experience: Yes

Exam Controller

- Age: 30-60
- Skills: Above Average
- Domain expert: Yes
- Application expert: No
- Office hour: Normal
- Frequency of use: Occasionally
- Consequence of a mistake: High
- General computer experience: Yes

Scrutinizing Team

- Age: 30-60
- Skills: Above Average
- Application expert: No
- Frequency of use: Occasionally
- Consequence of a mistake: High
- General computer experience: Yes

Receptionist

- Age: 30-60
- Skills: Above Average
- Office hour: Normal
- Frequency of use: Occasionally
- Consequence of a mistake: Low
- General computer experience: Yes

Applicant

- Age: 20-60
- Frequency of use: Frequently
- Consequence of a mistake: Low
- General computer experience: No

TASK ANALYSIS

We identify and analysis the tasks of all the users separately.

Teacher

The Teacher perform the following tasks.

1. Registration

Goal: Register to the system as a teacher.

Pre-condition:

- Need to be a teacher of IIT

Sub-tasks:

- Apply for registration by fulfilling the form
- Approve by program chairperson

2. Manage Courses

Goal: Update courses details, marks and notify students.

Pre-condition:

- Logged in as a teacher.

Sub-tasks:

- Update courses details.
- Update marks.
- Notify students.

3. View Viva Schedule

Goal: View and download the viva notices.

Pre-condition:

- Logged in as a teacher.

Sub-tasks:

- View the notices.
- Download from the web.

Student

The student performs the following tasks.

1. View Mark, CGPA and Result

Goal: View the mark sheet, CGPA and result of students.

Pre-condition:

- Logged in as a student.

Sub-tasks:

- View mark sheet, CGPA and result in the dashboard.

Program Chairperson

The program chairperson performs the following tasks.

1. Manage Users

Goal: Create, update user accounts and approve/deny new accounts request.

Pre-condition:

- Logged in as program chairperson.
- Request from teachers.

Sub-tasks:

- Create user account.
- Update user account.
- Approve/deny teacher's account request.

2. Manage Courses

Goal: Add/Change courses and add students to courses.

Pre-condition:

- Logged in as program chairperson.

Sub-tasks:

- Add/Change courses.
- Add students to courses.

Exam Controller

The exam controller performs the following tasks.

1. Initiate the Admission Process

Goal: Initiate the admission process by uploading the admission circular.

Pre-condition:

- Logged in as exam controller.

Sub-tasks:

- Upload the circular on the web.

2. Manage Schedule

Goal: Add viva schedule and update exam room.

Pre-condition:

- Logged in as exam controller.
- Scrutinizing team filters the applications.

Sub-tasks:

- Add viva schedule.
- Update exam rooms.

Scrutinizing Team

The scrutinizing team perform the following tasks.

1. Download Applications to Filter

Goal: Download all the applications to filter them manually.

Pre-condition:

- Logged in as scrutinizing team
- The applicant's payment verified by the receptionist.

Sub-tasks:

- Download the applications.

Receptionist

The receptionists perform the following tasks.

1. Update Payment Status

Goal: Update the payment status of applicants after paying the fees.

Pre-condition:

- Logged in as receptionist.
- The applicant paid the fees manually.

Sub-tasks:

- Update payment status.

2. Search Applicants

Goal: Search applicants by name or phone number.

Pre-condition:

- Logged in as receptionist.

Sub-tasks:

- Search by name or phone number.

Applicant

The applicants can perform the following tasks.

1. Apply for Admission

Goal: apply for admitting in PGDIT.

Pre-condition:

- The exam controller uploads the circular.

Sub-tasks:

- Visit the website.
- Apply by fulfilling the form and submit.

3.2 Interface Design Steps

We follow the following steps to design the Library Circulation System (LCS) user interface.

- i. Define interface objects and actions
- ii. Define events that will cause the state of the user interface to change
- iii. Depict each interface state as it look to end user

DEFINE INTERFACE OBJECTS AND ACTIONS

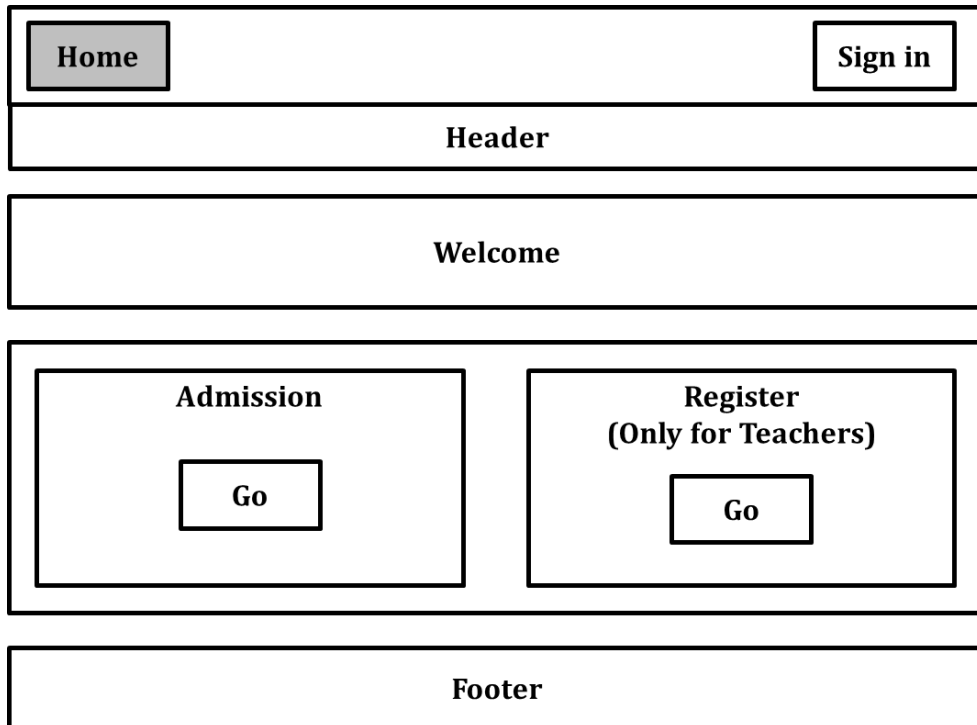


Figure 128: External (Home)

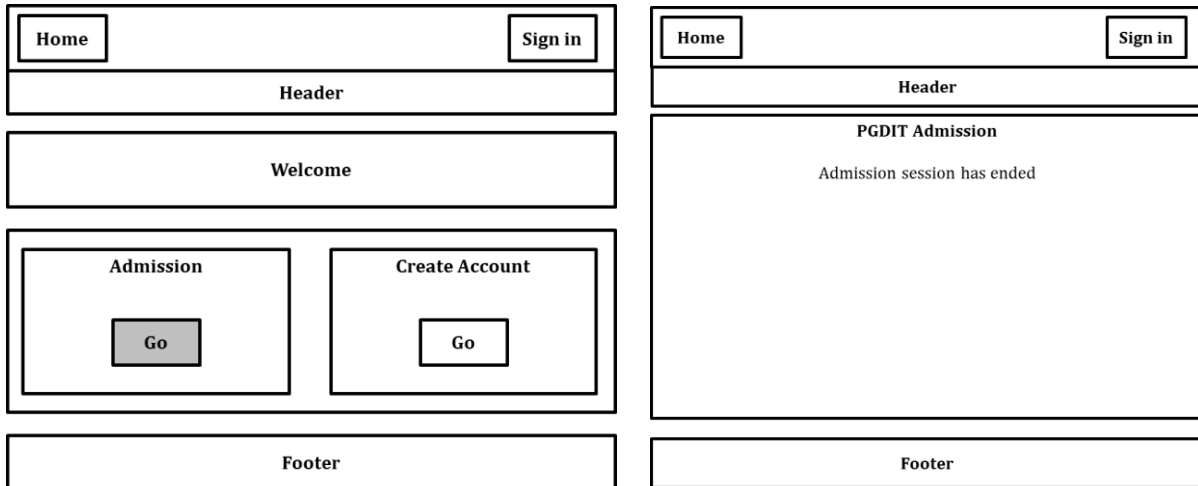


Figure 129: External (Admission – Session ended)

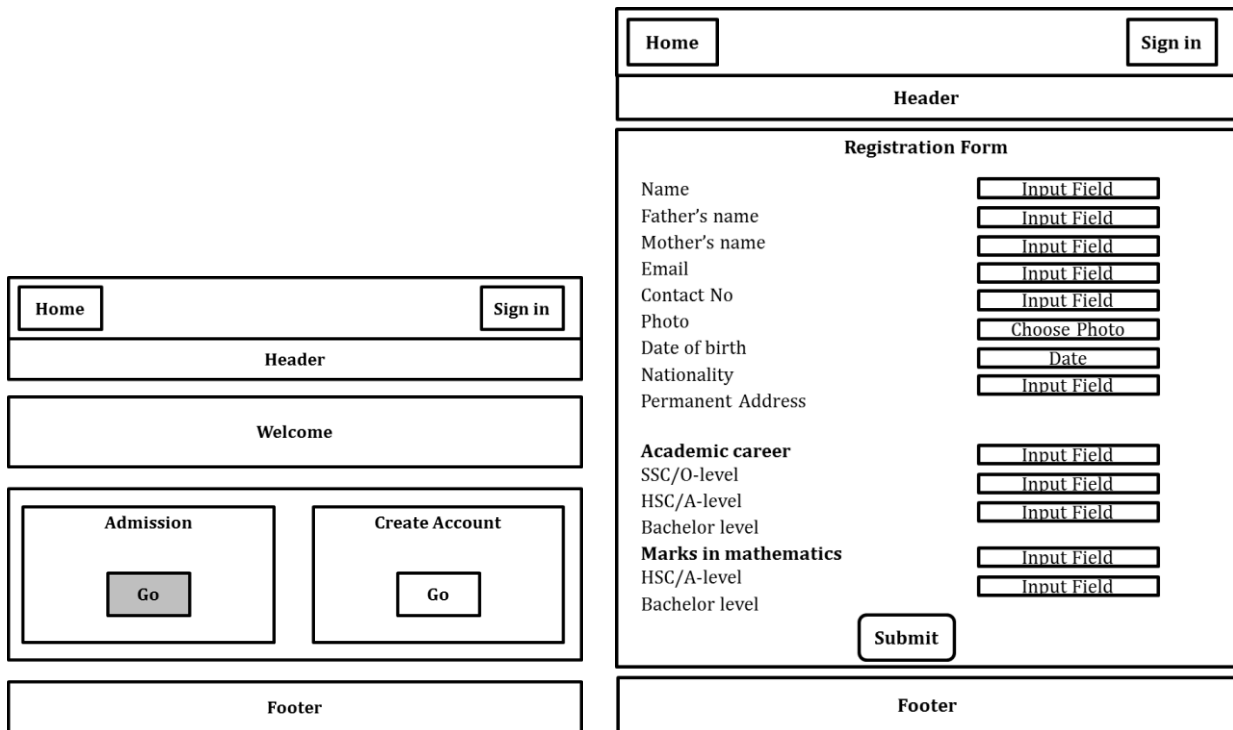


Figure 130: External (Admission - Session going admission form)

Home	Sign in
Header	
Registration Form	
Name	<input type="text"/>
Father's name	<input type="text"/>
Mother's name	<input type="text"/>
Email	<input type="text"/>
Contact No	<input type="text"/>
Photo	<input type="button" value="Choose Photo"/>
Date of birth	<input type="text" value="Date"/>
Nationality	<input type="text"/>
Permanent Address	<input type="text"/>
Academic career	<input type="text"/>
SSC/O-level	<input type="text"/>
HSC/A-level	<input type="text"/>
Bachelor level	<input type="text"/>
Marks in mathematics	<input type="text"/>
HSC/A-level	<input type="text"/>
Bachelor level	<input type="text"/>
<input type="button" value="Submit"/>	
Footer	

Home	Sign in
Header	
You have successfully applied for PGDIT Admission	
Application ID	Text
Name	Text
Father's name	Text
Mother's name	Text
Email	Text
Contact No	Text
Photo	Photo
Date of birth	Date
Nationality	Text
Permanent Address	Text
Academic career	
SSC/O-level	Text
HSC/A-level	Text
Bachelor level	Text
Marks in mathematics	
HSC/A-level	Text
Bachelor level	Text
Footer	

Figure 131: External (Admission - Session going - Admission form - Submit)

Home	Sign in
Header	
Welcome	
Admission <input type="button" value="Go"/>	Create Account <input type="button" value="Go"/>
Footer	

Home	Sign in
Header	
Admit Card	
Application ID	<input type="text"/>
Date of Birth	<input type="text"/>
<input type="button" value="Download Admit card"/>	
Footer	

Figure 132: External (Admission - Session going - Admit card - Available)

