**Assignment 3**

| **Name** | **NU ID** |
| --- | --- |
| Kshitija Suryavanshi | 1539935 |
| Animesh Shukla | 2109910 |
| Minal Makwana | 2130495 |

**Class Diagram:**

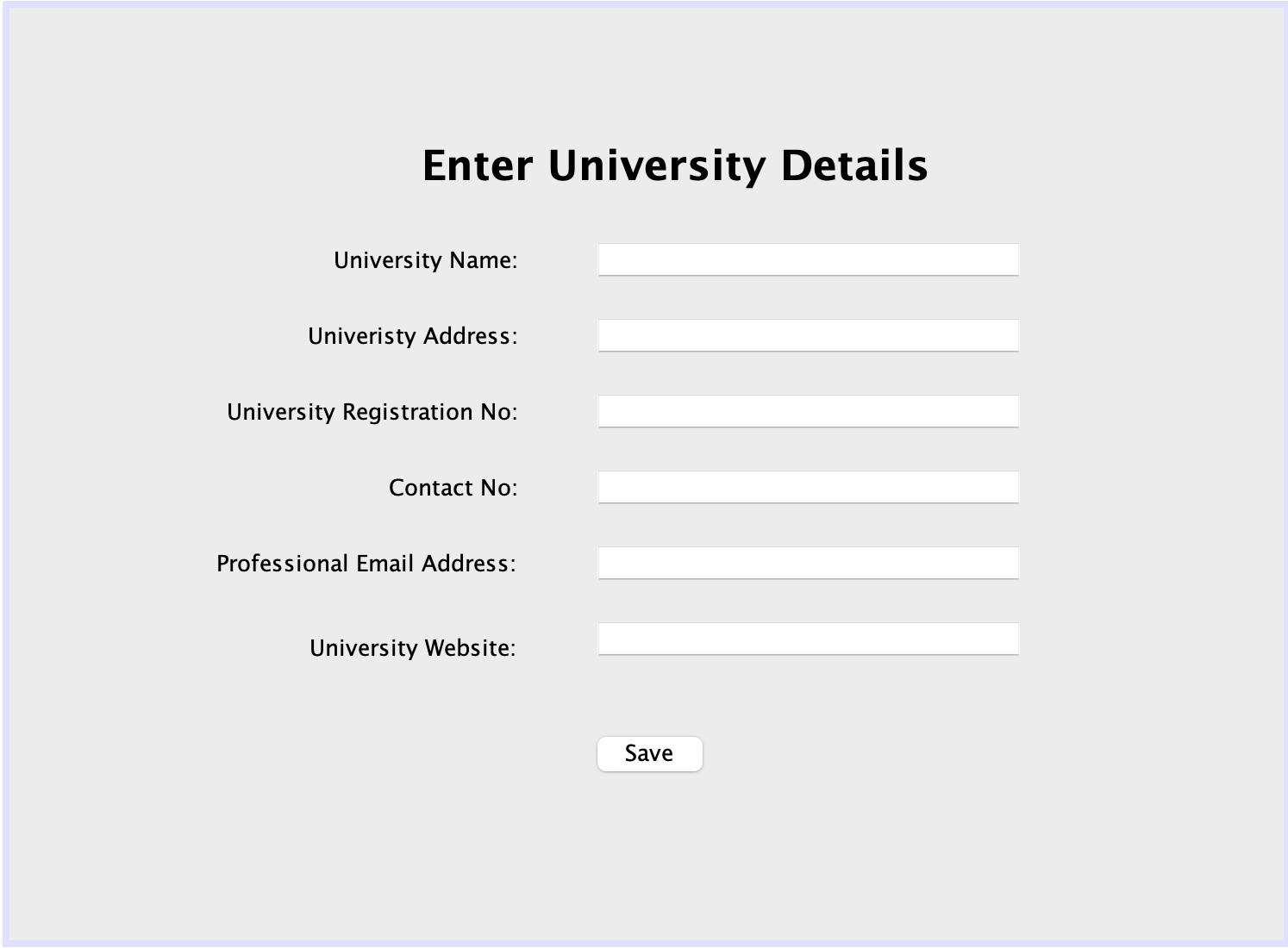
<https://lucid.app/lucidchart/8ecc0026-8b8e-4a7b-a884-e633f4092961/edit?view_items=ppjlqSx3Ewtu&invitationId=inv_5d5d9d77-a01e-4a47-88aa-c3fae4f8676a>

**Sequence Link:**

<https://lucid.app/lucidchart/144f6ef4-072d-4710-a13b-ded94594b72d/edit?viewport_loc=-9%2C50%2C1707%2C983%2C0_0&invitationId=inv_eb9f26f4-2bec-4a06-8dbd-af7d45f42724>

<https://lucid.app/lucidchart/125bfeee-4e86-41db-a7ab-dc73d71edc85/edit?viewport_loc=-70%2C-1654%2C2299%2C1324%2C0_0&invitationId=inv_99c7e447-1636-413d-8225-e5dc71c2322c>

