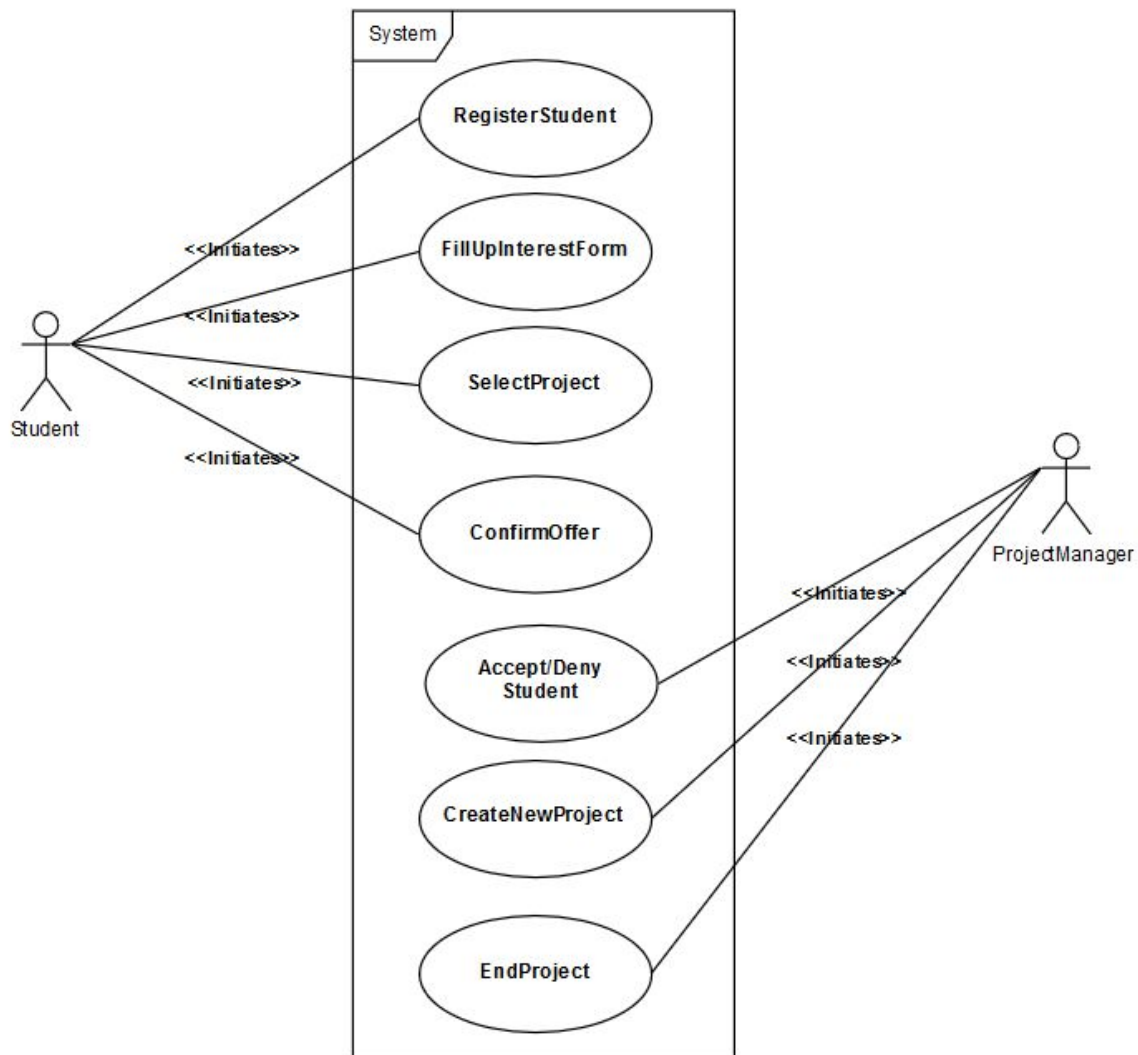**Assignment 2: Computer Science 6905**

Tian Wang
Lucas Critch
Carlos Salcedo
Ruben Chevez

VISION
GRADREC (Graduate Recruitment) is a web-based system that enables faculty members to advertise Master/PhD positions, provide information about their research projects and the availability of financial aid. Our goal is to become a world leader at connecting high skilled researchers with available working positions.
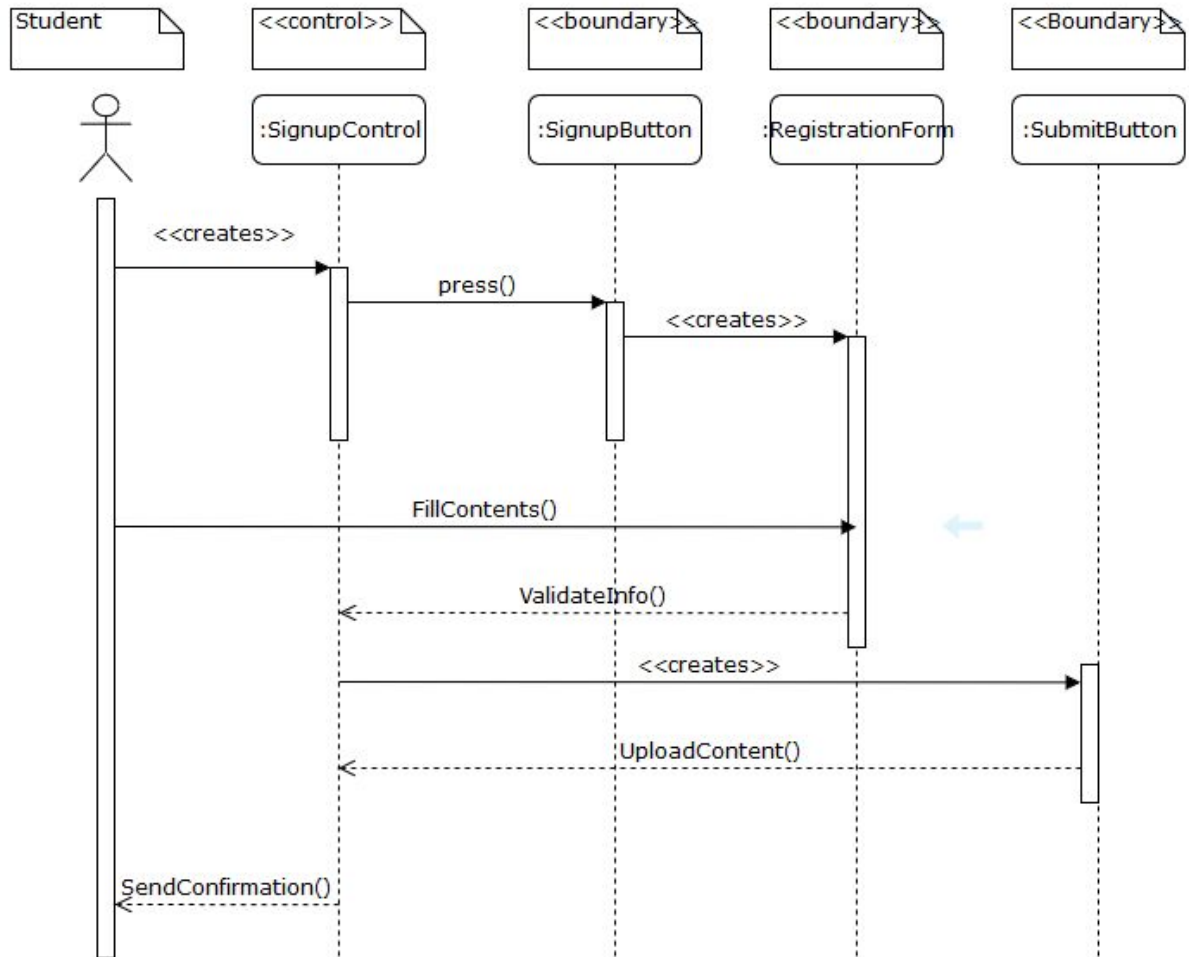
USECASES-DIAGRAM