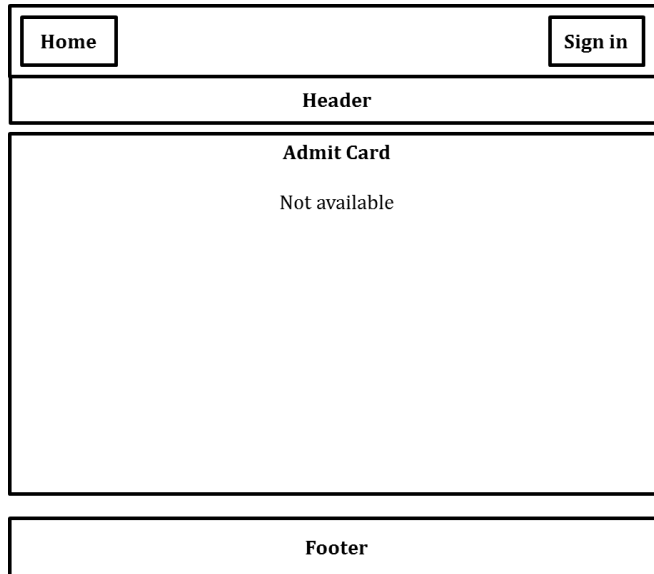


Figure 133: External (Admission – Session going – Admit card – Not Available)

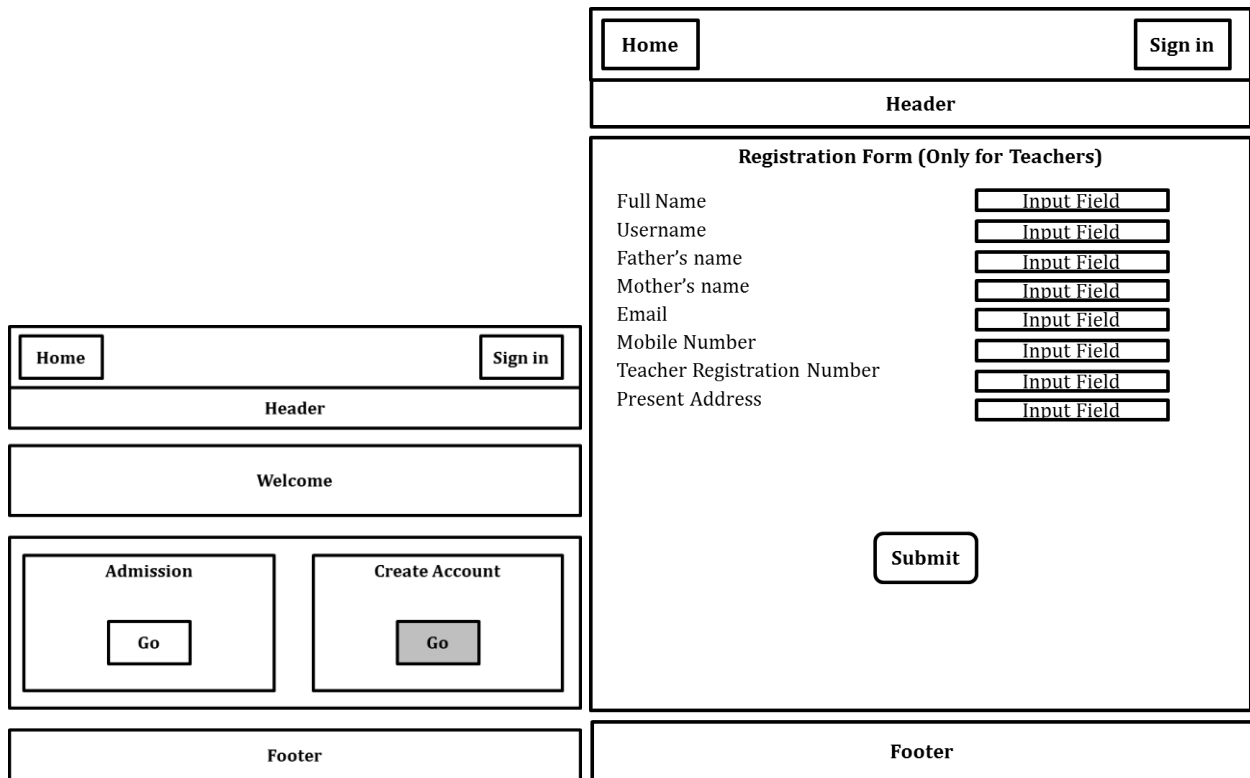


Figure 134: External (Create Account)

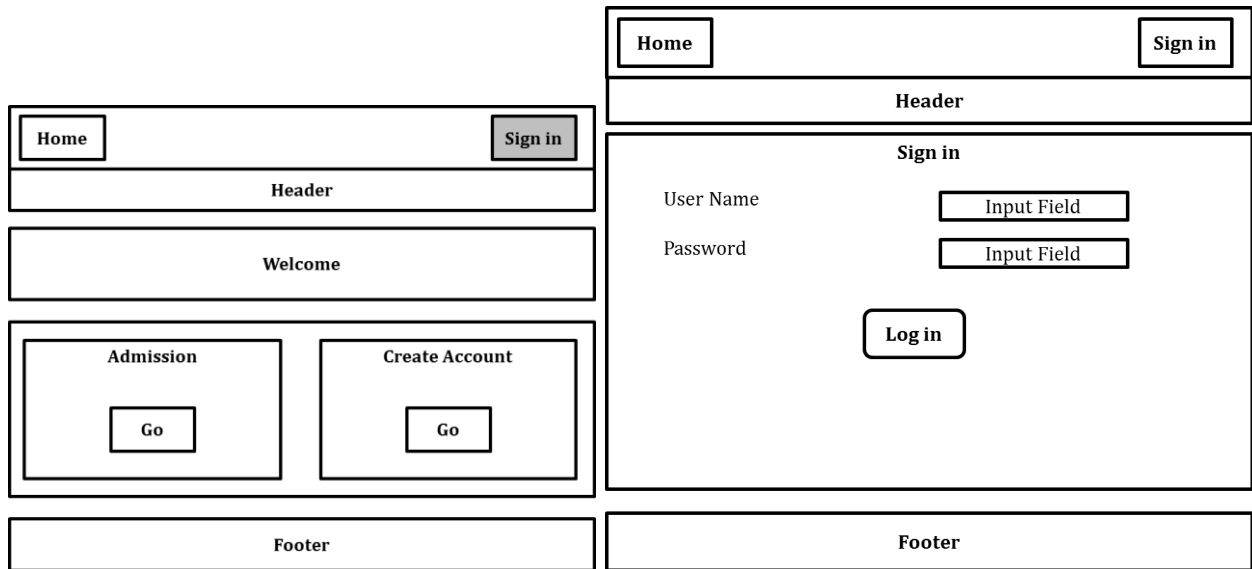


Figure 135: External (Sing in)

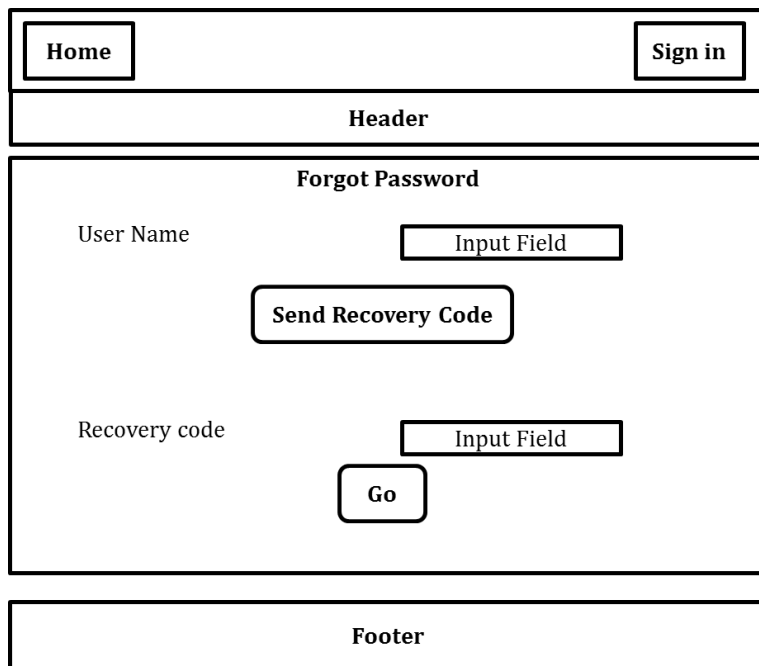


Figure 136: External (Sing in-Forgot password)

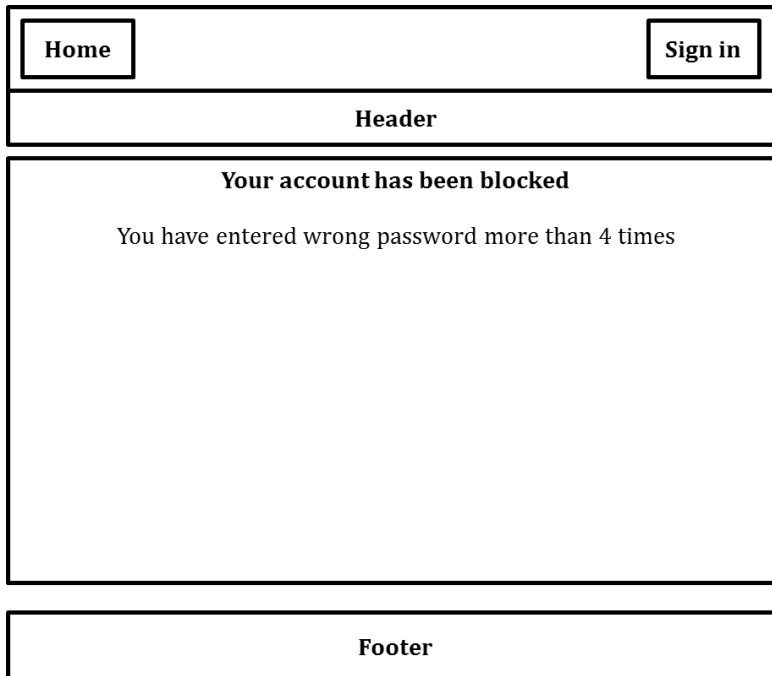


Figure 137: External (Sing in - Blocked)

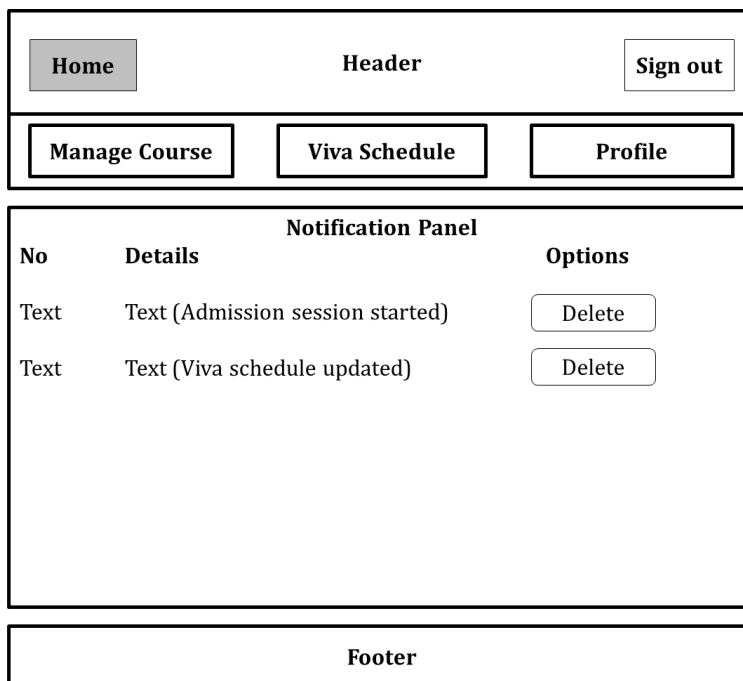


Figure 138: Internal (Teacher: Home)