**University Class:**



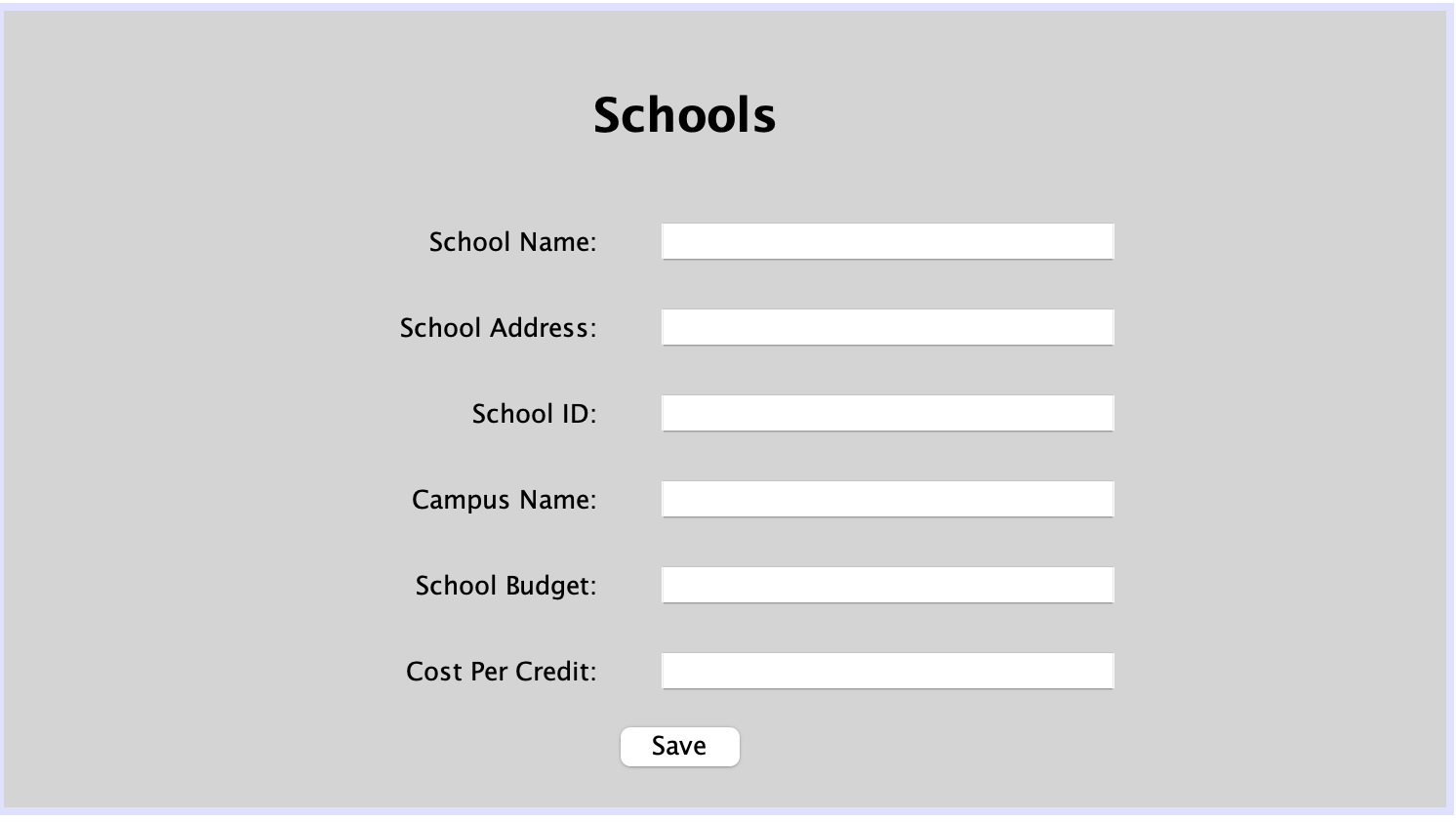
Attributes:

* +adminUsername: String
* +adminPassword: String
* +universityName: String
* +universityAddress: String
* +universityRegNo: String
* +universityContactNo: Int
* +univeristyEmailAdd: String
* +universityWebsite: String
* +schoolList:ArrayList - This will maintain the records of all school under the particular university.

Operations:

* +getadminUsername(): String
* This operation will return the admin Username.
* +getadminPassword(): String
* This operation will return the admin Password.
* +getuniversityName(): String
* This operation will return the University Name to the user.
* +getuniversityAddress(): String
* This operation will return the University Address to the user.
* +getuniversityRegNo(): String
* This operation will return the University Registration Number.
* +getschoolList(): String
* This operation will List the number of schools included in the University.
* +getDetails()
* This getter method will get the value of each university class variable.
* +setDetails()
* This setter method will set or update the value of each university class variable.