USECASE_REGISER_STUDENT

Use Case:	*RegisterStudent*
Participating Actors:	*Student*
Entry Conditions:	*Student* enters the university's website and decides to enroll in a Program

Flow of Events	1. The S*tudent* selects the sign-up option in the University's website.
	        2. The system displays a form to the S*tudent.* The system asks for personal / registration information.
	3. The *Student* fills out the Registration_Form and presses the submit option.
	        4. The system adds stores the student's registration information.
	        5. The system sends an email confirmation
	6. The student will receive and confirm the email

Exit Conditions	● Student Registers Successfully ● Student will now be able to sign in.
Quality requirements	● Student sign-ups before a Deadline ● Email is not already used

REGISTER_STUDENT_OBJECTS

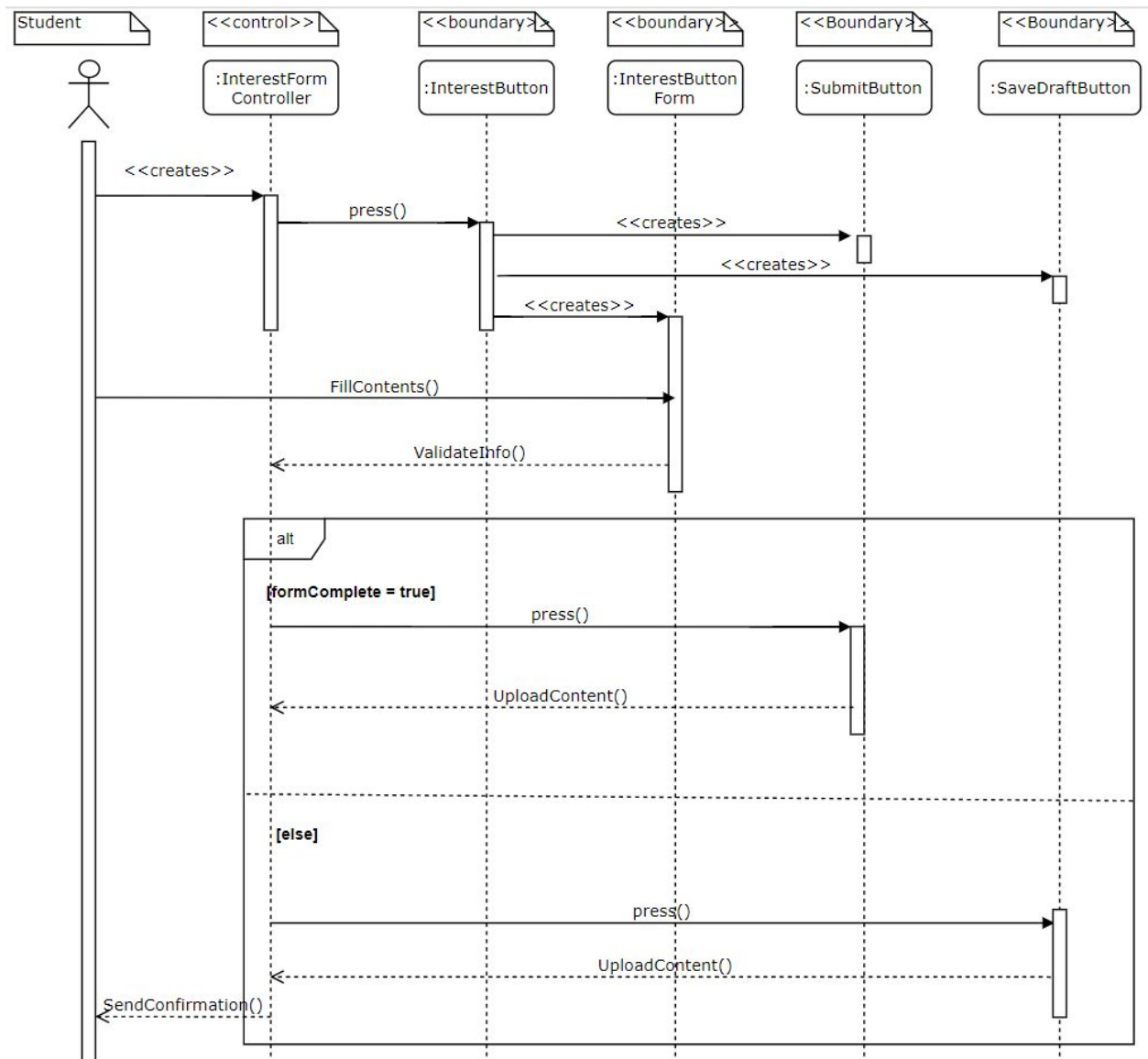| **ENTITY OBJECTS**	
Student	The potential candidate student to participate in the MSc or PhD program
Registration_Information	The student's registration information, which is filled out in the registration form.
List_of_Students	List of potential applicants to the university's Programs
BOUNDARY OBJECTS	
SignUpButton	Button which allows the user to begin the sign up process
SignUpForm	A form with the student's relevant contact and personal information, such as: date of birth, full name, nationality, gender, address, email, phone number, and dependants.
SubmitButton	Button which allows the user to upload the RegistrationForm to the system after the form has been filled out.
CONTROL OBJECTS	
SignUpControl	Creates the registration form, enables the user to fill out his/her information, validates that the mandatory information requirements are filled, and confirms the successful storage of the student's information.

REGISTER_STUDENT_SEQUENCE_DIAGRAM

Student	<<control>>	<<boundary>>	<<boundary>>	<<Boundary>>
	:SignupControl	:SignupButton	:RegistrationForm	:SubmitButton

<<creates>>

press()

<<creates>>

FillContents()

ValidateInfo()

<<creates>>

UploadContent()

SendConfirmation()

USECASE-FILL_UP_INTEREST_FORM

Use Case:	*FillUpInterestForm*
Participating Actors:	*Student*
Entry Conditions:	● *Student* is signed in the GradRec Website ● Student has not completed and submitted the InterestForm

Flow of Events	1. Student selects the fill InterestForm option.
	      2. The systems displays the Student's Personal InterestForm, the SaveDraftButton and the SubmitButton. The InterstForm would be empty if it is the first the Student is accessing this option, otherwise, the form would include the information from a previous draft.
	3. Student begins to fill out the interest form. Interest form includes the program the student wishes to enter to, academic strengths, and research areas of interest, among other questions.
	4. If the student is done with the form, the student presses the submit button
	      5. The system stores the Students information
	      6. The system sends a confirmation notice to the student indicating that the information has been saved.
	7. If the student wishes to continue at a later time, he/she presses the SaveDraftButton
	      8. The system stores the current information in the Student's Personal InterestForm as a draft for the Student to revisit next time he selects the Fill InterestForm option.
	      9. The system sends a confirmation notice to the student indicating that the information has been saved as a draft.

Exit Conditions	● The *Student*'s InterestForm is successfully saved and submitted ● OR The student's InterestForm is saved as a draft for later review
Quality requirements	● The submit button will only work if enough information has been filled out, notifying the user of any fields that still need to be completed.

FILL_UP_INTEREST_FORM_OBJECTS

ENTITY OBJECTS	
Student	The potential candidate student to participate in the MSc or PhD program
Educational_Information	The student's registration information, which is filled out in the registration form.
BOUNDARY OBJECTS	
FillInterestForm_Button	Button which allows the user to fill up his interest form.
Interest_Form	A form with the students relevant educational information, such as: educational background(University Diplomas or highest degree obtained),

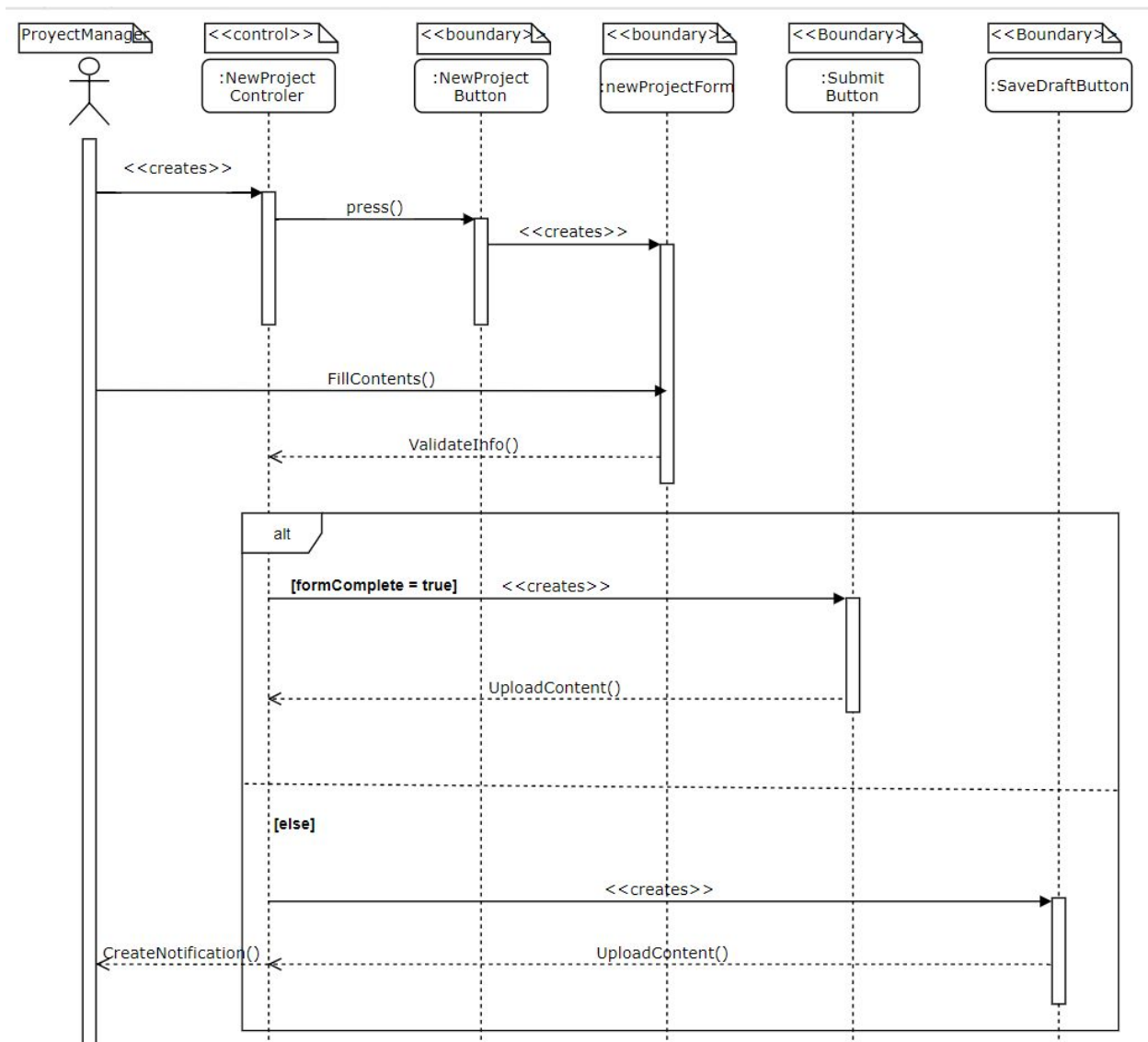|	
	resume, *research interests*, *skills*, published papers, projects, self-evaluation, character keywords.
SubmitButton	Button which allows the user to upload the Registration_Form to the system.
SaveDraftButton	Button which allows the Student to save his Registration as a draft..
CONTROL OBJECTS	
InterestFormControl	Creates the Interest form, enables the user to fill out his/her information, validates that the mandatory information requirements are filled, and stores the student's information.

FILL_UP_INTEREST_SEQUENCE_DIAGRAM

Student	<<control>>	<<boundary>>	<<boundary>>	<<Boundary>>	<<Boundary>>
	:InterestForm Controller	:InterestButton	:InterestButton Form	:SubmitButton	:SaveDraftButton

<<creates>>

press()

<<creates>>

<<creates>>

<<creates>>

FillContents()

ValidateInfo()

alt

[formComplete = true]

press()

UploadContent()

[else]

press()

UploadContent()

SendConfirmation()

USECASE-CREATE_NEW_PROJECT

Use Case:	*CreateNewProject*
Participating Actors:	*ProjectManager*
Entry Conditions:	*ProjectManager* is logged into the University's Website

Flow of Events	1. The *ProjectManager* selects the CreateNewProject option.
	     2. The system displays a Form. The Form asks for a title, abstract, MSc or PhD, Field of Study (Computer Science, Chemistry, Biology, Etc.), Specific area of Study (biomechanics, machine learning, Genes, etc.), Invited Coworkers and a few characteristics that a student should have to enter this project.
	3. The *ProjectManager* begins to fill out the Form.
	    4. If the primary fields (ProjectManager, Title, PhD or MSc, and ProjectDescription) are filled out the system displays the option to store the current project as a draft or as a new project.
	5. If the ProjectManager selects the save draft option
	    6. The system stores the current project's information in a temporary location as a draft for the ProjectManager to revisit at a later time.
	7. If the ProjectManager selects the submit project option
	    8. The system stores the project's information.
	    9. The system provides the project manager with a confirmation that the project has been created.

Exit Conditions	● The list of projects is successfully updated with the *ProjectManager's* new project
Quality requirements	● Certain fields in the application must be mandatory ● A *ProjectManager* can only have a certain amount of Projects at a time

CREATE_NEW_PROJECT_OBJECTS

ENTITY OBJECTS	
ProjectManager	The person in charge of creating the new project
Project_Information	The project's basic information, which can be stored temporarily as a draft, or can be saved to the system as a new project.
BOUNDARY OBJECTS	
NewProject_Button	Button which allows to the professor to begin the creation of a new project
NewProjectForm	A form with the new projects relevant information, such as: degree(PhD or MSc), field(biology, chemistry, Computer science, etc.), title, abstract, collaborators, financial aid available, prospective students(skills, such as Java, Lab experience, etc.), start date, projected end date, final product, etc.

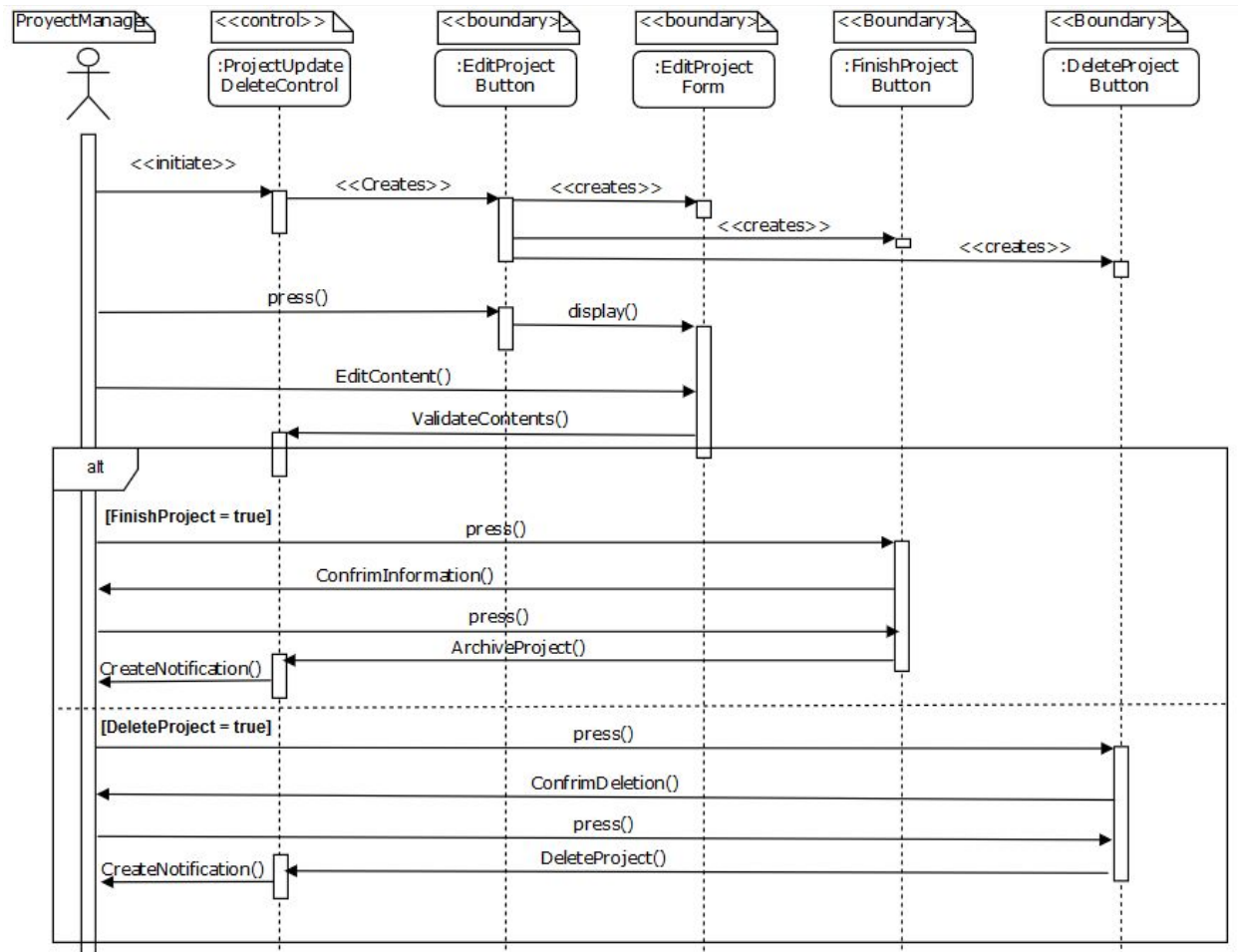|	
SubmitButton	Button which allows the user to store the NewProjectForm in the system after a minimum number of fields from the NewProjectForm have been filled out.
DraftButton	Button which allows the user to store the NewProjectForm in the system in a temporary location after a minimum number of fields from the NewProjectForm have been filled out.
CONTROL OBJECTS	
ProjectCreationControl	Creates the NewProjectForm, enables the user to fill out the project's information, validates that the mandatory information requirements are filled, and confirms the successful storage of information.

CREATE_NEW_PROJECT_SEQUENCE_DIAGRAM

USECASE_END_PROJECT

Use Case:	*EndProject*
Participating Actors:	*Project Manager*
Entry Conditions:	*Project Manager* is currently logged in the system

Flow of Events	1. The Project Manager presses the EditProjectButton, which is displayed next to each project in the UserHomePage. The UserHomePage initially shows all the available projects in which the Project Manager is a participant. A participant can be the owner of the project(Project Manager), or a contributor(student, professor, or other interested parties).
	
	      2. System Displays the ProjectInformation in an EditProjectForm. The EditProjectForm contains all the information filled out in the CREATE_NEW_PROJECT use case, and the following additional fields: 1) Conclusion, 2)Results, 3)FinishDate, 4) IsProjectFinished, 5)FinishProjectButton and 6) DeleteProjectButton. (*Other fields and buttons will also be included in this Form, like lessons learned, special thanks, UpdateProject button, etc, but those are not relevant to the current use case*).
	
	3. If the Project Manager clicks on the DeleteProjectButton
	
	      4. The System asks for an additional confirmation to make sure that the Project Manager wants to delete the project.
	
	5. The Project Manager confirms that she/he wants to delete the project from the list of projects, essentially deleting the project permanently from the system
	
	      6. The system removes the project from storage, and sends a notification that the project has been deleted
	
	7. If the project manager presses the FinishProjectButton
	
	      8. The system verifies that the fields 2), 3) and 4), have been filled out and asks the Project Manager to confirm that he/she wants to Archive the project
	
	9. The Project manager confirms that the project is over and no more future modifications will be done
	
	      10. The system stores the project in the archives, removing it from the list of possible project that students can apply for, and disabling any future modifications on the project. The system will essentially store the project for historical purposes only, and send a notification that the project is now in the archives of GradRec.

Exit Conditions	● The project is deleted from the list of projects, or ● The project is archived after confirming that the project has now ended.
Quality requirements	● The system will always ask for confirmation before deleting or finishing the project.

END_PROJECT_OBJECTS

ENTITY OBJECTS	
ProjectManger	The person in charge of managing the project.
UserProfile	Each user has a profile that includes their personal information and the projects they are a part of, or the projects that they are applying for. Based on the UserProfile, the contents of the HomePage will vary whenever a user logs into the system. For example, the project manager and a student may both view the same project in their home page, but only the project manager will have the EditProjectButton available in his/her homepage.
ProjectInformation	The information related to the project. This information will include all relevant aspects of the project as they were filled out in the CREATE_NEW_PROJECT use case and may at some point also include any other additional information that the project manager or a contributor may have submitted.
BOUNDARY OBJECTS	
UserHomePage	The first page the user sees when he logs into the system. The contents of this page will vary from user to user according to their profile.
EditProjectButton	The button will display the EditProjectForm, the FinishProjectButton and the DeleteProjectButton.
FinishProjectButton	The button allows the ProjectManager to archive the project, meaning the project is finished and that no more modifications will be made to it. The project will also be removed from the list of available projects.
DeleteProjectButton	The button allows the ProjectManager to delete the project from storage.
EditProjectForm	The EditProjectForm contains all the information as it was filled out in the CREATE_NEW_PROJECT use case. However, it will now include the additional fields needed to document the conclusion of a project.
CONTROL OBJECTS	
ProjectUpdateControl	Enables the user to fill out the project's new information, validates that the mandatory information requirements are filled out, and updates the project's information or deletes it from storage.

END_PROJECT_SEQUENCE_DIAGRAM

USECASE_SELECT_PROJECT

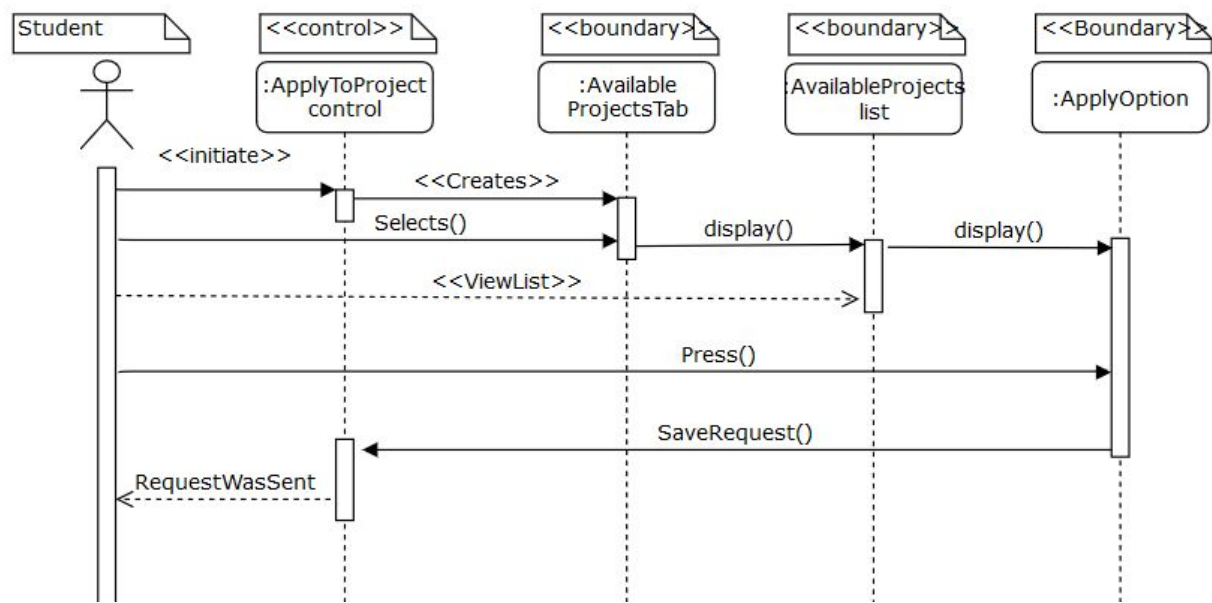| Use Case:	Select Project
Participating Actors:	*Student*
Entry Conditions:	* *Student* is currently logged into the system. * The student has already filled out the InterestForm from the *FillUpInterestForm* use case. * The student hasn't been rejected to all possible projects he/she may be eligible for.

Flow of Events	1. The student selects the ViewAvailableProjects option in his/her UserHomePage. 2. System displays a list of available projects based on the user's profile information, and next to each project, an ApplyOption that the user may select if he/she is interested in participating in that project. 3. The student selects the ApplyOption next to a project of his/her interest. 4. The system stores the request, saving it for later review by the corresponding Project Manager. 5. The system sends a notification to the student saying that his/her request has been received.
Exit Conditions	● The student receives a notification that his/her request has been sent
Quality requirements	● Regardless of the number of eligible projects, the student will only be allowed to have a limited number(determined by the system) of pending requests at a time.

SELECT_PROJECT_OBJECTS

ENTITY OBJECTS	
Student	The student interested in participating in a research project.
UserProfile	Each user has a profile that includes their personal information and the projects they are a part of, or the projects that they could or are applying for. Based on the UserProfile, the contents of the HomePage will vary whenever a user logs into the system.
BOUNDARY OBJECTS	
ViewAvailable Projects	This is an option in the student's home page, which contains the list of possible projects the student may apply to.
AvailableProjectsList	A list of all the available projects the student may apply to. If there are no matches, a message will be displayed telling the user that he/she may not apply to the program, and to reapply next semester, or to apply to a different academic program.
ApplyOption	This option allows the student to apply to a project.

CONTROL OBJECTS	
ApplyToProjectControl	Enables the student to view, apply for projects, and to store the student's project requests.

SELECT_PROJECT_SEQUENCE_DIAGRAM

USECASE_ACCEPT/DENY_STUDENT

Use Case:	Accept/Deny Student
Participating Actors:	*Project Manager*
Entry Conditions:	* The ProjectManager is logged into the system. * A *Student* has sent a request to participate in one of the ProjectManager's Project.

Flow of Events	1. The ProjectManager decides to view his notifications and selects the ViewNotifications option in his UserHomePage.
	      2. System displays a list of the ProjectManager's notifications.
	3. The ProjectManager selects the Student's ProjectParticipantRequest.
	      4. The system displays the ViewStudentProfile option, an AcceptStudent option and a RejectStudent option.
	5. If the ProjectManager selects the AcceptStudent option
	      6. The system stores the **Acceptance Offer**, for the student to view later and confirm that he/she will participate in the project
	      7. The system displays a message indicating that a request has been sent to the student to confirm his participation in the project
	8. If the ProjectManager selects the RejectStudent option
	      9. The system removes the Student's request from the ProjectManager's notification list.
	      10. The system removes the Project from the Student's List of available projects.
	      11. The system sends a notification to the student saying that his/her request has been denied.

Exit Conditions	● The system stores the ProjectManagers' Accept or Reject decision so that the student may view it later and take action accordingly. ● The ProjectManager accepts the student and an acceptance offer for the student is created, or ● The ProjectManager rejects the student
Quality requirements	● To make a better decision, the project manager will have the option to view the Student's Profile before deciding to accept or reject a student.

ACCEPT/DENY_STUDENT_OBJECTS

ENTITY OBJECTS	
ProjectManager	The person in charge of managing the research project.
Notifications	Notifications are the messages the system must sent to the users of the system. However, every user will only have access to view his/her notifications.
BOUNDARY OBJECTS	
ViewNotifications	This is an option in the User's HomePage, as all users in the system will eventually have notifications.
ProjectParticipant Request	A special type of notification, since it requires additional interaction from the user. This particular notification, includes the Accept, Reject and View Student profile options.
AcceptStudentOption	This option allows the ProjectManager to accept a student's request to participate in the project.
DenyStudentOption	This option allows the ProjectManager to reject a student's request to participate in the project.
ViewStudentProfile Option	This is an optional button in the use case, that allows the Project Manager to view the student's profile. It will allow
CONTROL OBJECTS	
AcceptDenyStudent Control	Enables the Project Manager to view the notifications option in his/her user profile, and accept or deny a student's request. It will also display an option to view the Student's Profile.

USECASE_CONFIRM_OFFER

Use Case:	Confirm Offer
Participating Actors:	*Student*
Entry Conditions:	* The ProjectManager has accepted the student's request to participate in the Project. * The student is logged into the system. .
Flow of Events	1. The Student decides to view his notifications and selects the ViewNotifications option in his UserHomePage. 2. System displays a list of the Student's notifications. 3. The Student selects the ProjectAcceptanceOffer Notification. 4. The system displays the AcceptOffer option and the DeclineOffer option. 5. If the Student selects the AcceptOffer option 6. The system adds the student as a participant of the Project, and send a notification to the project manager indicating that a new participant has been added to the project. 7. All other pending offers and project applications the student may have are removed from the system. 8. The system displays a message indicating that the Student is now an official participant of the Project. 9. If the Student selects the DeclineOffer option 10. The system removes the Project from the Student's List of available projects. 11. The system sends a notification to the Project Manager saying that the student will not be participating in the project.
Exit Conditions	● The Student becomes a participant of the Project, or ● The Student declines the offer to participate on the Project.
Quality requirements	●

CONFIRM_OFFER_OBJECTS

ENTITY OBJECTS	
Student	The student interested in participating in a project

|	
Notifications	Notifications are the messages the system must send to the users of the system. However, every user will only have access to view his/her notifications.
BOUNDARY OBJECTS	
ViewNotifications	This is an option in the User's HomePage, as all users in the system will eventually have notifications.
ProjectAcceptanceOffer Notification	A special type of notification, since it requires additional interaction from the user. This particular notification, includes the Accept and Decline offer options.
AcceptOffer Option	This option allows the Student to confirm his interest in participating in a project, and would automatically make the student a participant in the project.
DeclineOffer Option	This option allows the Student to decline his participation in a project.
CONTROL OBJECTS	
AcceptDeclineOffer Control	Enables the Student to view the notifications option in his/her user profile, and confirm or decline his acceptance to participate in a project.

DESIGN GOALS

- The Web application will be implemented using Node.js framework
- The end-user will access the system through a web browser on a computer or mobile device via a web page
  - *The web page will be responsive*
  - The user will be able to save some information as drafts before submitting it to the system
- User Profiles and Project Information will be stored by the system.
- User passwords will be encrypted, and access to profile information will be limited.
- Contact information will only be shared between participants if they are all accepted into the same project.
- The contents of a Participant's home page should include:
  - The project he/she is participating it
 - An edit personal profile option

- A view notifications option
- Sign-out option
- **The system will have encrypted connections and no significant data must will be sent via e-mail by the system.**
- **The system will be self-explanatory**

OBJECT MODELLING

All students will have a one to one relationships with their registration form and their interest form, as this forms can only be submitted once. There is a possibility to save the Interest Form temporarily as a draft, but for the purpose of matching students to projects later on during development, the system will only take into account submitted InterestForms. The following diagram displays the relationship between the Student and the forms.
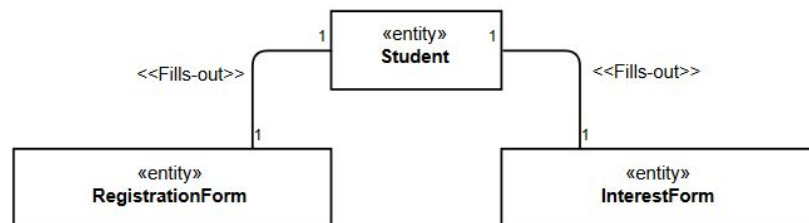
The primary actors participating in the system will be Students and Professors. However, each user will have different roles in the creation and participation of the Projects. Only professors will be able to create and manage projects as Project Managers, while Students will only be able to participate as collaborators in the Project.
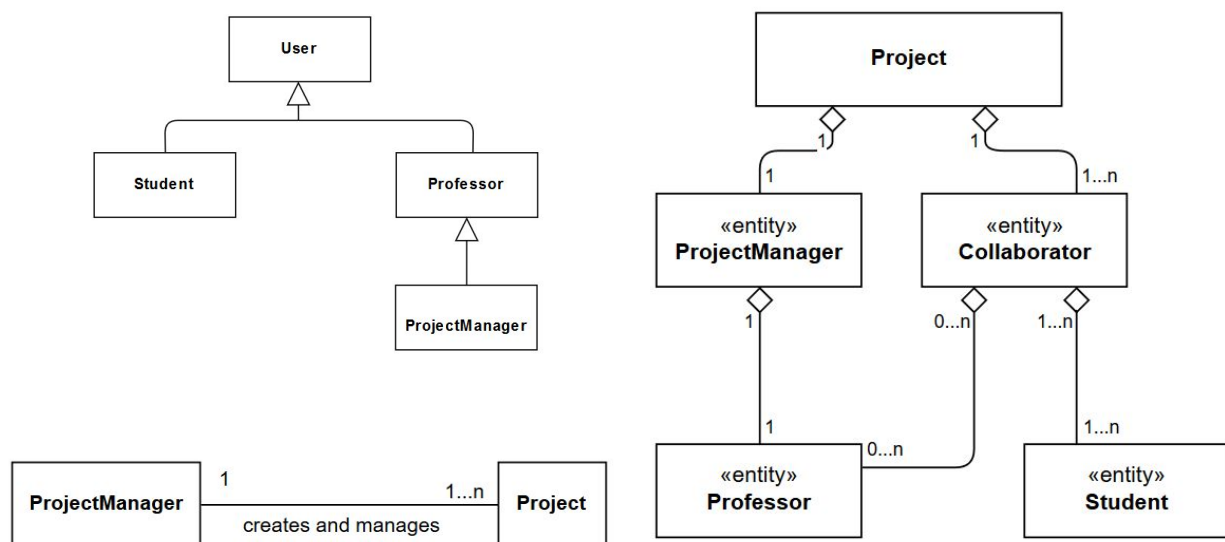
SUBSYSTEM DECOMPOSITION
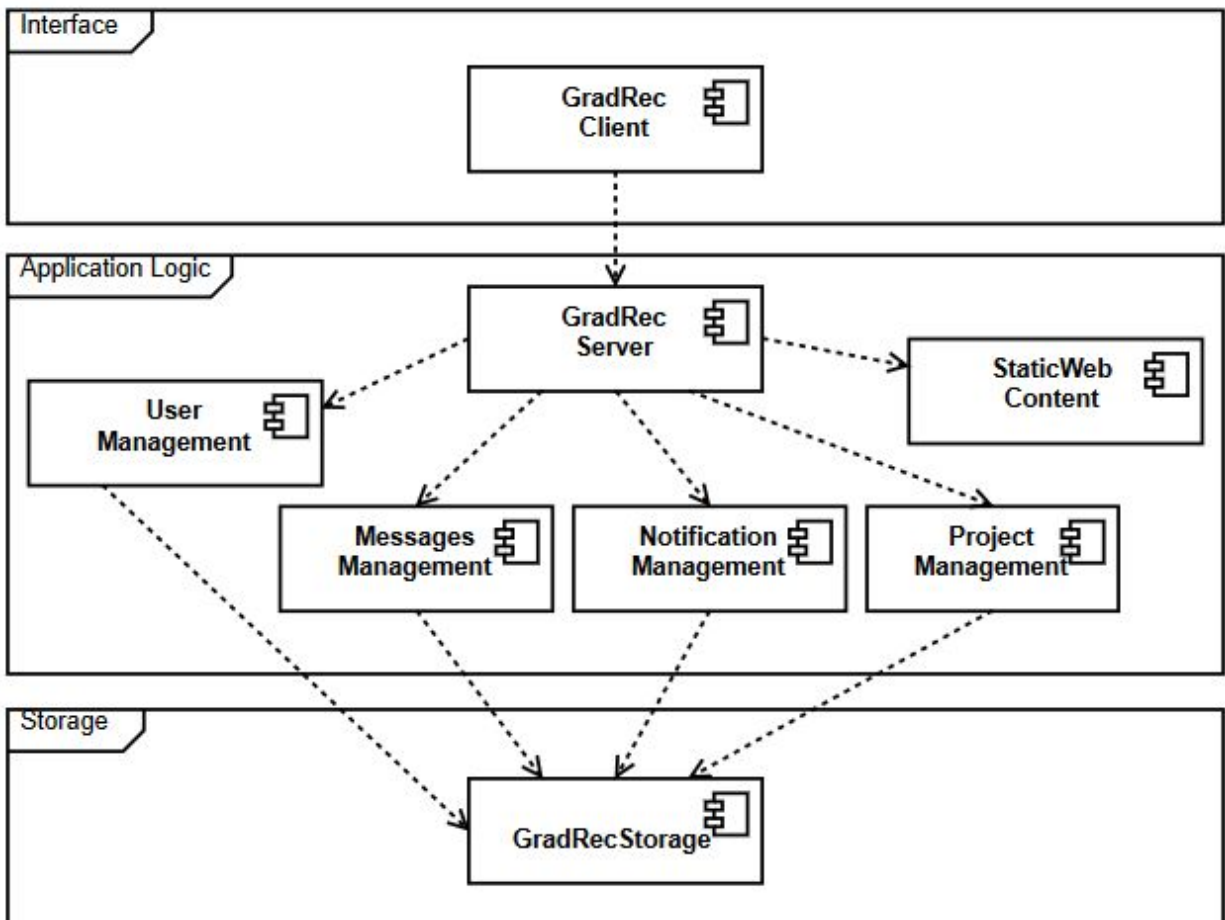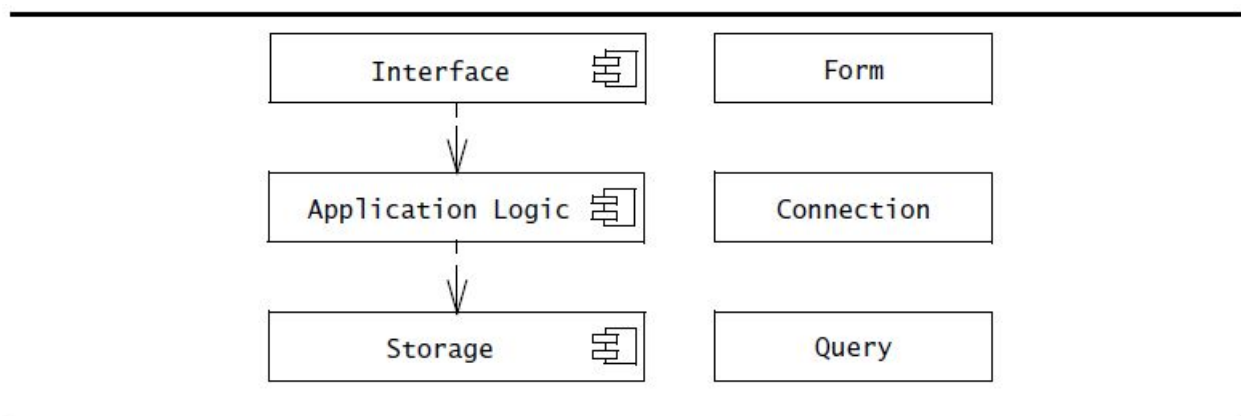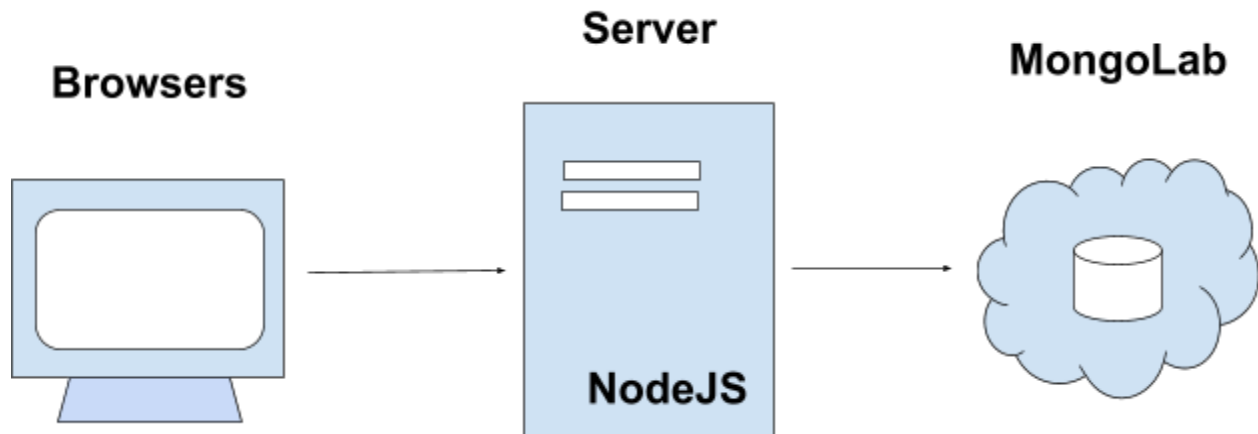
SYSTEM ARCHITECTURE

Figure 6-22   Three-tier architectural style (UML component diagram). Objects are organized into three layers realizing the user interface, the processing, and the storage.

DEMO and SOURCE CODE

Source Code: https://github.com/rubencg195/GRADREC/blob/master/index.html

Demo Web Page: http://sc-1.cs.mun.ca/