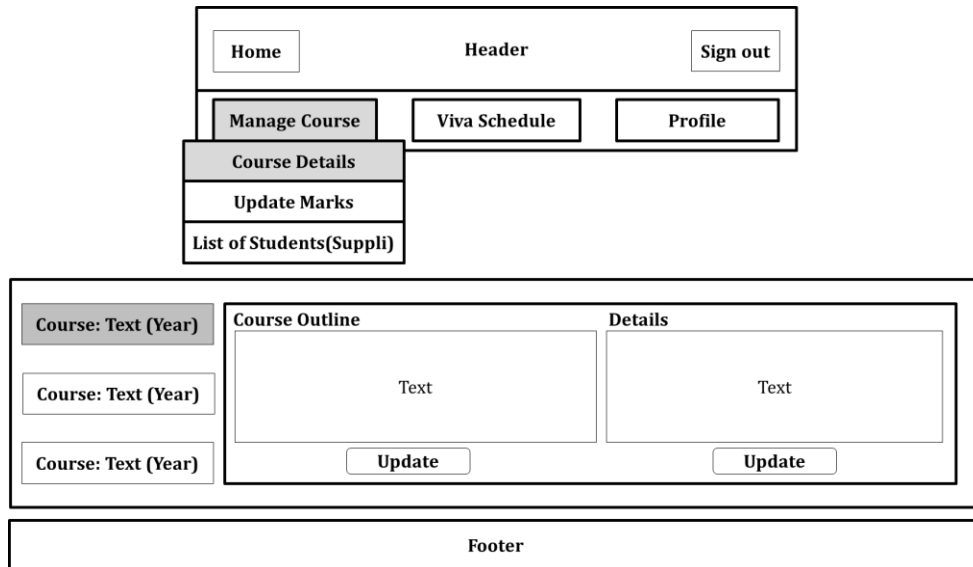


Figure 139: Internal (Teacher: Manage Course- Course Details)

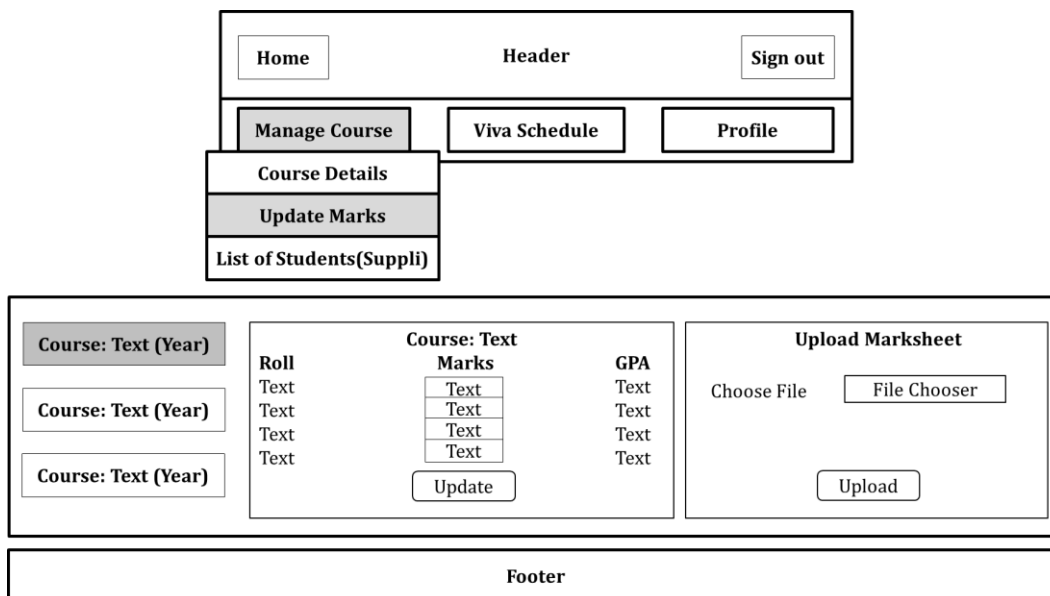


Figure 140: Internal (Teacher: Manage Course- Update Marks)

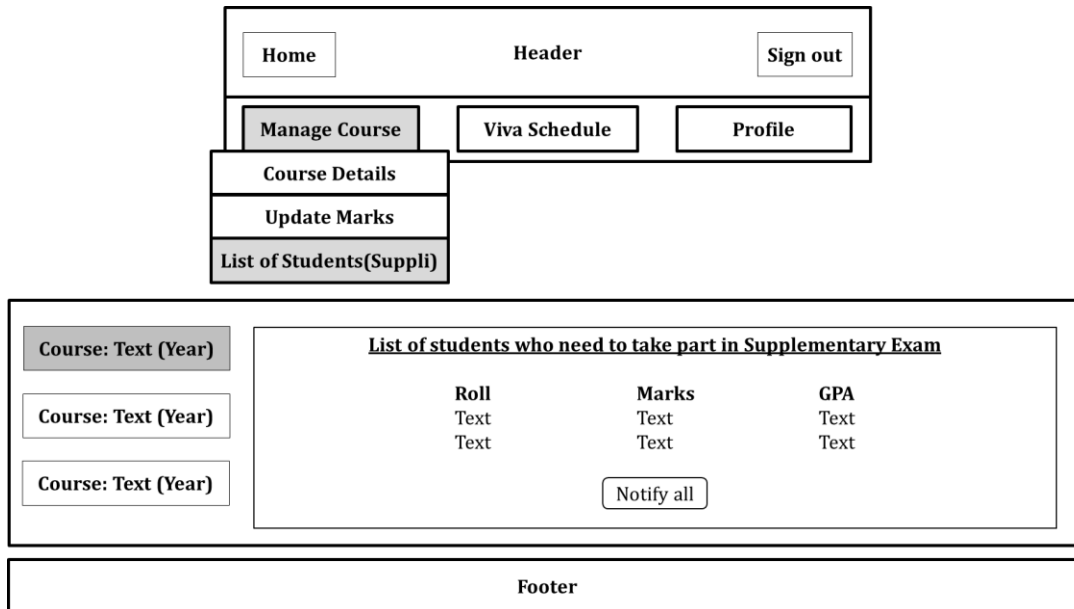


Figure 141: Internal (Teacher: Manage Course- List of Students (Suppli))

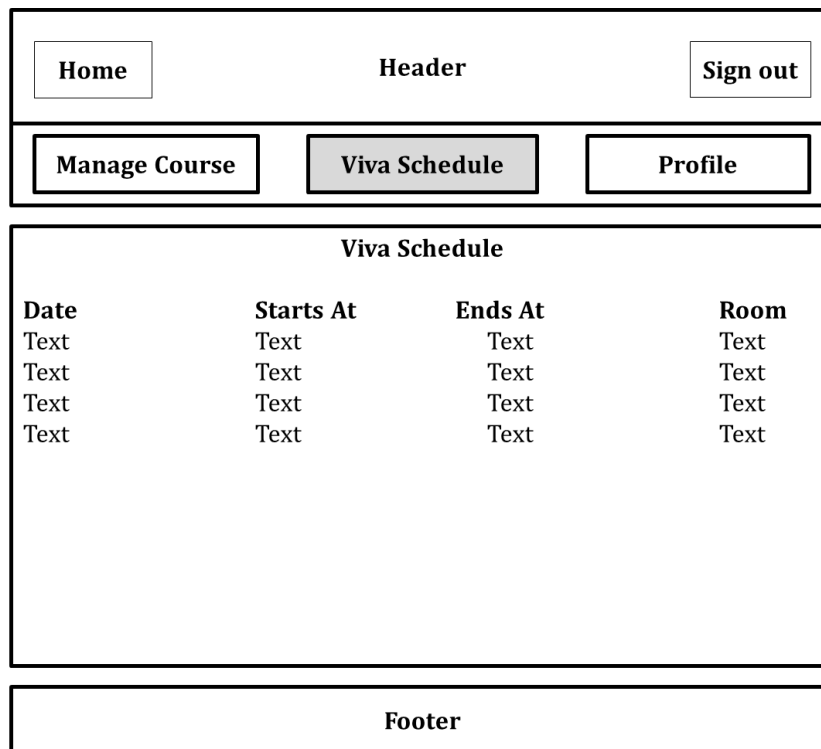


Figure 142: Internal (Teacher: Viva Schedule)

Home	Header	Sign out
Manage Course	Viva schedule	Profile
<div style="border: 1px solid black; width: 60px; height: 60px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">Photo</div>		<div style="border: 1px solid black; width: 100px; height: 25px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">Update</div>
Full Name	Text	<div style="border: 1px solid black; width: 100px; height: 25px; display: flex; align-items: center; justify-content: center;">Update</div>
Father's name	Text	<div style="border: 1px solid black; width: 100px; height: 25px; display: flex; align-items: center; justify-content: center;">Update</div>
Mother's name	Text	<div style="border: 1px solid black; width: 100px; height: 25px; display: flex; align-items: center; justify-content: center;">Update</div>
Email	Text	<div style="border: 1px solid black; width: 100px; height: 25px; display: flex; align-items: center; justify-content: center;">Update</div>
Mobile Number	Text	<div style="border: 1px solid black; width: 100px; height: 25px; display: flex; align-items: center; justify-content: center;">Update</div>
Present Address	Text	<div style="border: 1px solid black; width: 100px; height: 25px; display: flex; align-items: center; justify-content: center;">Update</div>
Footer		

Figure 143: Internal (Member: Profile)

Home	Header	Sign out
View Course	Result	Profile
Notification Panel		
No	Details	Options
Text	Text (Supplementary exam)	<div style="border: 1px solid black; width: 80px; height: 25px; display: flex; align-items: center; justify-content: center;">Delete</div>
Footer		

Figure 144: Internal (Student: Home)

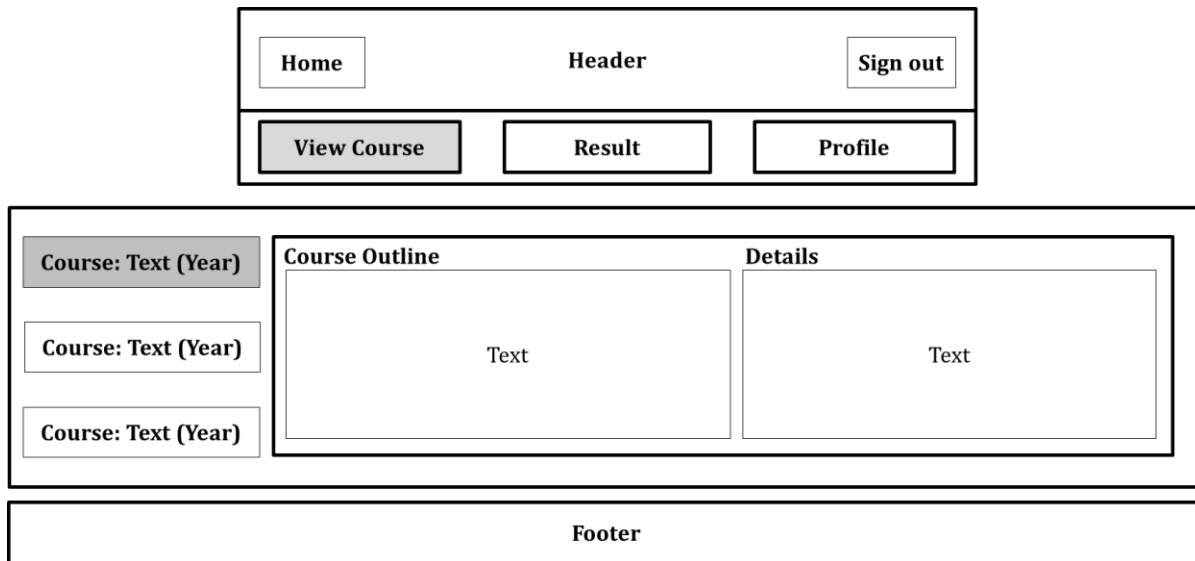


Figure 145: Internal (Student: View Course)

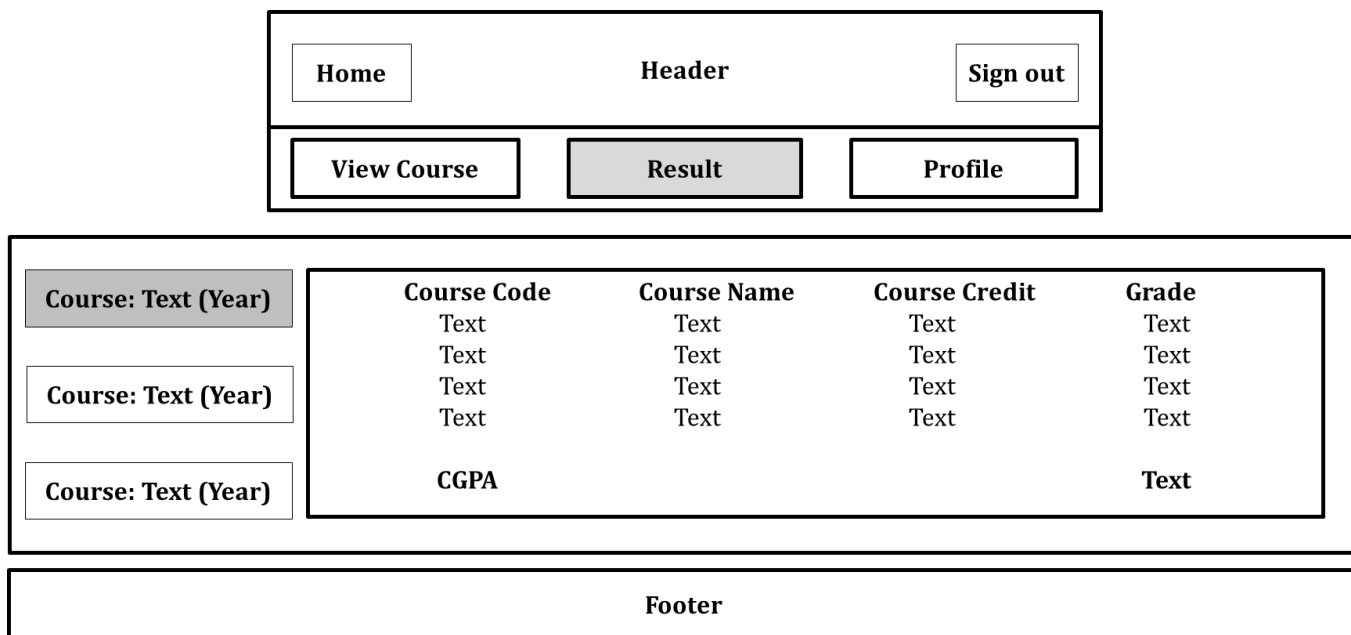
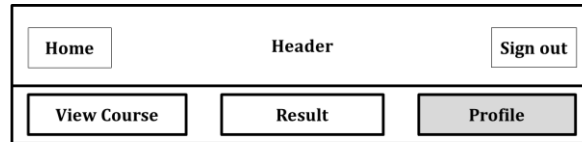