**School class:**



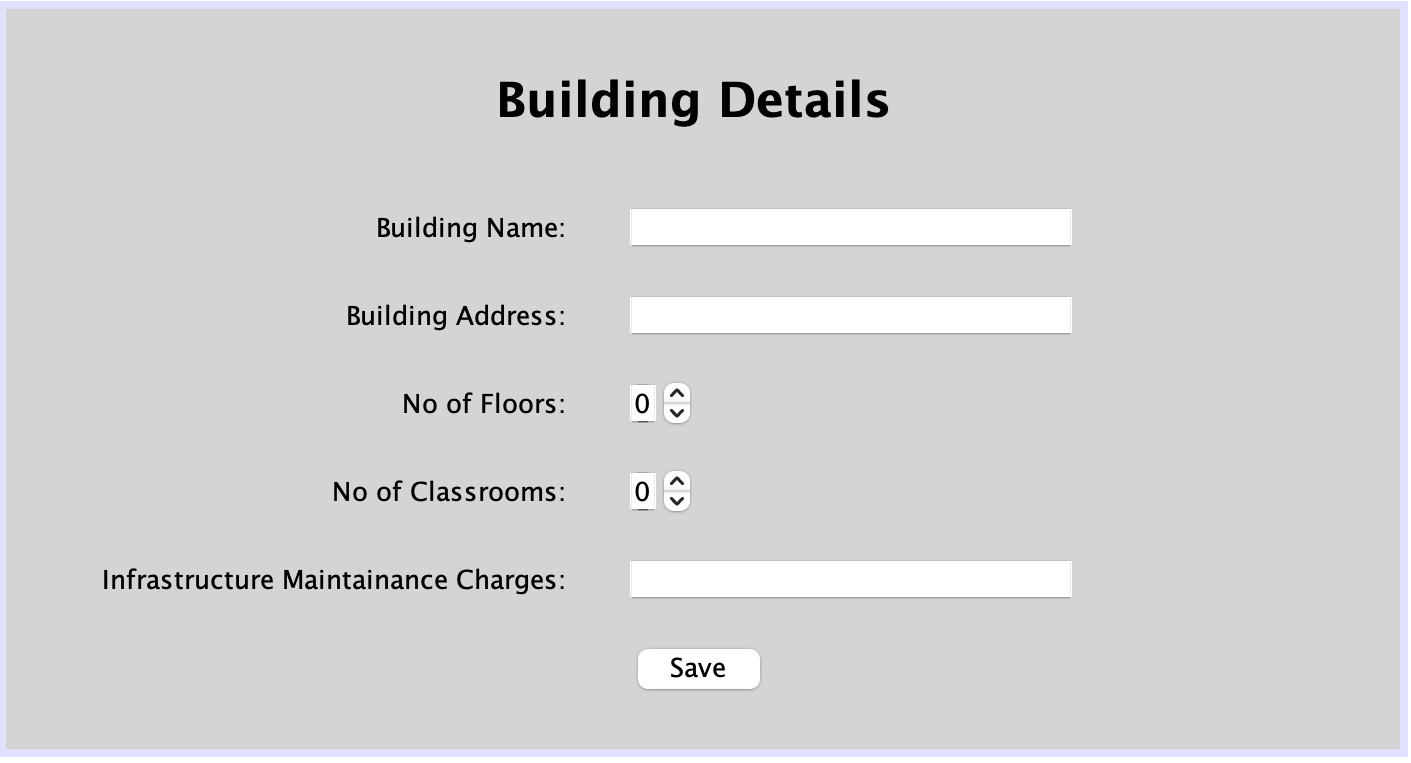
Attributes:

* +schoolName:String
* +schoolAddress:String
* +idSchool:Int
* +campusName:String
* +budgetSchool:Int
* +costPerCredit:Int
* +departmentList:ArrayList - This will maintain the records of all department under the particular school.

Operations:

* +getschoolName(): String
* This operation will return the school Name.
* +getCurrentEnrollment(Semester)
* This operation will return the number of students who are currently enrolled in all departments of the school.
* +getschoolAddress(): String
* This operation will return the school Address.
* getCurrentTutionTotalCreditsBySemester(Semester)
* This operation will return the current total tution fees per credit by semester.
* +getCurrentTutionTotalCreditsByDepartment(departmentId)
* This operation will return the current total tution fees per credit by department.
* +getAllDepartment()
* This operation will return all the department names under the school.
* +getTotalExpenses()
* This operation will return the total expenses of school so this will calculate the salary of faculty, staff and infrastructure Maintenace charges for the respective school buildings and thus generate total expenses under school level.
* +getDetails()
* This getter method will get the value of each student class variable.
* +setDetails()
* This setter method will set or update the value of each student class variable.

**Building class:**



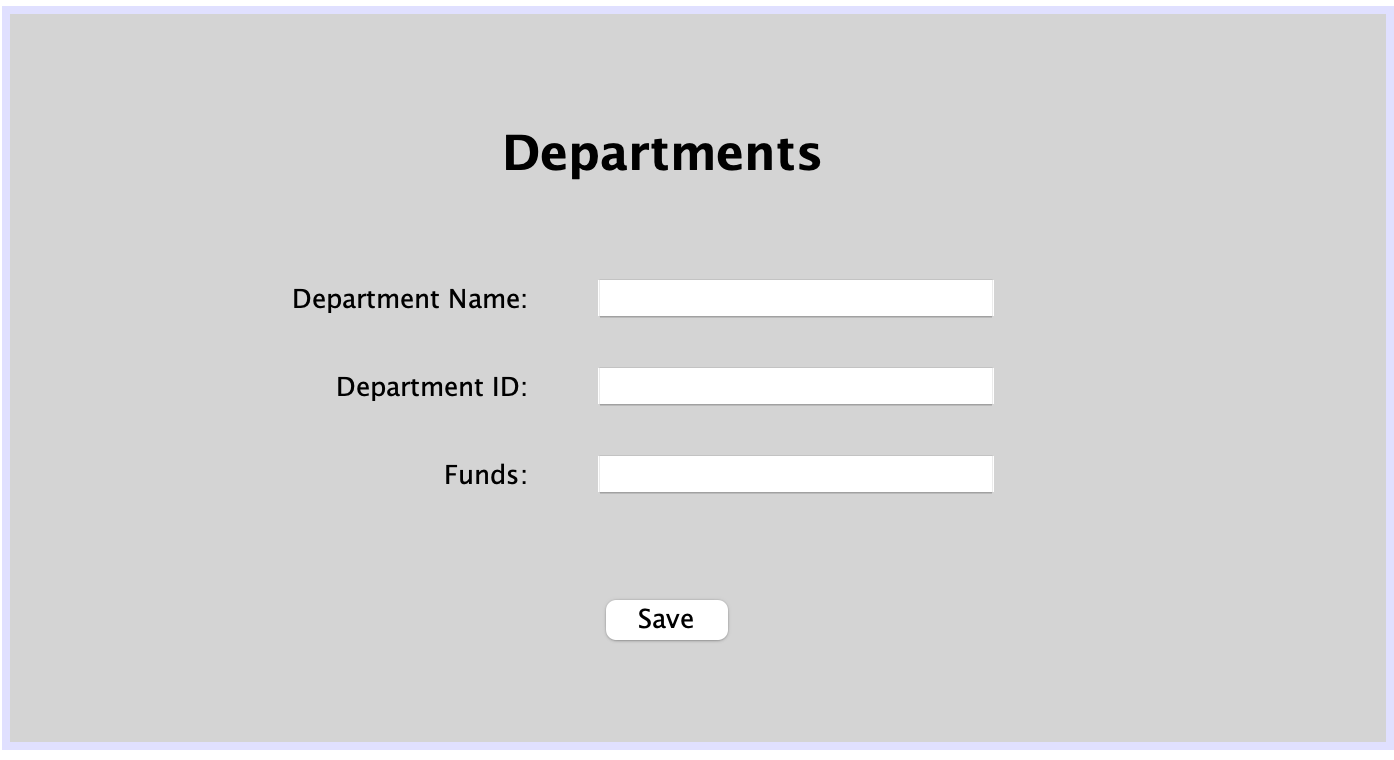
Attributes:

* +buildingName: String
* +buildingAddress: String
* +noOfFloors: Int
* +noOfClassrooms: Int
* +InfraMaintananceCharges: Int

Operations:

* +getbuildingName(): String
* This operation will return the Building Name
* +getbuildingAddress(): String
* This operation will return the Building Address.
* +getNoOfFloors() : Int
* This operation will return number of floors in the building.
* +getNoOfClassrooms(): Int
* This operation will return number of classrooms in the building
* +getDetails()
* This getter method will get the value of each Building class variable.
* +setDetails()
* This setter method will set or update the value of each Building class variable.

**Department class:**



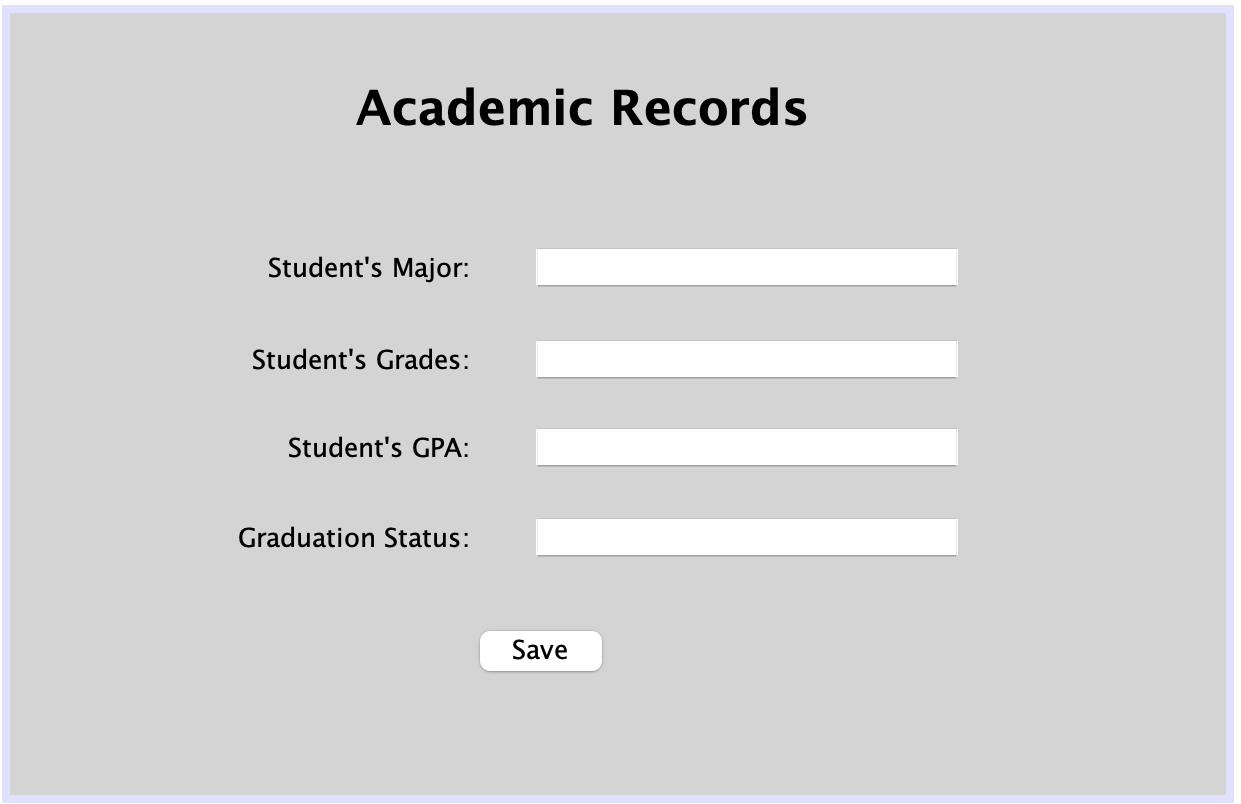
Attributes:

* +departmentName:String
* +departmentId:Int
* allCourseList:ArrayList - This will maintain the records of all courses offered at the department level.
* mandatoryCourseList:ArrayList - This will maintain the records of all Mandatory courses required at the department level.
* facultyList:ArrayList - This will maintain the records of all the teaching faculty associated with their respective department.
* currentStudentList:ArrayList - This will maintain the records of all students currently studying in various courses at the department level.