Figure 146: Internal (Student: View Course)



Same as Teacher: Profile

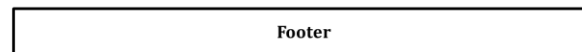


Figure 147: Internal (Student: Profile)

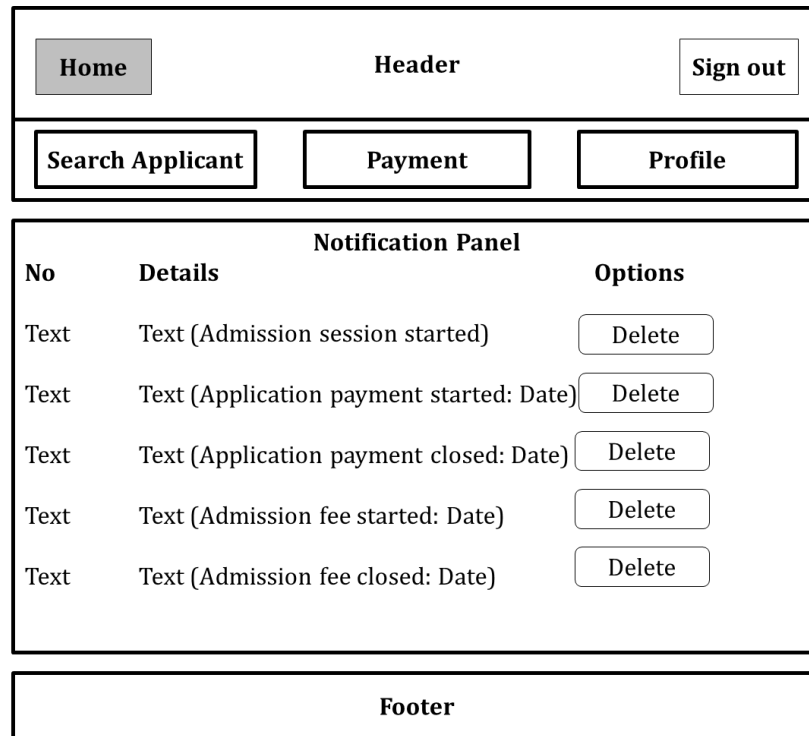


Figure 148: Internal (Receptionist: Home)

Home	Header	Sign out
Search Applicant	Payment	Profile
Search by name		
Search by mobile no		

<p>Applicant name:</p> <input style="width: 80%;" type="text" value="Input Field"/> <input type="button" value="Search"/>	<table style="width: 100%; border: 1px solid black;"> <tr> <th colspan="3" style="text-align: center;"><u>Applicant Information</u></th> </tr> <tr> <td style="width: 50%;">Application ID</td> <td style="width: 30%;">Text</td> <td rowspan="8" style="width: 20%; text-align: center; vertical-align: middle;">Photo</td> </tr> <tr> <td>Name of Applicant</td> <td>Text</td> </tr> <tr> <td>Father's name</td> <td>Text</td> </tr> <tr> <td>Mother's name</td> <td>Text</td> </tr> <tr> <td>Email</td> <td>Text</td> </tr> <tr> <td>Mobile Number</td> <td>Text</td> </tr> <tr> <td>Present Address</td> <td>Text</td> </tr> <tr> <td>Date of Birth</td> <td>Text</td> </tr> </table>	<u>Applicant Information</u>			Application ID	Text	Photo	Name of Applicant	Text	Father's name	Text	Mother's name	Text	Email	Text	Mobile Number	Text	Present Address	Text	Date of Birth	Text
<u>Applicant Information</u>																					
Application ID	Text	Photo																			
Name of Applicant	Text																				
Father's name	Text																				
Mother's name	Text																				
Email	Text																				
Mobile Number	Text																				
Present Address	Text																				
Date of Birth	Text																				

Footer

Figure 149: Internal (Receptionist: Search Applicant-By name)

Home	Header	Sign out
Search Applicant	Payment	Profile
Search by name		
Search by mobile no		

<p>Applicant's Mobile No:</p> <input style="width: 80%;" type="text" value="Input Field"/> <input type="button" value="Search"/>	<table style="width: 100%; border: 1px solid black;"> <tr> <th colspan="3" style="text-align: center;"><u>Applicant Information</u></th> </tr> <tr> <td style="width: 50%;">Application ID</td> <td style="width: 30%;">Text</td> <td rowspan="8" style="width: 20%; text-align: center; vertical-align: middle;">Photo</td> </tr> <tr> <td>Name of Applicant</td> <td>Text</td> </tr> <tr> <td>Father's name</td> <td>Text</td> </tr> <tr> <td>Mother's name</td> <td>Text</td> </tr> <tr> <td>Email</td> <td>Text</td> </tr> <tr> <td>Mobile Number</td> <td>Text</td> </tr> <tr> <td>Present Address</td> <td>Text</td> </tr> <tr> <td>Date of Birth</td> <td>Text</td> </tr> </table>	<u>Applicant Information</u>			Application ID	Text	Photo	Name of Applicant	Text	Father's name	Text	Mother's name	Text	Email	Text	Mobile Number	Text	Present Address	Text	Date of Birth	Text
<u>Applicant Information</u>																					
Application ID	Text	Photo																			
Name of Applicant	Text																				
Father's name	Text																				
Mother's name	Text																				
Email	Text																				
Mobile Number	Text																				
Present Address	Text																				
Date of Birth	Text																				

Footer

Figure 150: Internal (Receptionist: Search Applicant-By mobile no)

Home
Header
Sign out

Search Applicant
Payment
Profile

Application payment
Admission fees

Application ID:

Input Field

Paid

List of Students who have paid Application fees

No	Application ID	Name	Contact No
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text

Footer

Home
Header
Sign out

Search Applicant
Payment
Profile

Application payment
Admission fees

Payment Session is closed

Footer

Figure 151: Internal (Receptionist: Payment-Application payment)

Home
Header
Sign out

Search Applicant
Payment
Profile

Application payment

Admission fees

Application ID:

List of Students who have paid Admission fees

No	Application ID	Name	Contact No
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text

Footer

Home
Header
Sign out

Search Applicant
Payment
Profile

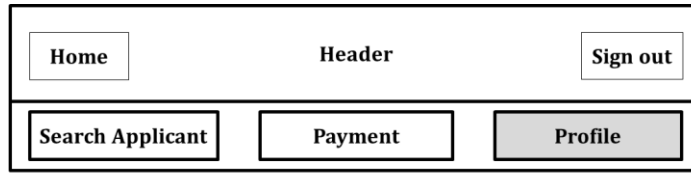
Application payment

Admission fees

Payment Session is closed

Footer

Figure 152: Internal (Receptionist: Payment-Admission fees)



Same as Teacher: Profile

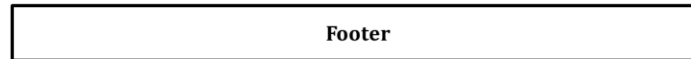


Figure 153: Internal (Receptionist: Profile)

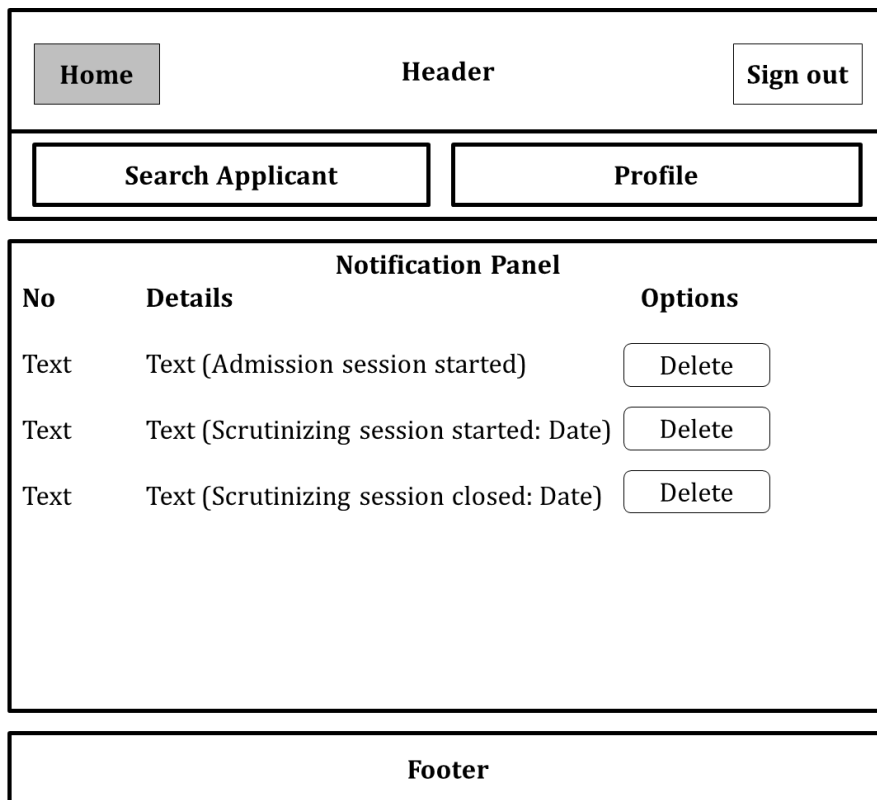


Figure 154: Internal (Scrutinizer: Home)

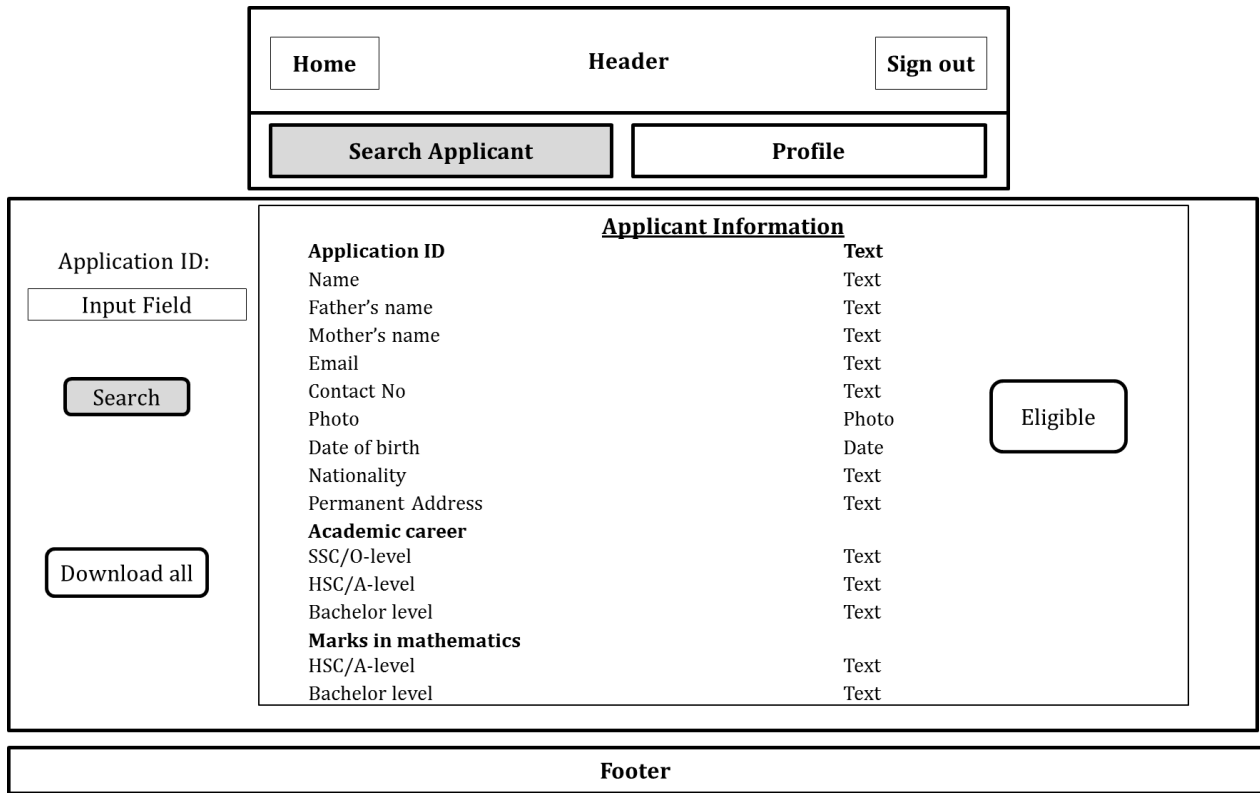
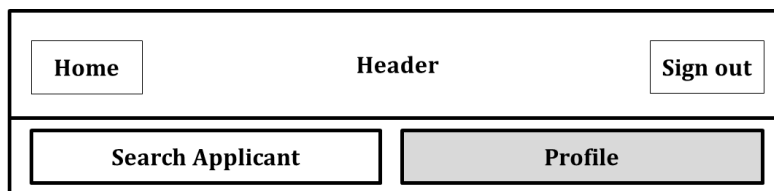