Operations:

* +getTotalNoOfRegisteredStudents(CourseId)
* This operation will return total number of registered students.
* +getTotalINoOfAvailableSeats(Course)
* This operation will return total number of available seats.
* +getTotalCreditHoursEnrolledPerSemseter(Semester)
* This operation will return total credit hours enrolled per semester.
* +getAllCoursesWithEmptySeats()
* This operation will return all courses with empty seats.
* +getGrantsAllocatedToResearch()
* This operation will return total grants allocated to research department.
* +getcourseList()
* This operation will return list of courses
* +getfacultyList()
* This operation will return list of faculties.
* +getcurrentStudentList()
* This operation will return list of current students.
* +getTotalNoOfSeatsDept()
* This operation will return total number of seats in a department.
* +getTotalNoOfFilledSeatDept
* This operation will return total number of seats filled in a department.
* +getDetails()
* This getter method will get the value of each Department class variable.
* +setDetails()
* This setter method will set or update the value of each Department class variable

**Academic Record:**



Attributes:

* +studentMajor
* +studentGrade
* +studentGPA: float
* +studentgraduationstatus:String

Operations:

* +getAllStudentGrades()
* This operation will return grades of all students.
* +getAllStudentGPA()
* This operation will return GPA of all students.
* +getAvgStudentGPA()
* This operation will return average of student’s GPA.
* +getAllStudentEligibleForGraduation()
* This operation will return all students who are eligible for graduation
* +getDetails()
* This getter method will get the value of each Academic Record class variable.
* +setDetails()
* This setter method will set or update the value of each Academic Record class variable

**Staff class:**



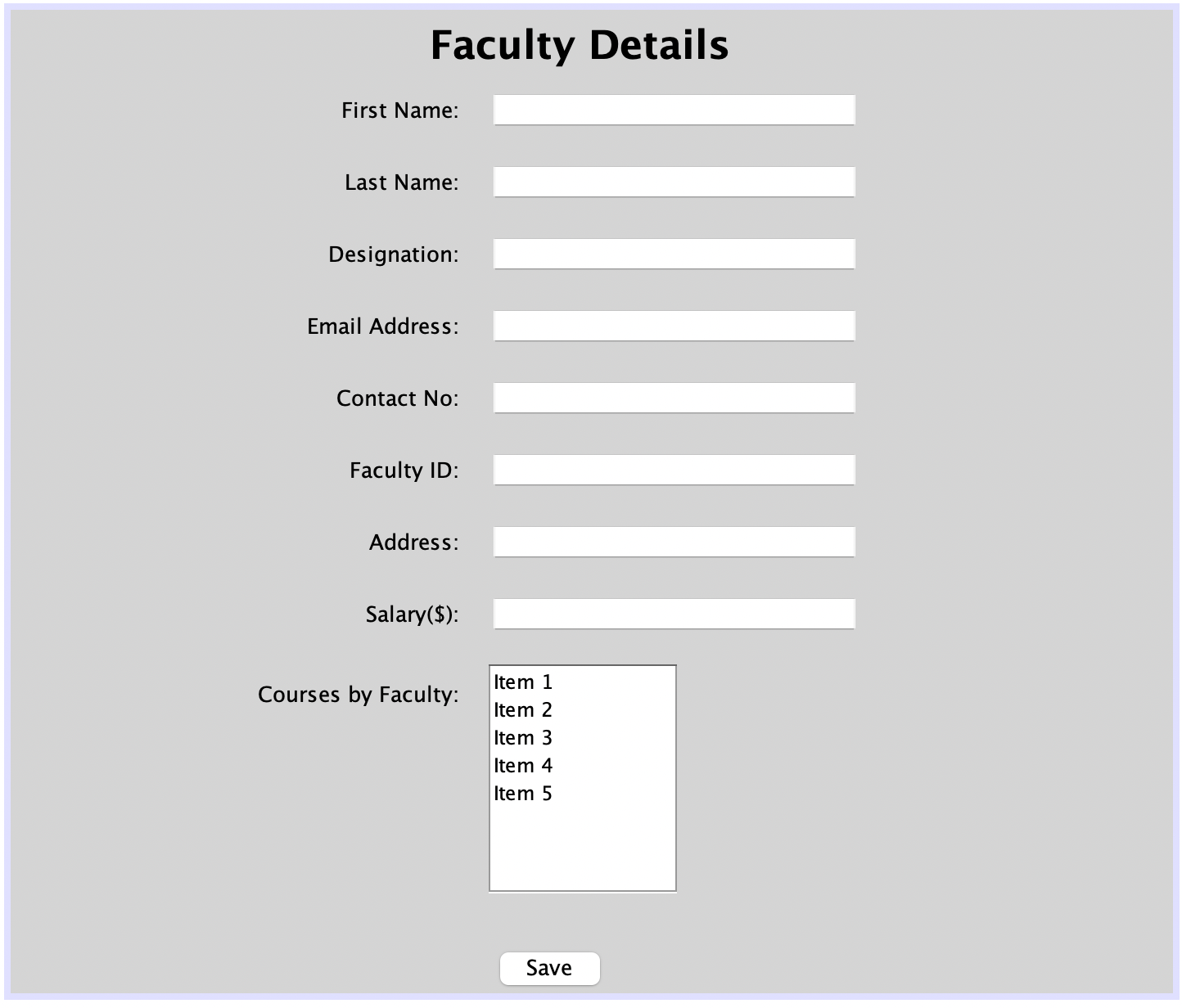
Attributes:

* + firstNameStaff:String
* +lastNameStaff:String
* +designationStaff:String
* +emailAddressStaff:String
* +contactNoStaff:Int
* +idNoStaff:Int
* +addressStaff:String
* +salaryStaff:Int
* +campus:String
* +SSN:Int
* +departmentType:String
* +employeeType:String

Operations:

* +updateStaffAttendance()
* This operation will return the updated staff attendance.
* +getDetails()
* This getter method will get the value of each staff class variable.
* +setDetails()
* This setter method will set or update the value of each staff class variable

**Faculty class:**



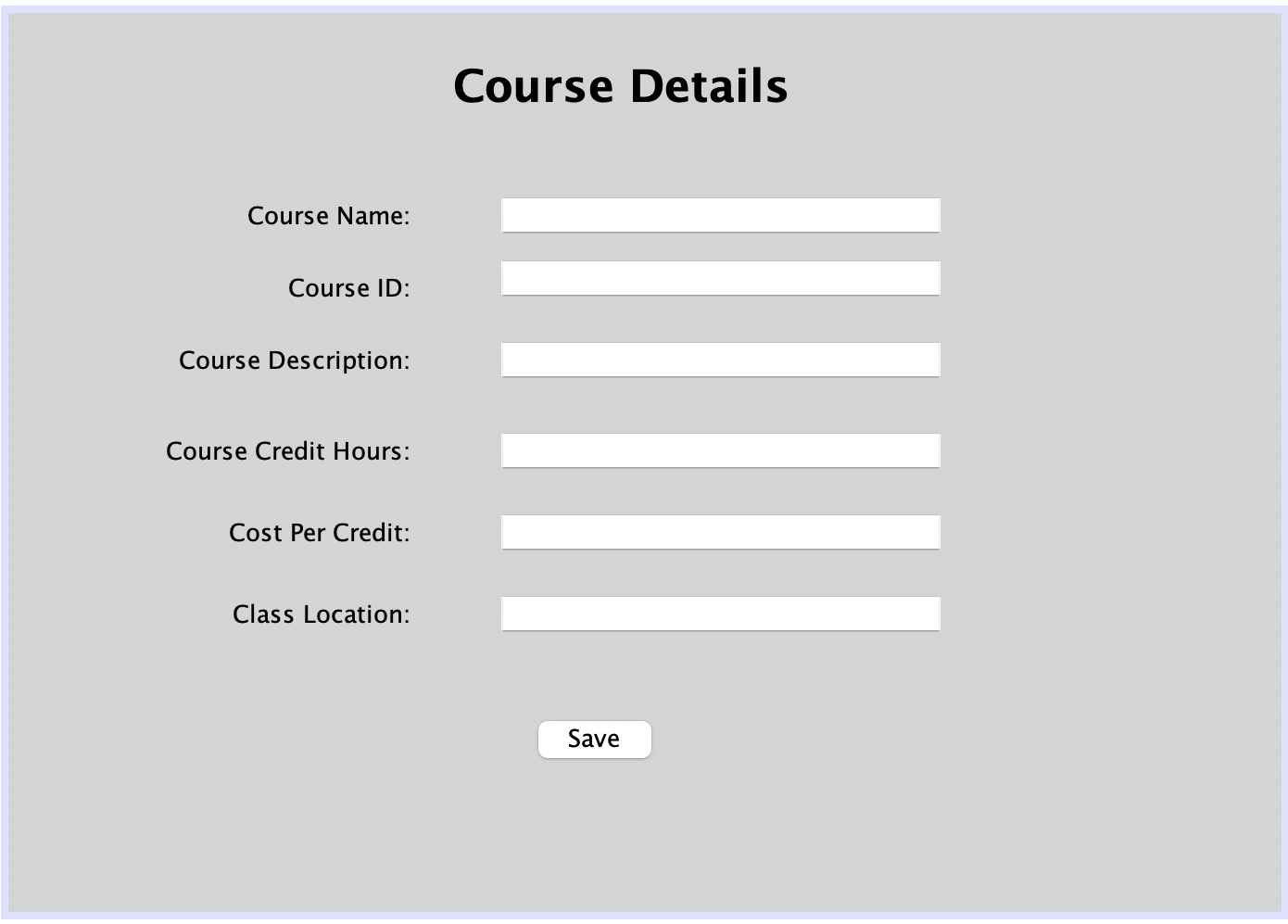
Attributes:

* +firstNameFaculty:String
* +lastNameFaculty:String
* +designationFaculty:String
* +emailAddressFaculty:String
* +contactNoFaculty:Int
* +idNoFaculty:Int
* +addressFaculty:String
* +salaryFaculty:Int
* +coursesTaught:ArrayList - This will maintain the records of all the courses taught by faculty.

Operations:

* +updateCourseSyllabus(courseDescription)
* This operation will update the course Syllabus.
* +getFacultySchedule(idNoFaculty)
* This operation will return the faculty schedule.
* +updateAssignments()
* This operation will update the assignment details.
* +updateGrade()
* This operation will update the grades.
* +updateFacultyAttendance()
* This operation will update the faculty attendance.
* +getDetails()
* This getter method will get the value of each faculty class variable.
* +setDetails()
* This setter method will set or update the value of each faculty class variable

**Course class:**



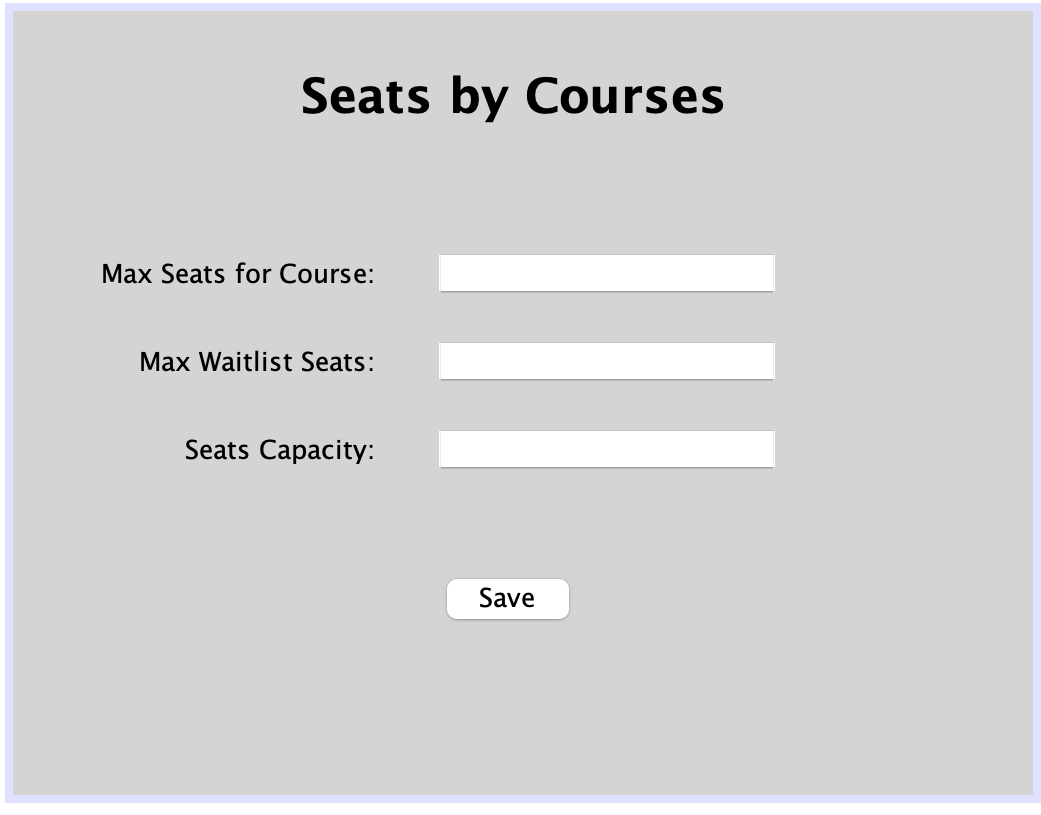
Attributes:

* +courseName:String
* +courseId:Int
* +courseDescription:String
* +creditHours:Int
* +costPerCredit:Int
* +classroomLocation:String
* +seatsList: ArrayList - This will maintain the records of all the seats available under each courses.

Operations:

* +findCourseByName(courseName)
* This operation will return the course by name.
* +findCourseById(courseId)
* This operation will return the course by ID
* +getAllCourses()
* This operation will return all courses
* +getTotalNoOfCourseFilledSeats()
* This operation will return total number of seats filled in one course.
* +hasEmptySeats()
* This operation will return all the empty seats in one course.
* +getAllSeats()
* This operation will return list of all the seats.
* +getDetails()
* This getter method will get the value of each course class variable.
* +setDetails()
* This setter method will set or update the value of each course class variable

**Seat class:**



Attributes:

* +maxSeats:Int
* +maxWaitlist:Int
* +seatCapacity:Int

Operations:

* +getDetails()
* This getter method will get the value of each seat class variable.
* +setDetails()
* This setter method will set or update the value of each seat class variable

**Current Students:**

Attributes:

* +firstName:String
* +lastName:String
* +emailAddress:String
* +contactNo:Int
* +idNo:Int
* +gender:String
* +nationality:String
* +address:String
* +fees:Int
* +program:String
* +studentLevel:String
* +courseRating:ArrayList - Current Student will rate the course on the scale of 0 to 5.
* +facultyRating:ArrayList - Current Student will rate the faculty on the scale of 0 to 5.

Operations:

* +getDetails()
* This getter method will get the value of each current student’s class variable.
* +setDetails()
* This setter method will set or update the value of each current student’s class variable
* +payFees()
* This operation will return the current student’s fees.
* +getprogram()
* This operation will return the current student’s program detail.