Figure 155: Internal (Scrutinizer: Search Applicant)



Same as Teacher: Profile

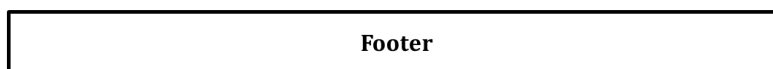


Figure 156: Internal (Scrutinizer: Profile)

Home	Header	Sign out
Manage Course	Manage User	Profile
Notification Panel		
No	Details	Options
Text	Text (Admission session started, Sent by Exam controller)	Delete
Footer		

Figure 157: Internal (Program chairperson: Home)

Home	Header	Sign out																																																		
Manage Course	Manage User	Profile																																																		
Course Details																																																				
Add Course																																																				
<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Course: Text (Year)</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Course: Text (Year)</div> <div style="border: 1px solid black; padding: 5px;">Course: Text (Year)</div>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4" style="text-align: center;"><u>Student</u></th> </tr> <tr> <th style="text-align: left;">Role</th> <th style="text-align: left;">Name</th> <th style="text-align: left;">Marks</th> <th style="text-align: left;">GPA</th> </tr> </thead> <tbody> <tr> <td>Text</td> <td>Text</td> <td>Text</td> <td>Text</td> </tr> <tr> <td>Text</td> <td>Text</td> <td>Text</td> <td>Text</td> </tr> <tr> <td>Text</td> <td>Text</td> <td>Text</td> <td>Text</td> </tr> <tr> <td>Text</td> <td>Text</td> <td>Text</td> <td>Text</td> </tr> <tr> <td colspan="4" style="text-align: center;">Lock result</td> </tr> <tr> <td colspan="4" style="text-align: center;">Add new Student</td> </tr> <tr> <td style="text-align: left;">Roll</td> <td colspan="3"> <input type="text" value="Text"/> <input style="float: right;" type="button" value="↓"/> </td> </tr> <tr> <td colspan="4" style="text-align: center;">Add</td> </tr> </tbody> </table>	<u>Student</u>				Role	Name	Marks	GPA	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Text	Lock result				Add new Student				Roll	<input type="text" value="Text"/> <input style="float: right;" type="button" value="↓"/>			Add				<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;"><u>Instructor</u></th> </tr> </thead> <tbody> <tr> <td colspan="2" style="text-align: center;">Present Instructor: Text</td> </tr> <tr> <td colspan="2" style="text-align: center;">Change Instructor</td> </tr> <tr> <td style="text-align: left;">Instructor</td> <td> <input type="text" value="Text"/> <input style="float: right;" type="button" value="↓"/> </td> </tr> <tr> <td colspan="2" style="text-align: center;">Change</td> </tr> </tbody> </table>	<u>Instructor</u>		Present Instructor: Text		Change Instructor		Instructor	<input type="text" value="Text"/> <input style="float: right;" type="button" value="↓"/>	Change	
<u>Student</u>																																																				
Role	Name	Marks	GPA																																																	
Text	Text	Text	Text																																																	
Text	Text	Text	Text																																																	
Text	Text	Text	Text																																																	
Text	Text	Text	Text																																																	
Lock result																																																				
Add new Student																																																				
Roll	<input type="text" value="Text"/> <input style="float: right;" type="button" value="↓"/>																																																			
Add																																																				
<u>Instructor</u>																																																				
Present Instructor: Text																																																				
Change Instructor																																																				
Instructor	<input type="text" value="Text"/> <input style="float: right;" type="button" value="↓"/>																																																			
Change																																																				
Footer																																																				

Figure 158: Internal (Program chairperson: Manage Course- Details)

Home
Header
Sign out

Manage Course
Manage User
Profile

Course Details

Add Course

Add new Course

Year

Course ↓

Instructor ↓

Footer

Figure 159: Internal (Program chairperson: Manage Course- Add)

Home
Header
Sign out

Manage Course
Manage User
Profile

View User

New User Request

Create Student Account

List of Users

Username	Full Name	Type	Email	Option
Text	Text	Text	Text	<input type="button" value="Edit"/>
Text	Text	Text	Text	<input type="button" value="Edit"/>
Text	Text	Text	Text	<input type="button" value="Edit"/>
Text	Text	Text	Text	<input type="button" value="Edit"/>

Footer

Figure 160: Internal (Program chairperson: Manage User- View)

Home	Header	Sign out
Manage Course	Manage User	Profile
View User		
New User Request		
Create Student Account		

List of Users				
Username	Full Name	Type	Email	Option
Text	Text	Text	Text	Edit
Text	Text	Text	Text	Edit
Text	Text	Text	Text	Edit
Text	Text	Text	Text	Edit

Footer

Home	Header	Sign out
Manage Course	Manage User	Profile
View User		
New User Request		
Create Student Account		

User Information	
Username:	Text
Full name:	<input type="text" value="Text"/>
Father's name:	<input type="text" value="Text"/>
Mother's name:	<input type="text" value="Text"/>
Email:	<input type="text" value="Text"/>
Update	

Footer

Figure 161: Internal (Program chairperson: Manage User- View - Edit)

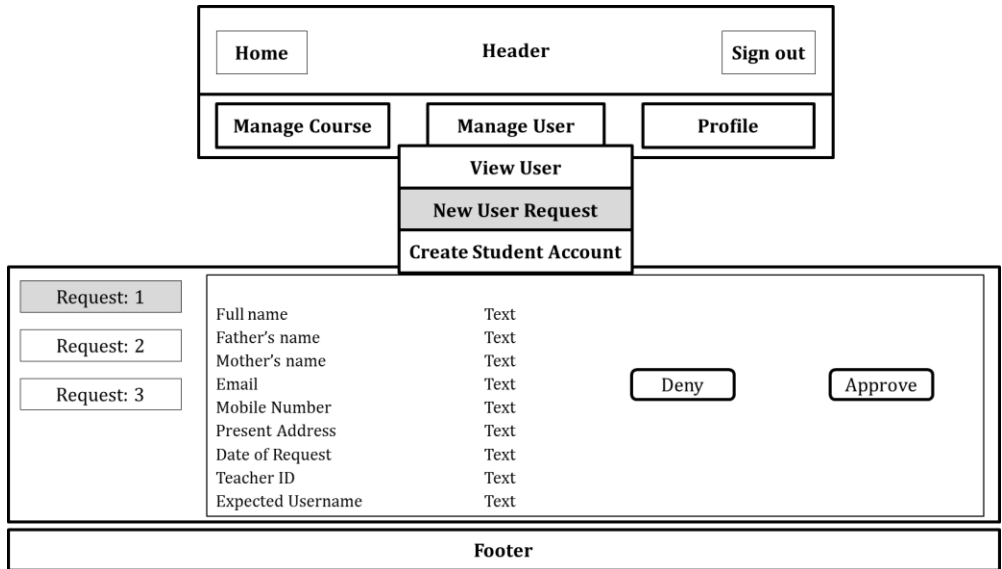


Figure 162: Internal (Program chairperson: Manage User- New User Request)

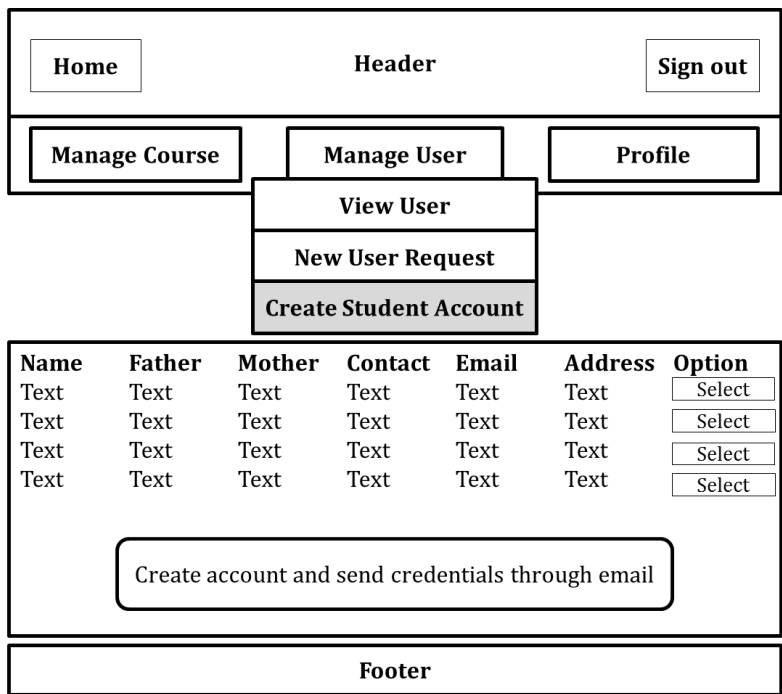
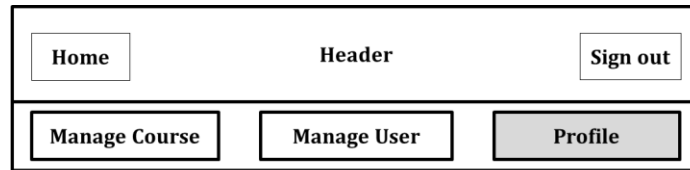


Figure 163: Internal (Program chairperson: Manage User- Create Student Account)



Same as Teacher: Profile

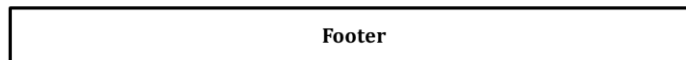


Figure 164: Internal (Program Chairman: Profile)

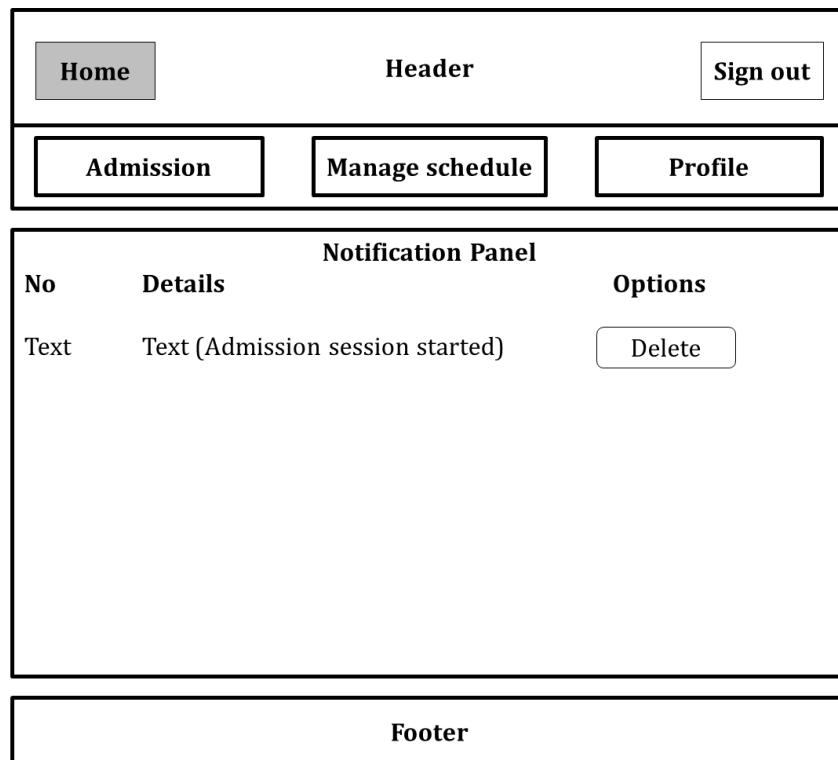


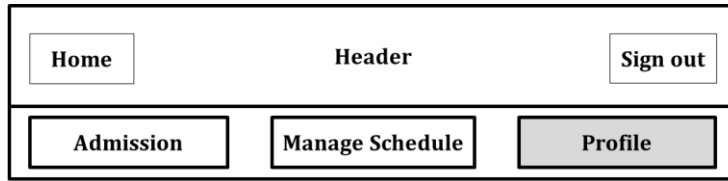
Figure 165: Internal (Exam controller: Home)

Home	Header	Sign out
Admission	Manage schedule	Profile
Session		
Eligible Applicants		
Admission Year		<input type="text" value="Date"/>
Form fill up (start)		<input type="text" value="Date"/>
Form fill up (end)		<input type="text" value="Date"/>
Application payment (Start)		<input type="text" value="Date"/>
Application payment (end)		<input type="text" value="Date"/>
Scrutinizing Process (start)		<input type="text" value="Date"/>
Admission date		<input type="text" value="Date"/>
Admission payment (start)		<input type="text" value="Date"/>
Admission payment (end)		<input type="text" value="Date"/>
<input type="button" value="Start New Session"/>	<input type="button" value="Update Session"/>	<input type="button" value="Stop Session"/>
Footer		

Figure 166: Internal (Exam controller: Admission-Session)

Home	Header	Sign out	
Admission	Manage schedule	Profile	
Session			
Eligible Applicants			
Application ID	Name	Contact No	Email
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
Text	Text	Text	Text
<input type="button" value="Stop Scrutinizing"/>		<input type="button" value="Generate Admit Card"/>	
Footer			

Figure 167: Internal (Exam controller: Admission-Eligible Applicants)



Same as Teacher: Profile

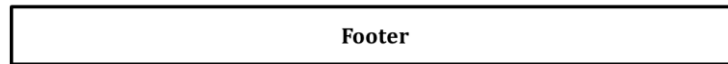


Figure 170: Internal (Exam controller: Profile)

DEFINE EVENTS THAT WILL CAUSE THE STATE OF THE USER INTERFACE TO CHANGE

1. Admission

1.1. when admission session is ended:

Ex. Home → Admission Go → Admission Session Ended

1.2. When admission is going

Ex. Home → Admission Go → Fill up form → Submit → Show form info.

2. Admit Card

2.1. When card is available

Ex. Home → Admission Go → Download

2.2. When card is unavailable

Ex. Home → Admission Go → Download → Not available

3. Create Account (Only for teacher)

Ex. Home → Account Go → Fill up form → Submit

4. Sign In

Ex. Home → Sign in → Fill up input → Log in → Home

5. Profile

Home → Profile → Fill up input → Update

6. Update Courses Details

Home → Manage course → Course details → Fill up input → Update

7. Update Marks

Home → Manage course → Update marks → Fill up input → Update

8. Notify Supplementary Student

Home → Manage course → List of students → Notify All

9. Show Viva Schedule

Home → Viva schedule

10. View Courses

Home → View courses

11. View Result

Home → View result

12. Search Applicant By Name

Home → Search applicant → by phone n ^{Fill up input} → Search

13. Search Applicant by Phone No.

Home → Search applicant → by phone no. ^{Fill up input} → Search

14. Verify Admission Payment

Home → Payment → Admission payment ^{Fill up input} → Paid

15. Verify Semester Fees

Home → Payment → Application fee ^{Fill up input} → Paid

16. Download All Application

Home → Search applicant → Download all

17. Verify Applications

Home → Search applicant ^{Fill up input} → Search → Eligible

18. Add Student to Course

Home → Manage course → Details ^{Fill up input} → Add

19. Change Course Teacher

Home → Manage course → Details ^{Fill up input} → Change

20. Add Course



21. Update User Information



22. Approve/Deny User Request



23. Create User



24. Start Admission Session



25. Stop/Update Session



26. Generate Admit Card



27. Manage Viva Schedule



28. Update Schedule Room



29. Log out



DEPICT EACH INTERFACE STATE AS IT LOOK TO END USER

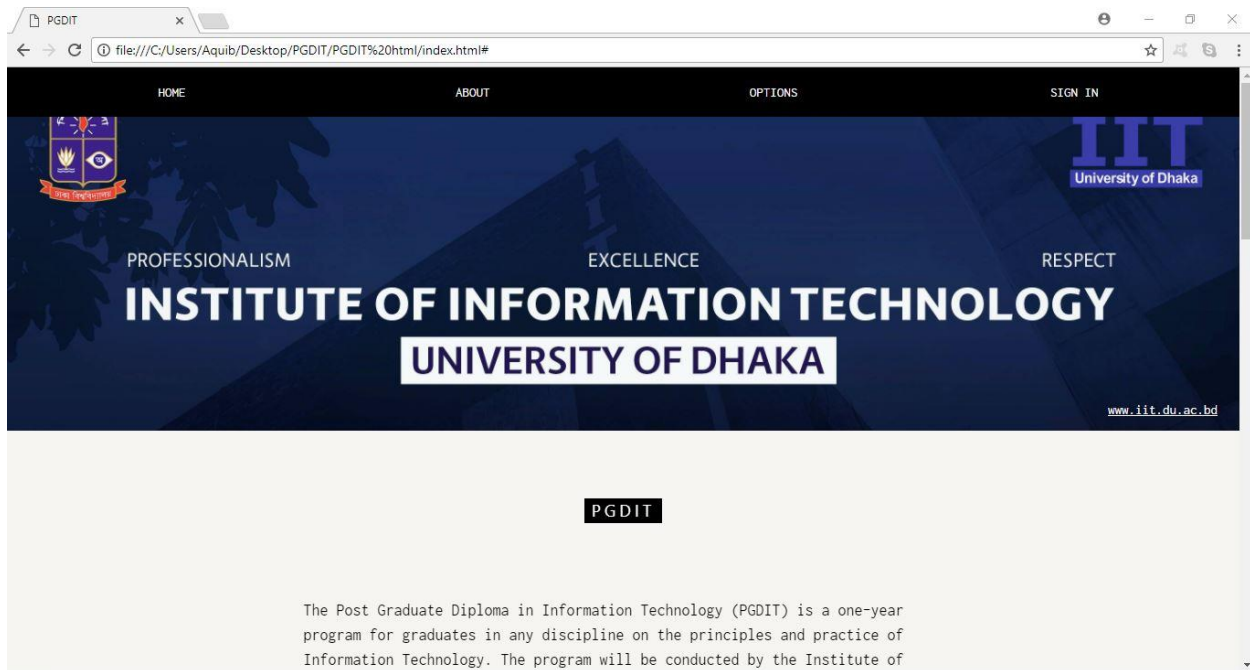


Figure 171: Home



Figure 172: Notices

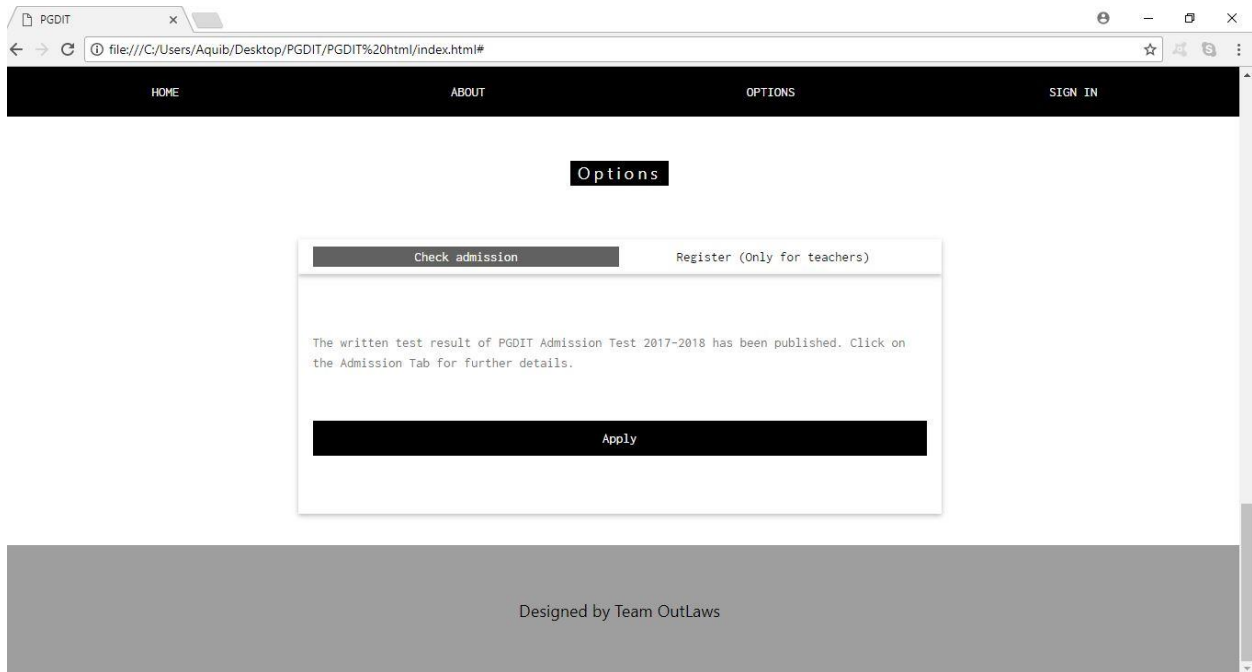


Figure 173: Options

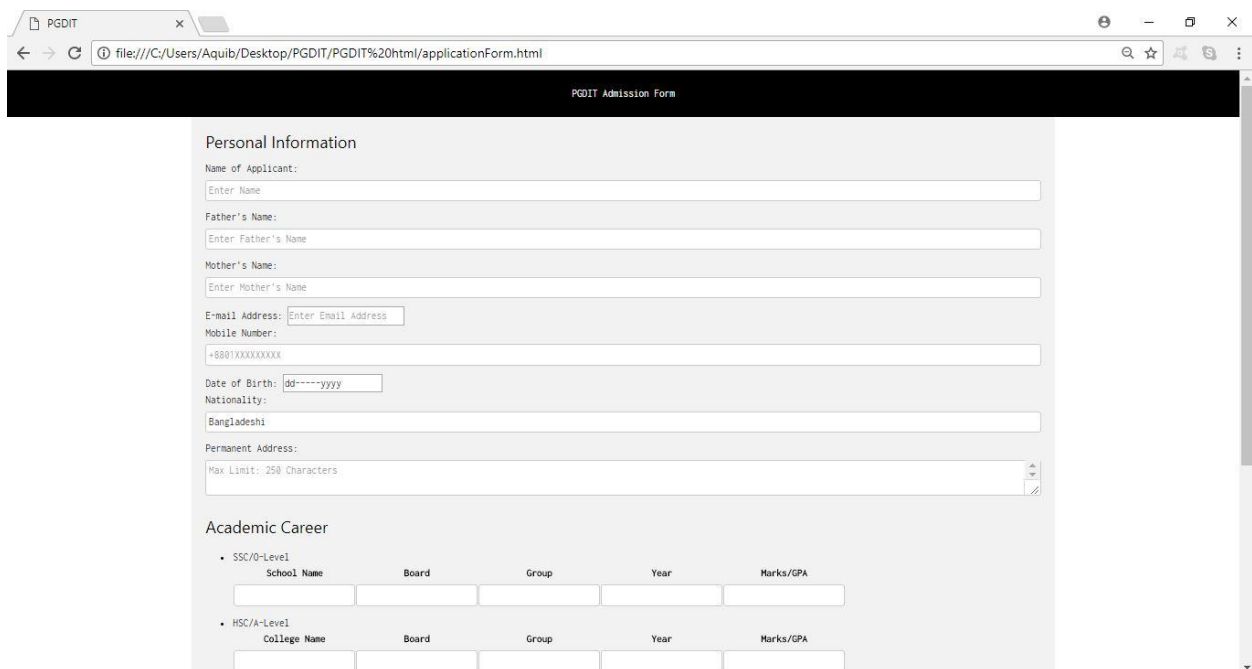


Figure 174: Admission form

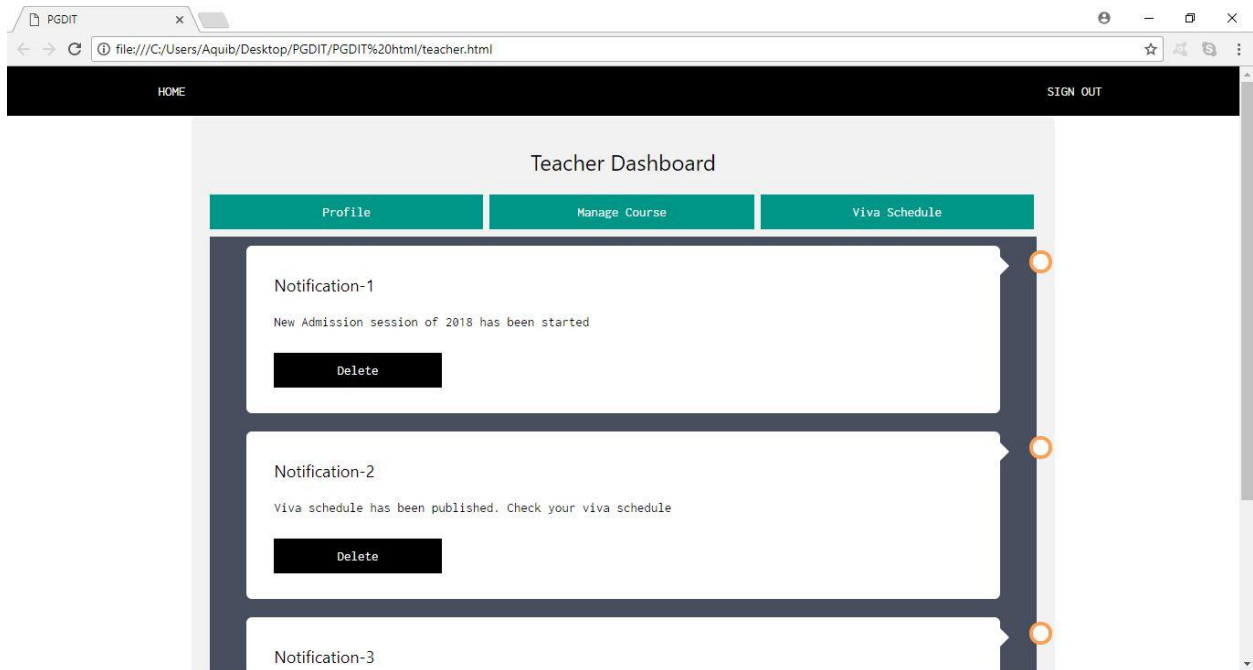


Figure 175: Dashboard of Teacher

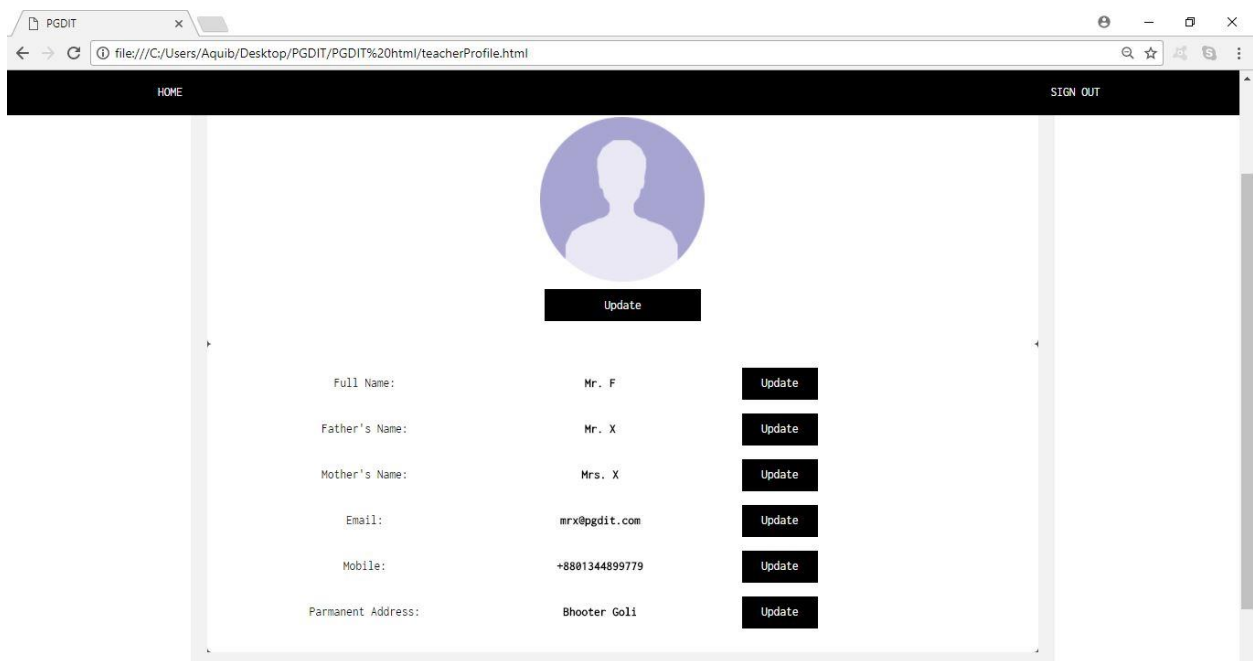


Figure 176: Profile

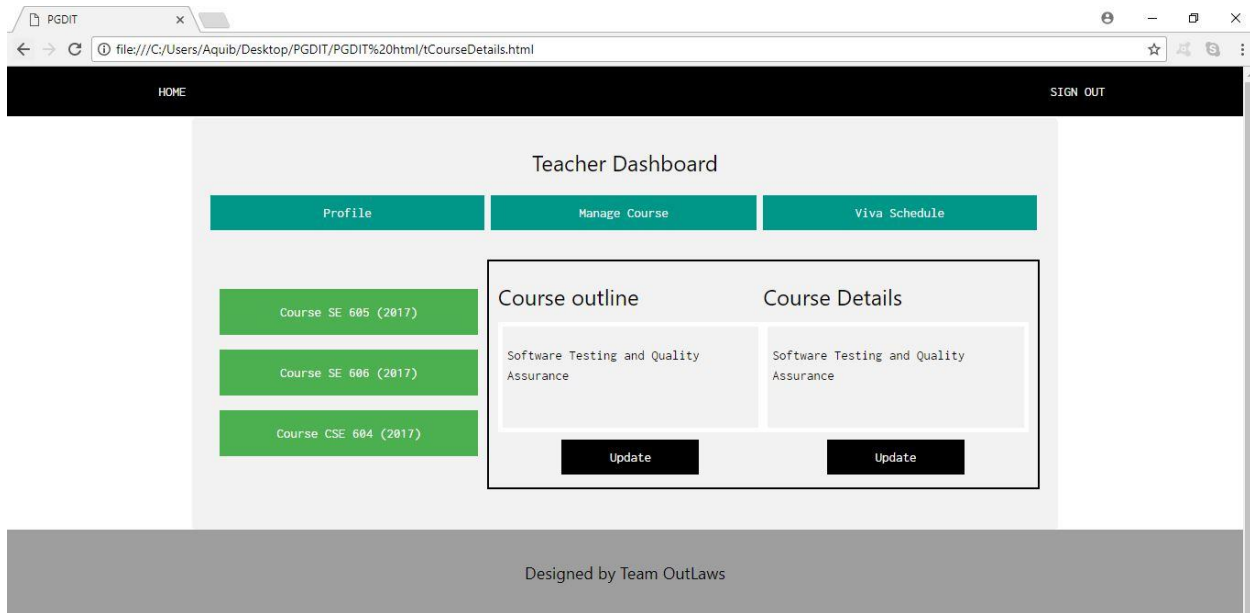


Figure 177: Course details

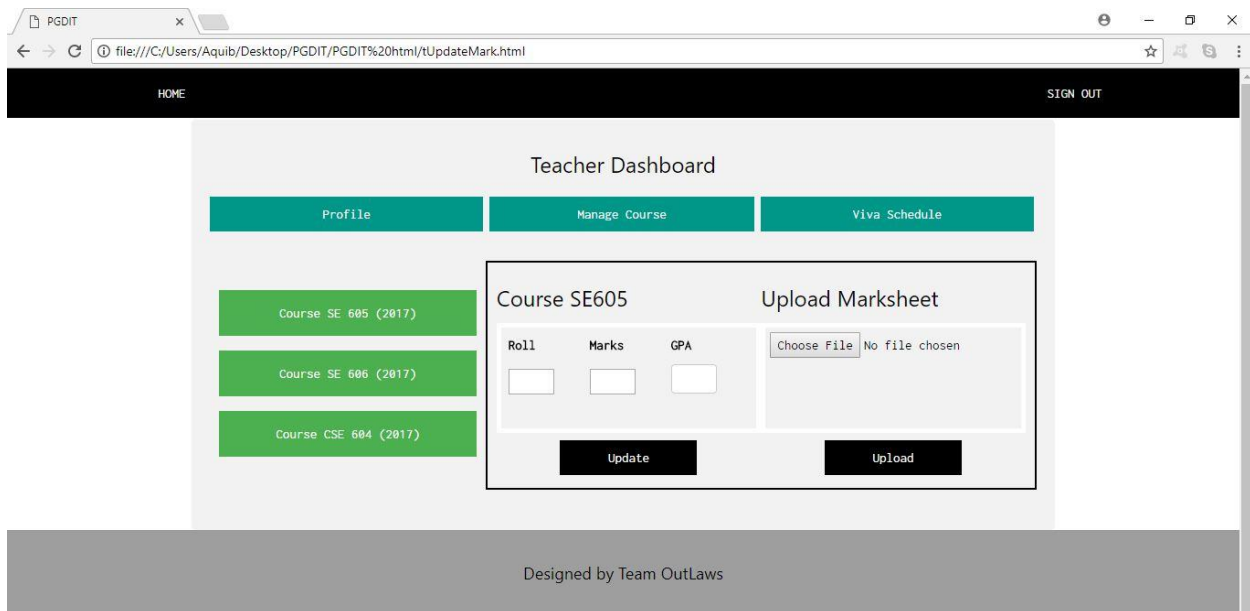


Figure 178: Update marks

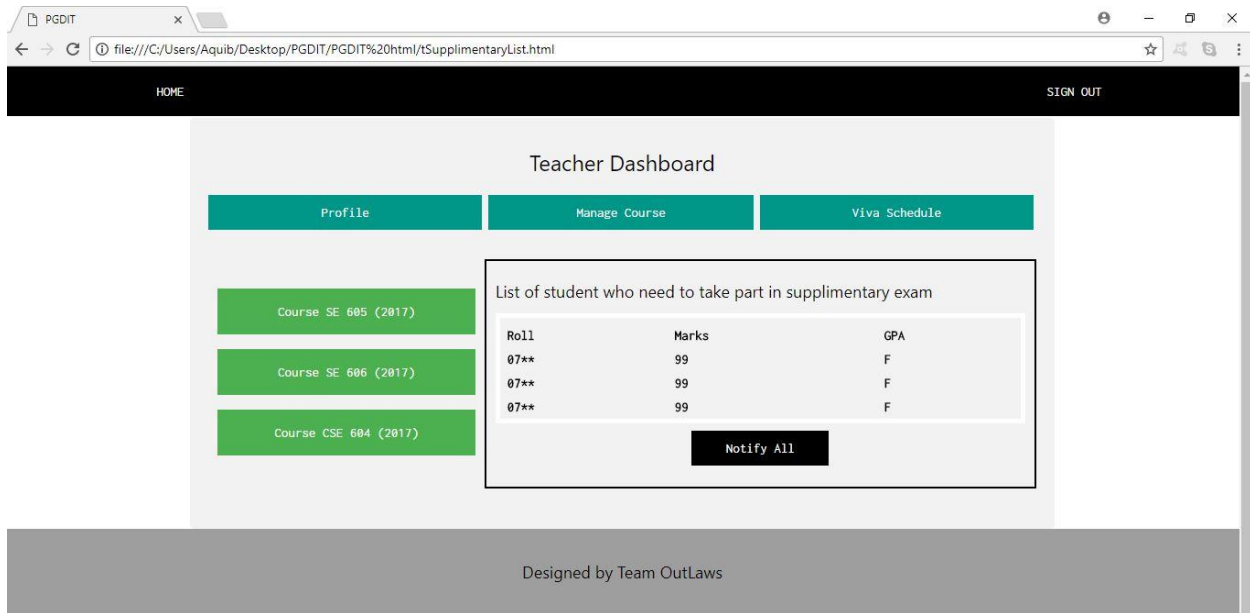


Figure 179: Supplementary exam list

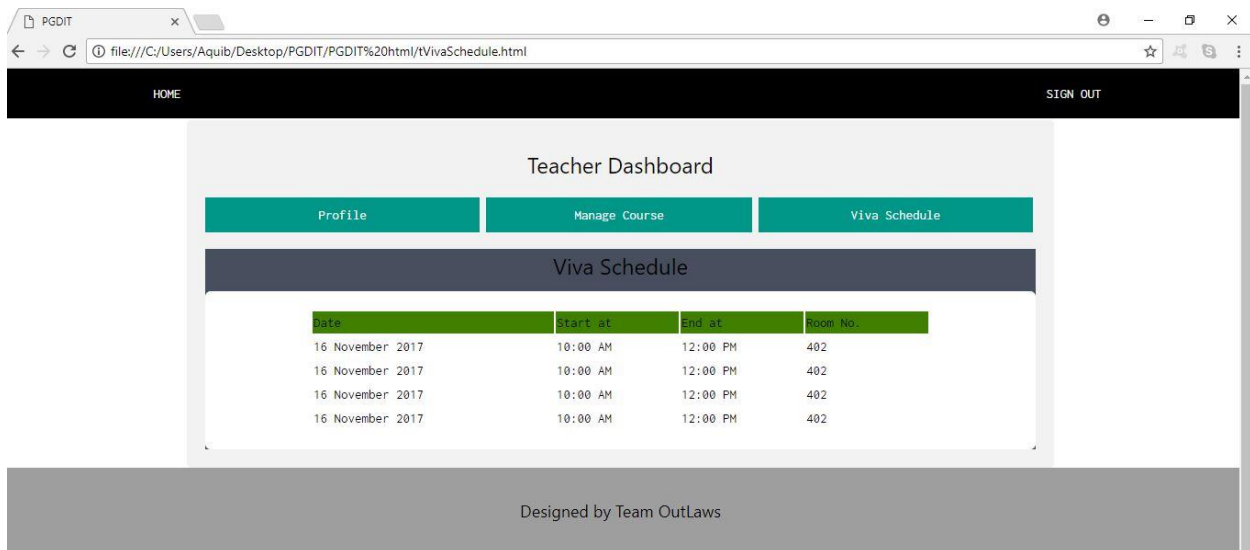


Figure 180: Viva schedule

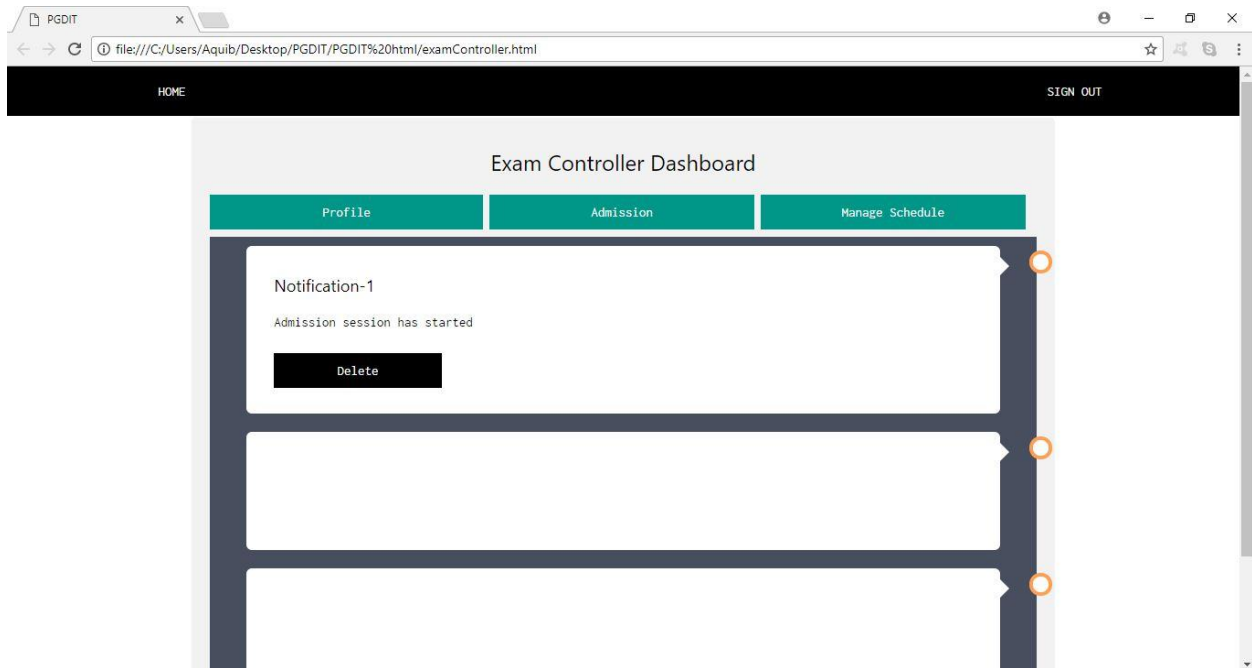


Figure 181: Dashboard of exam controller

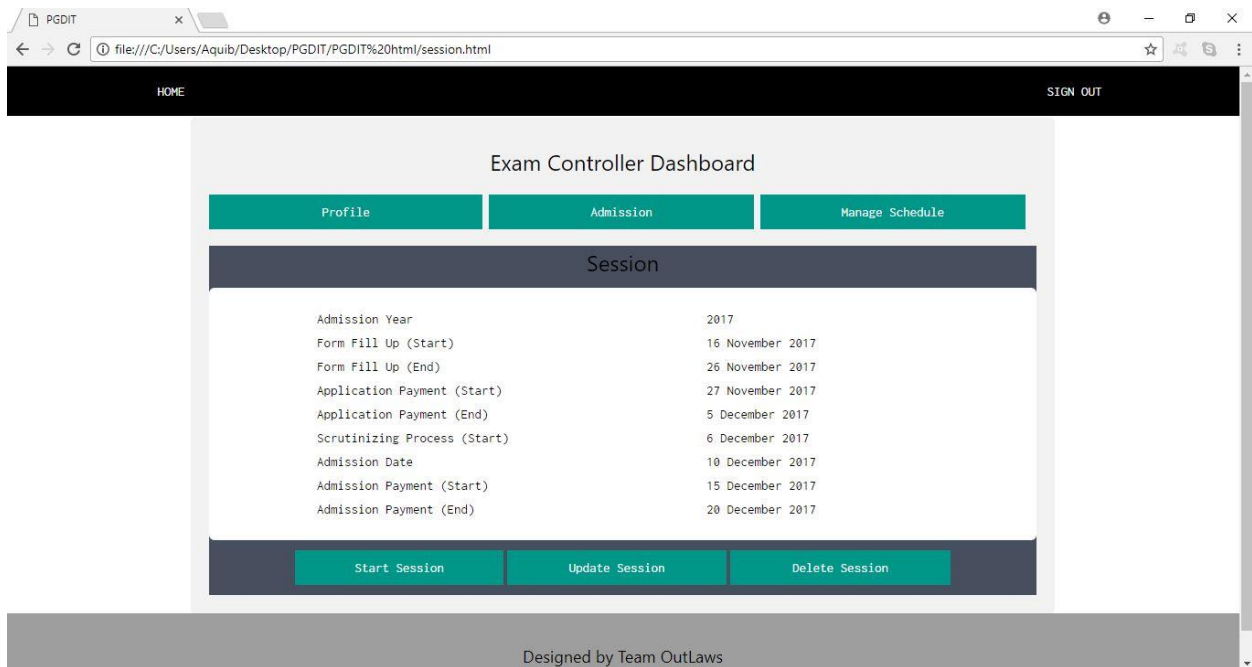


Figure 182: Admission session

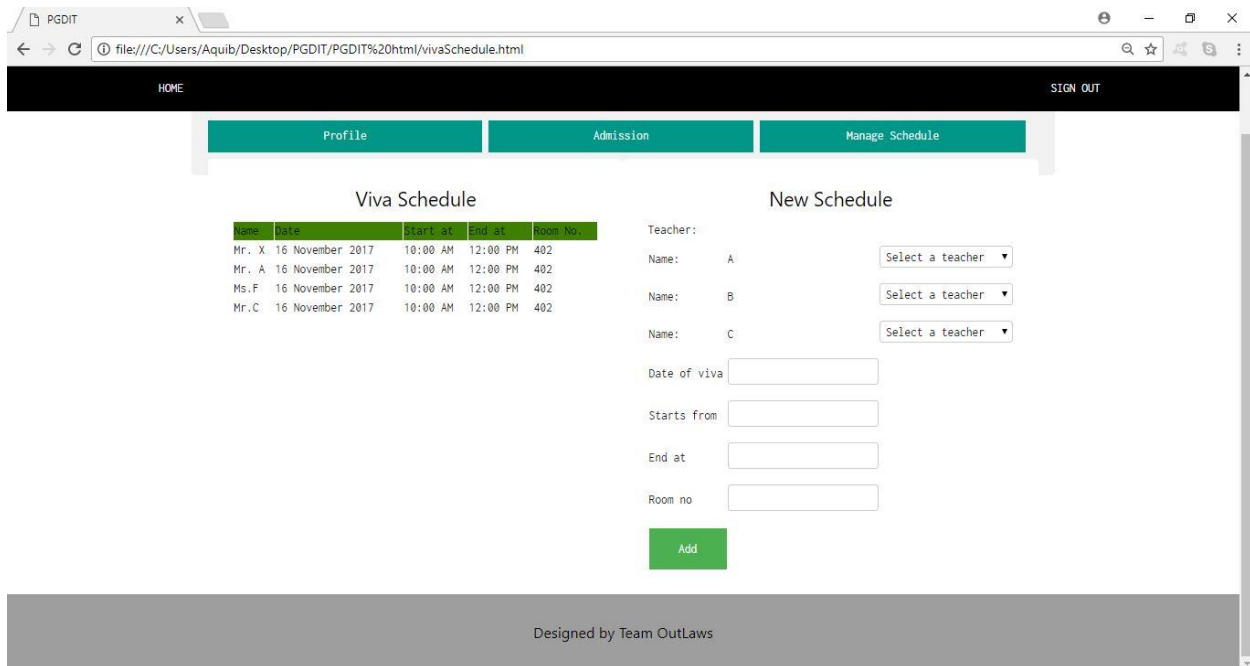


Figure 183: Viva schedule manage

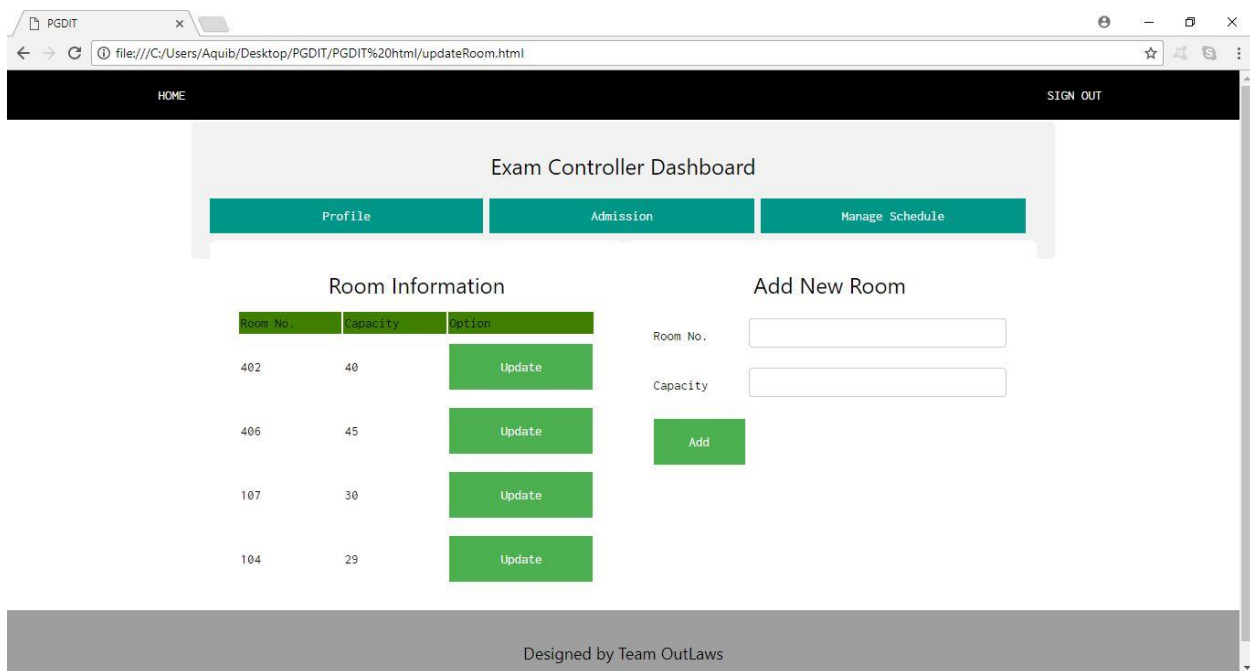


Figure 184: Update room