**Alumni Employment:**



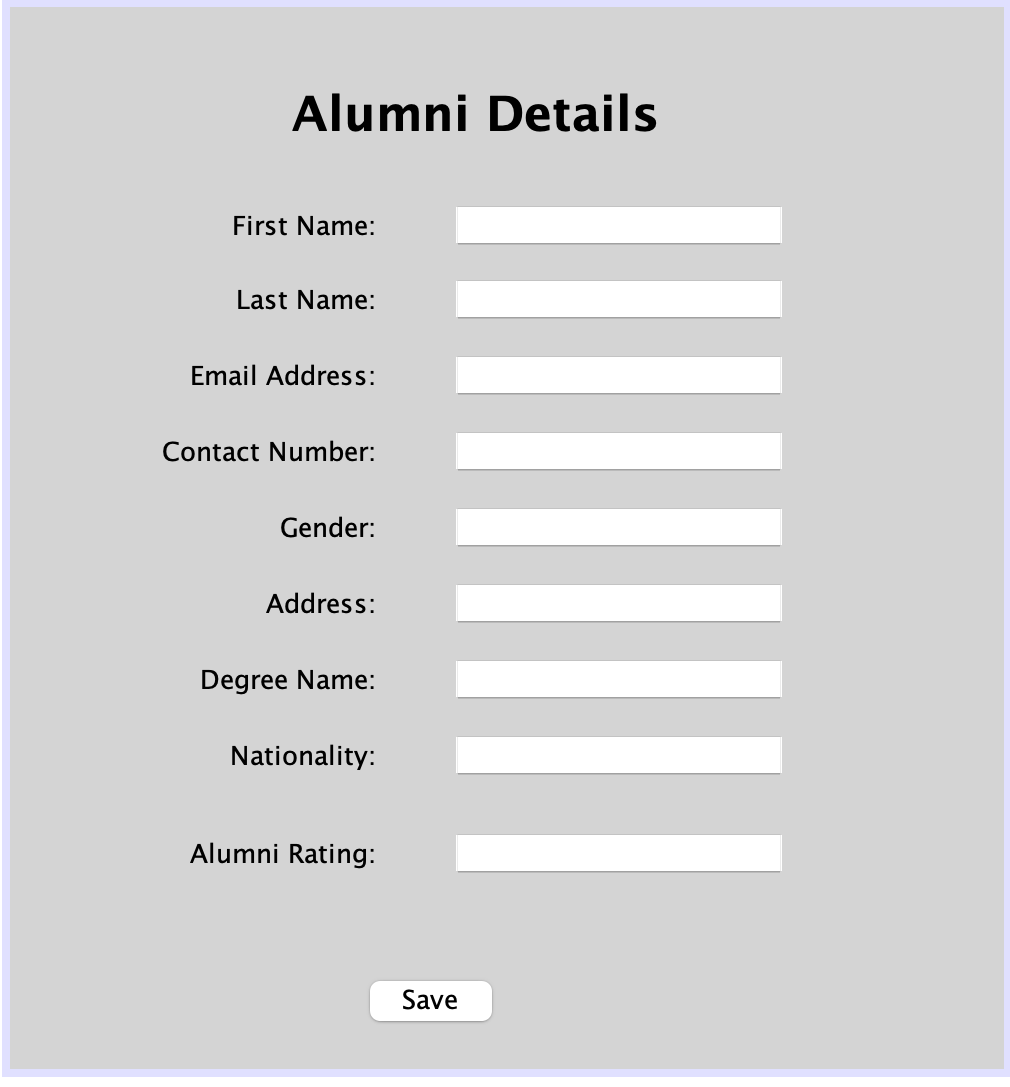
Attributes:

* +companyname: String
* +employeename: Varchar
* +employeeId: Int
* +dateofjoining: Int
* +employeeDesignation: Varchar
* +employeeSalary: Float
* +employeeDepartment: Varchar
* +yearsofexperience: String
* +employerRating:ArrayList. - Employer will rate the Almuni on the scale of 0 to 5 on basis annual performance and promotions.

Operatons:

* +getDetails()
* This getter method will get the value of each Almuni Employment Details class variable.
* +setDetails()
* This setter method will set or update the value of each Almuni Employment Details class variable

**Almuni class:**



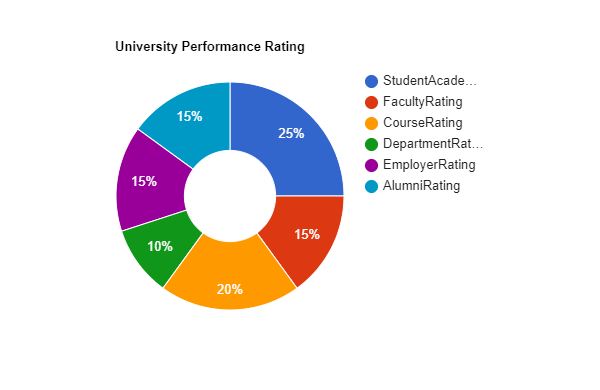
Attributes:

* + alumnifirstName:String
* +alumnilastName:String
* +alumniemailAddress:String
* +alumnicontactNo:Int
* +alumnigender:String
* +alumninationality:String
* +alumniaddress:String
* +alumnidegreename:String
* +alumniRating:ArrayList - Almuni will rate on the scale of 0 to 5 on the basis on overall experience.

Operations:

* +getDetails()
* This getter method will get the value of each almuni class variable.
* +setDetails()
* This setter method will set or update the value of each alumni class variable

**Performance Report class:**



Attributes:

* +schoolRating:ArrayList - School rating is internally generated.
* +departmentRating:ArrayList – Department rating is internally generated.

Operations:

* +getcourseRating()
* This operation will return the course rating.
* +getfacultyRating()
* This operation will return the faculty rating.
* +getschoolRating()
* This operation will return the school rating.
* +getdepartmentRating()
* This operation will return the department rating.
* GenerateUniPerformanceGraph()
* getDetails()
* This getter method will get the value of each performance report class variable.
* +setDetails()
* This setter method will set or update the value of each performance report class variable

**Performance Calculation class:**

Operations:

* +calculateSchoolRating():
* This operation will calculate the school rating.
* +calculateDepartmentRating():
* This operation will calculate the department rating.

**WEIGHTAGE FOR UNIVERSITY RATING CALCULATION:**

Through the following chart, the performance analysis can be done:

