

Group 9 – Graduates Admission

May 25th, 2014
Iteration 2

Members

Joshua Cohen-Collier
Oluwakorede Ogunyinka
Mohammed Zeinallah

*NOTE, this project version is an extension of the work done by Group 4 for the first submission

Original Group Members

Samson Truong
Kais Hassanali
Kyle Jackson
Spencer Sullivan

Extending Group Work

UML Class Diagram: Josh Cohen-Collier and Oluwakorede Ogunyinka
UML Sequence Diagram for Apply Use Case: Josh Cohen-Collier and Oluwakorede Ogunyinka
UML Stategraph Diagram: Josh Cohen-Collier
UML Activity Diagram: Mohammed Zeinallah
Formatting and text summarization: Josh Cohen-Collier

1. Introduction

The Graduate Admissions Project will be a system used by students (applicants) in order to apply for a graduate program (such as a Masters or PHD) at Carleton University.

2. Functional Requirements

2.1. System scope

- An applicant should be able create an application and be required to enter their personal information (name, contact information) and select the graduate program that they are interested in applying to
- An applicant should be able to upload supporting documents to the application (Transcripts, curriculum vitae (CV), reference letters)
- An applicant should be able to accept or decline an offer, if presented with one, or be allowed to withdraw their application
- The administration office should be able to match professors with applicants based on the research field of the professor, and add the professors to the application
- Professors should be able comment on and make recommendations to the application

- Professors should be able to allocate funds for the application
- The associate chair should be able to accept or reject an applicant into a graduate program
- The administration office should be able to create an offer of acceptance for the applicant
- All users (applicants, administration office, associate chair, professors) should be able to look over the entirety of the application
- All users (applicants, administration office, associate chair, professors) should be able to check the current status of the application and see what steps are needed for the application to continue

2.2. Actors

2.2.1.Applicants

Applies for the graduate program, sets up the application package to meet the requirements set out by the specified graduate program. Ensures that all required document and information is present and valid.

2.2.2.Administration office

Does the preliminary screening of the applicant's application and provided simple comments to the professor and associate chair. Also is tasked with ensuring that all applications are processed and reviewed by the professors. Also tasked with ensuring that sufficient professors have commented on the application so that it can be properly examined by the associate chair. Should gather all relevant applications and forwards them to the appropriate department.

2.2.3.Professor

Reviews the applications and provides a rating of the overall applicant and provided some key points for the associate chair to make a final decision.

2.2.4.Associate Chair

Reviews the application and professor's comments to make a final decision to the application.

3. Non-Functional Requirements

- The system should be interoperable with established Carleton services (Carleton Central, Carleton Banner)
- The system should allow for applications from the Ontario Universities' Application Centre (OUAC)

4. System Models

4.1 Scenarios

Scenario Name	<u>FrankApplies</u>
Participating Actor instances	<u>Frank:Applicant</u> <u>James:AdministrationOffice</u> <u>Bob, Jenny, Babak:Professor</u>
Basic Flow of Events	<ol style="list-style-type: none">1. Frank applies to the Graduate admissions office for a M.Eng in Software Engineering online and includes all pertinent documents in his application.2. James Receives Frank's application and reviews the important information. James makes note that Frank does not require funding and that Frank's desired research field is in Operating Systems.3. James sends Frank's applications with James' notes to all of the Professors with similar research fields which includes Bob, Jenny and Babak.4. Bob, Jenny and Babak are notified that they should now begin to make recommendations on Frank's application.

Scenario Name	<u>JamesReviews</u>
Participating Actor Instances	<u>Paul:Applicant</u> <u>James:AdministrationOffice</u>
Basic Flow of Events	<ol style="list-style-type: none"> 1. Paul applies to the Graduate Admissions Office for a M.Eng in Mechanical engineering. 2. James receives Paul's application and upon reviewing finds that Paul is missing one of the required documents. 3. James sends a notification to Paul informing Paul that he is missing a required document and asking Paul to re-submit his application. 4. James changes the status of Paul's application to incomplete.

Scenario Name	<u>BabakRecommends</u>
Participating Actor Instances	<u>Frank:Applicant</u> <u>James:AdministrationOffice</u> <u>Babak:Professor</u>
Basic Flow of Events	<ol style="list-style-type: none"> 1. Babak receives Frank's application from James. 2. Babak reviews the application and gives Frank a rating of 4/5 and recommends that he be accepted and provided full funding. 3. Babak notes that he personally does not have the funding for Frank. 4. Babak submits his completed recommendations. 5. James is notified that Babak has completed his recommendations.

Scenario Name	<u>KhooReviews</u>
Participating Actor Instances	<u>Khoo:AssociateChair</u>

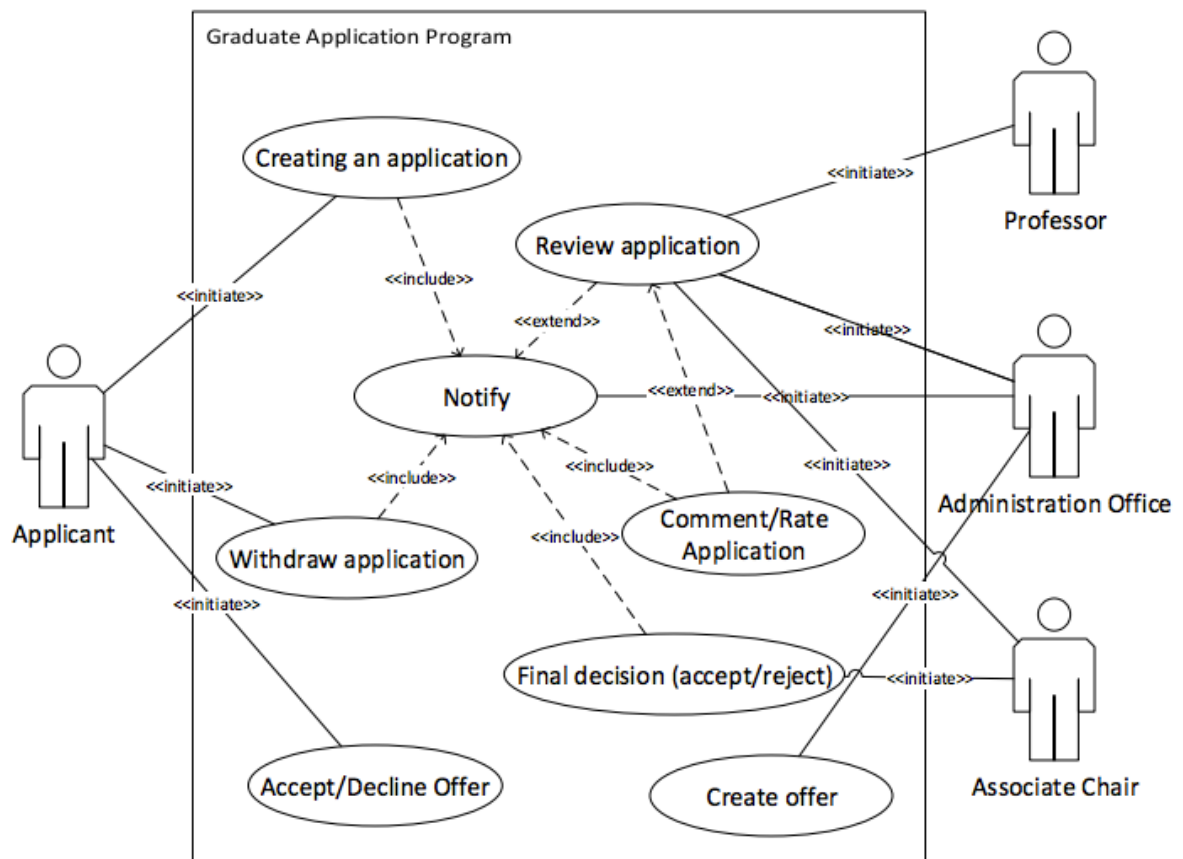
	<u>James:AdministrationOffice</u> <u>Babak, Jenny, Bob:Professor</u> <u>Frank:Applicant</u>
Basic Flow of Events	<ol style="list-style-type: none"> 1. Khoo receives a notification on his computer because Babak, Jenny and Bob have made recommendations on an Frank's application 2. Khoo reviews the recommendations which all suggest that the Frank should be accepted and offered full funding. 3. Khoo decides that Frank should be offered a position and provided with full funding with Jenny as his advisor. 4. Khoo prepares an offer of admission and sends the offer to James to be proofread. 5. Jenny is notified that Khoo is preparing an offer with her as Frank's advisor.

Scenario Name	<u>KhooMakesOffer</u>
Participating Actor Instances	<u>Khoo:AssociateChair</u> <u>James:AdministrationOffice</u> <u>Frank:applicant</u> <u>Jenny:Professor</u>
Basic Flow of Events	<ol style="list-style-type: none"> 1. James reviews the offer of admission and returns it to Khoo. 2. Khoo finalises his offer of admission to

	<p>Frank which includes researching with Jenny and being provided full funding.</p> <ol style="list-style-type: none"> 3. Khoo sends the formal offer of admission to Frank. 4. Jenny is notified that Frank has been issued an offer of admission with Jenny as his advisor. 5. James is notified that Frank has been officially sent an offer of admission.
--	--

Scenario Name	<u>FrankAccepts</u>
Participating Actor Instances	<u>Frank:Applicant</u> <u>Khoo:AssociateChair</u> <u>James:AdministrationOffice</u>
Basic Flow of Events	<ol style="list-style-type: none"> 1. Frank receives an offer of admission from Khoo. 2. Frank accepts the offer of admission. 3. James is notified that Frank has accepted. 4. James updates Frank's application status to accepted.

4.2 Use Case with Descriptions



Use Case Name	Creating an application	
Brief description	Applicant will apply for graduate studies using the system	
Precondition	The system is up and running and no maintenance is being performed Applicant is logged in to Carleton Central	
Primary Actor	Applicant	
Secondary Actor	None	
Dependencies to other use cases	Notify	
Basic Flow	Steps	
	1	INCLUDES USE CASE NotifyAdminOff
	2	Applicant clicks to apply for graduate studies
	3	Applicant enters in required personal information
	4	The system VALIDATES THAT all the information required has been entered and loads the page for the next step
	5	Applicant enters in required education information
	6	The system VALIDATES THAT all the information required has been entered and loads the page for the next step
	7	Applicant selects which field of study they wish to apply for and which area they would like to focus in
	8	The system VALIDATES THAT all the information required has been entered and loads the page for the next step
	9	Applicant must provide all documentation that the system requests at this point and time
	10	The system VALIDATES THAT all the information required has been entered and loads the page for the next step
	11	System give a confirmation page that shows the applicant his/her application
	Postcondition	The applicant's application has been created and notification has been sent to administration office Application is set to default to be reviewed by "ADMINOFF"
Bounded alternative flow	BFS 4, 6, 8, 10	
	1	The system displays a prompt to notify the applicant that there is missing information/document
	2	The system will reopen the same form with previous field filled and direct the applicant to the missing fields/documents
	Postcondition	System will operate as intended

Use Case Name	Review application	
Brief description	Administration office reviewing/screen the application for the first time	
Precondition	The system is up and running and no maintenance is being performed The reviewer is logged into the systems as "administration office" role There is at least one active application	
Primary Actor	Administration office	
Secondary Actor	None	
Dependencies to other use cases	Notify	
Basic Flow	Steps	
	1	INCLUDES USE CASE NotifyAdminOff
	2	Administration officer selects an un-reviewed application to be reviewed
	3	System will show the reviewer the content that is to be screened by the reviewer
	4	Review will input any relevant comments and INCLUDES USE CASE NotifyProf
	Postcondition	Application is marked to be reviewed by "PROF" and is made available to be viewed by the professor role
Global alternative flow	If the reviewer spots an error/issue and selects the action to Notify Applicant THEN	
	1	A draft email is created to notify the applicant that his/her application

		has an issue.
	2	Reviewer will write his/her comments on the issue around the application
	Postcondition	System will send an email to the applicant and put the status of this application to be reviewed by "APPLICANT"

Use Case Name	Notify	
Brief description	Use to send a notification to a desired recipient	
Precondition	The system is up and running and no maintenance is being performed The reviewer is logged into the systems as “administration office”, “professor” or “associate chair” role There is at least one active application	
Primary Actor	Administration office, Professor, Associate Chair	
Secondary Actor	None	
Dependencies to other use cases	None	
Basic Flow	Steps	
	1	User selects Notify...(Prof, AdminsOff, AssociateCh) depending on which stage the application is at. (ie. Associate chair cannot be notified until 3-5 professors have commented on the application etc)
	2	The system will increment the applications progress
	3	The system will display the application in the users notification window for visibility
	4	System will VALIDATES THAT the notification is successful
	Postcondition	Application is now emphasized on the selected role. Status of the application is set to need attention from “PROF”, “ASSOCIATECHAIR”, “APPLICANT”, “ADMINOFF”
Bounded alternative flow	BFS 4	
	1	The system displays a prompt to notify the applicant that an error has occurred
	Postcondition	The system will recover and revert back to pre-notify state

Use Case Name	Final Decision	
Brief description	The concluding in the reviewing process	
Precondition	The system is up and running and no maintenance is being performed The reviewer is logged into the systems as “associate chair” role There is at least one active application with X comments/ rating form professors	
Primary Actor	Associate Chair	
Secondary Actor	Administration office	
Dependencies to other use cases	None	
Basic Flow	Steps	
	1	INCLUDES USE CASE NotifyAssociateCh
	2	Associate chair selects the application he/she wished to finalize
	3	System will show the general information of the application along with the professors inputs
	4	The associate chair will make a decision and INCLUDES USE CASE NotifyAdminOf
	Postcondition	Administration office is notified of a completed application and an offer can be sent to the applicant

		Status of application is set to "TO BE SENT TO APPLICANT"
Global alternative flow	If the associate chair has any questions related to the professors' comments or the information on the application itself THEN	
	1	System will prompt user to select whom to notify
	2	INCLUDES USE CASE Notify...(Prof, AdminsOff, AssociateCh)
	Postcondition	System will notify the selected recipients and put the status of this application to "CAUTION/WAIT"

Use Case Name	Accept/Decline Offer	
Brief description	Applicant will choose whether to Accept or Decline offer they receive	
Precondition	The system is up and running and no maintenance is being performed Applicant is logged in to Carleton Central	
Primary Actor	Applicant	
Secondary Actor	None	
Dependencies to other use cases	None	
Basic Flow	Steps	
	1	Applicant loads the page for the offer and examines the information displayed on the conditions of the offer
	2	Applicant will hit the accept or decline button based on their decision
	3	Applicant is prompted for their Carleton Central password to confirm their Identify of security reasons
	4	The system VALIDATES THAT all the password has been entered correctly and loads the page confirming that the their decision has been recorded
	Postcondition	The applicant's offer has been accepted and recorded in the system
Bounded alternative flow	BFS 4	
	1	The system displays a prompt to notify the applicant that the password entered is incorrect
	2	The system will reopen the same page with the prompt for the applicant's Carleton Central password
	Postcondition	System will operate as intended

5. User Interface

The user interface for the graduate admission will be similar to that of a program installation wizard (ex. Installing Microsoft Office). There will be steps that users will take. Each box in the diagrams below will represent a different step / view that the users will see. Each view will have the list of all the steps and beside each step there will be an icon that shows the status of each step. Each step will also have buttons to move to the next or previous step.

Example Step

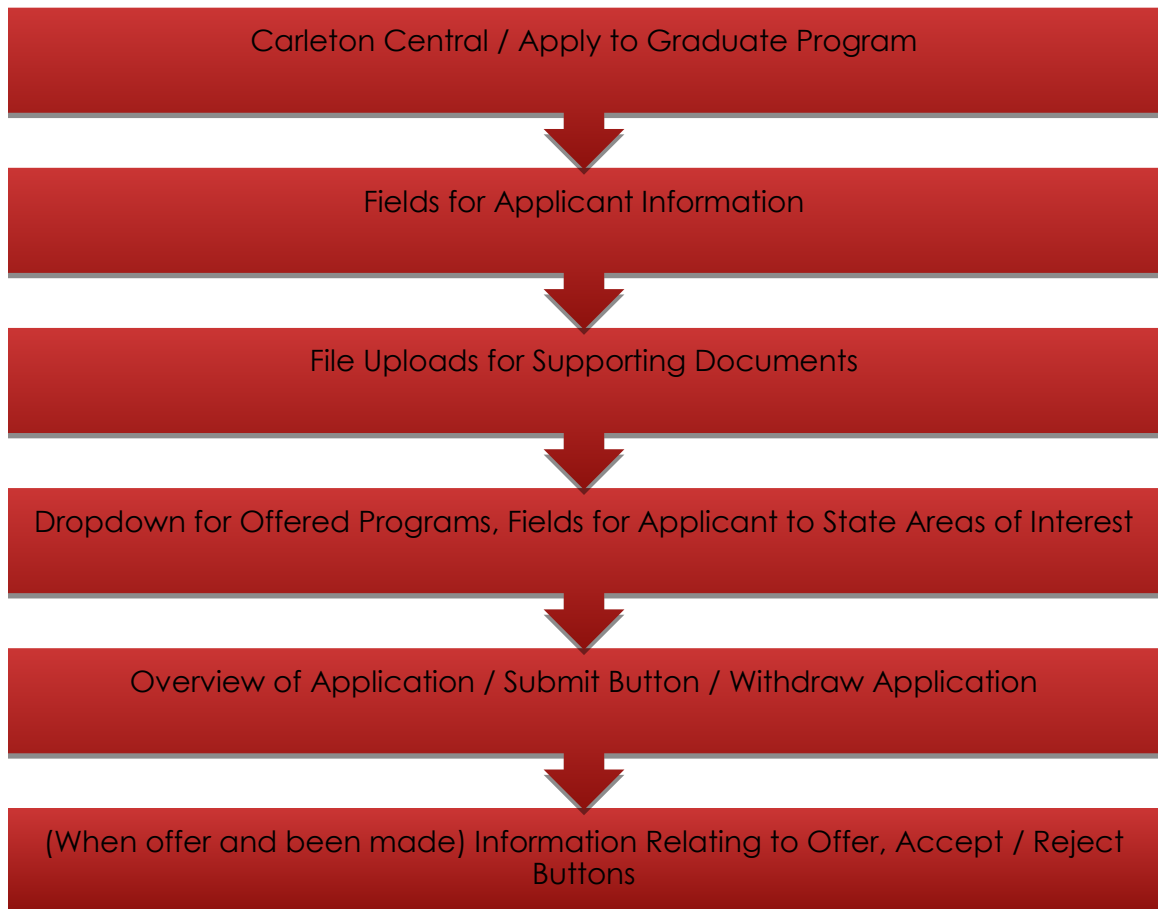
[illegible]

Applicant

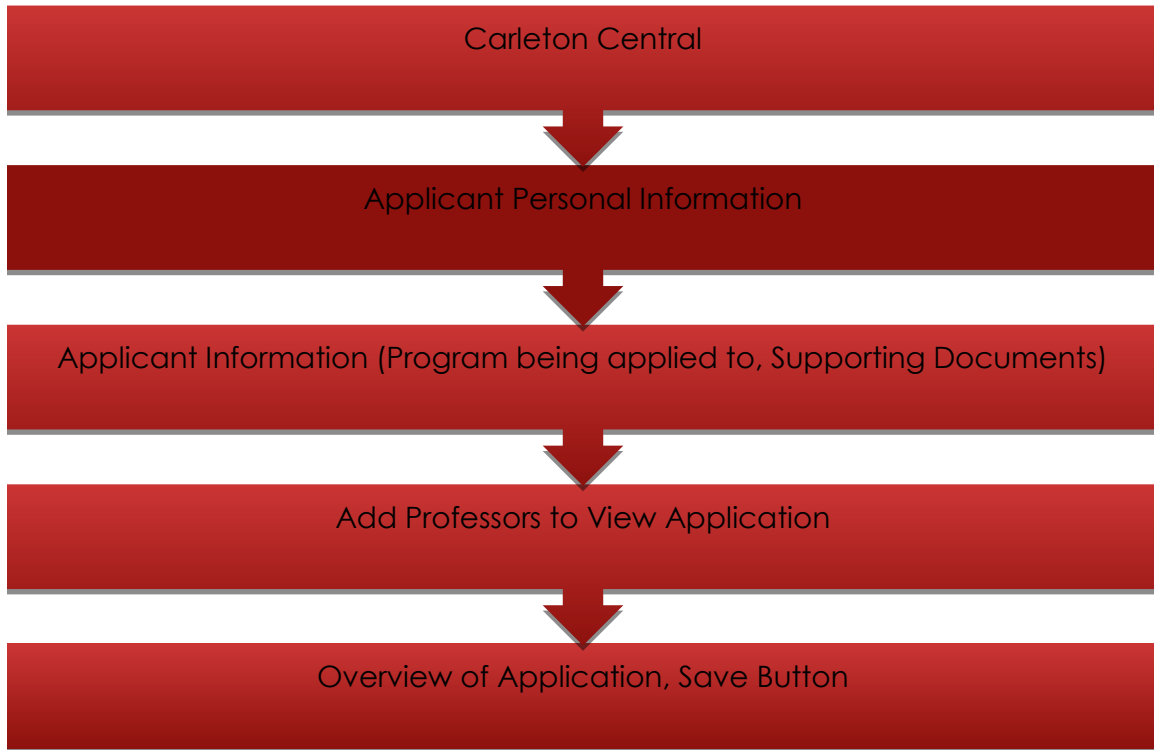


Student Online Applications

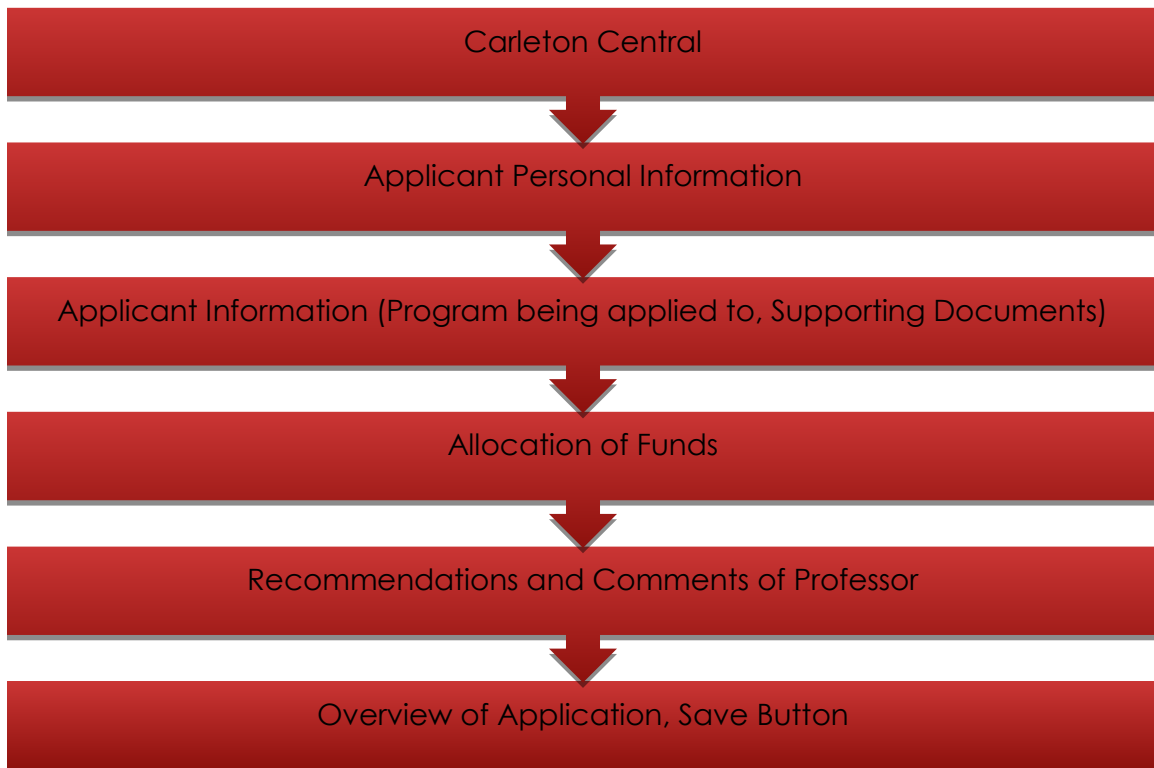
- Apply to Graduate
- Change Program Elements
- Request a Transcript/Certificate of Enrolment
- Undergrad TA Applications
- TA Mentoring Applications
- Letter Of Permission Applications



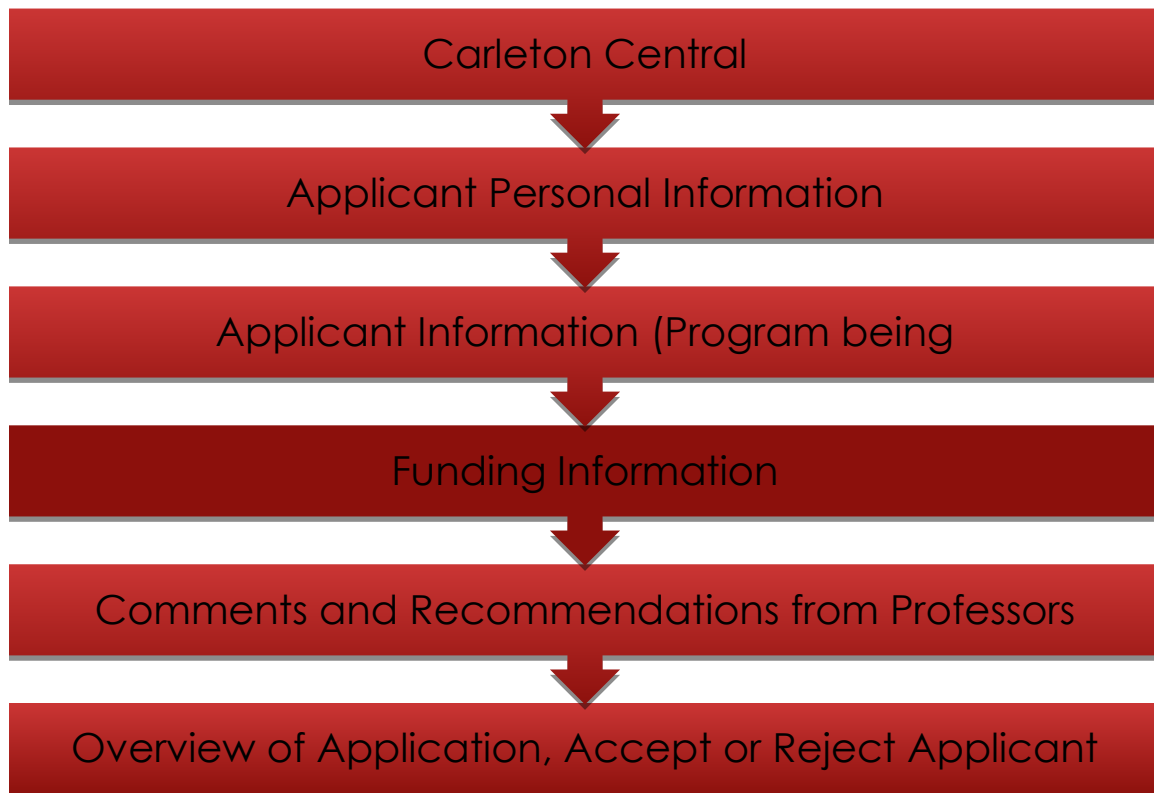
Administration Office



Professors



Associate Chair



As progress occurs on the application, different information may be available. For example: when professors have made their comments and recommendations, and have provided information related to funding, that information will become available to the Applicant, Admin Office, and Associate Chair. By having steps and different views for each step, this approach will allow for easier changes, such as addition more steps, and allows the information to be presented cleaner than before (in the currently implemented system, users are presented with all information on a single page which users have said is overwhelming and brings confusion by not clearly showing what needs to be done in order to move the application forward.

6. Risks and Open Issues

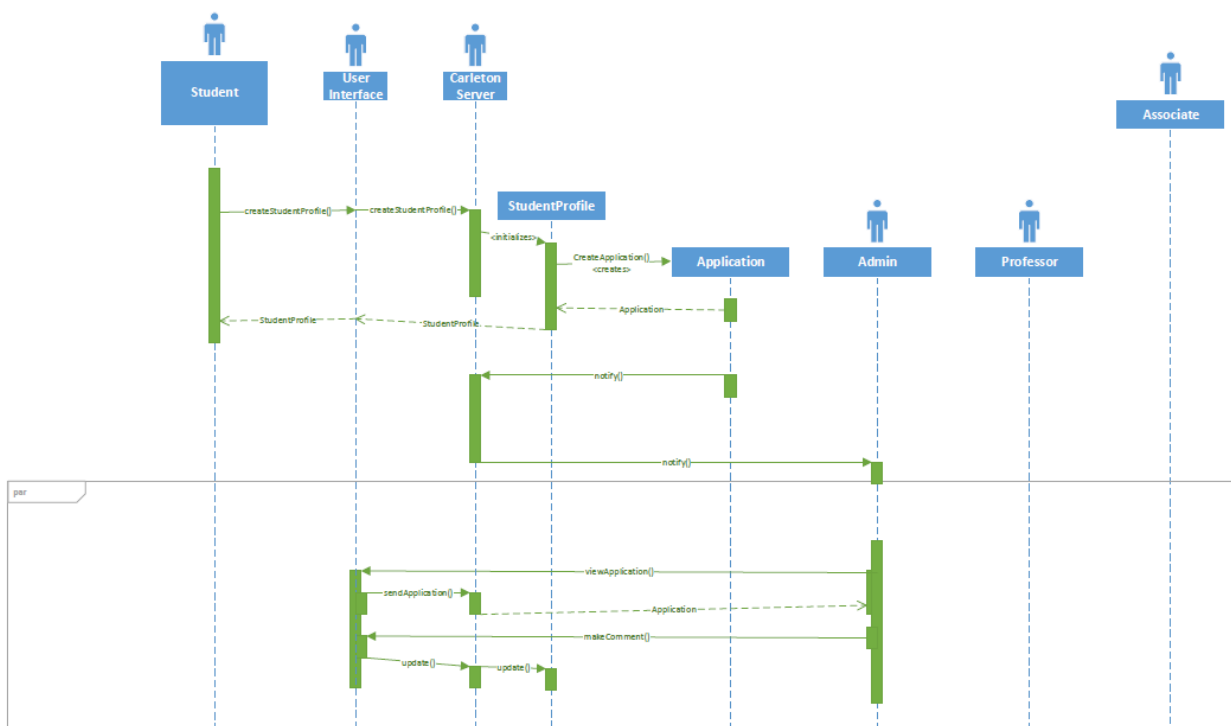
6.1 Risks

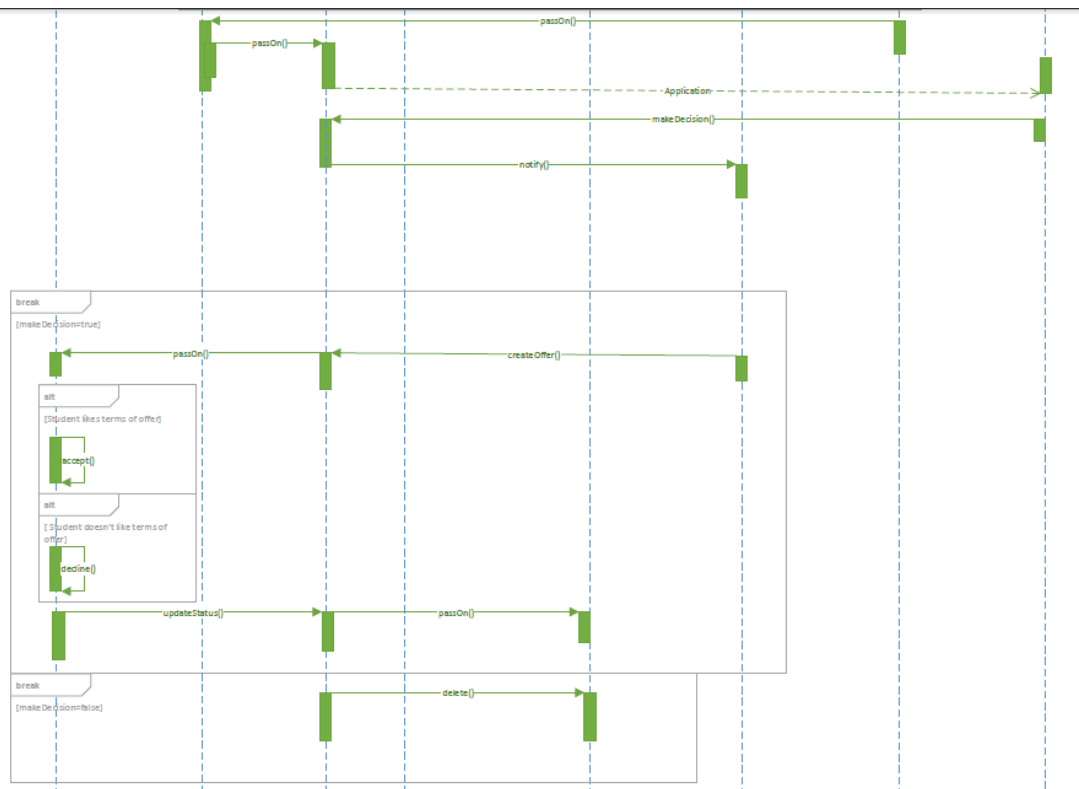
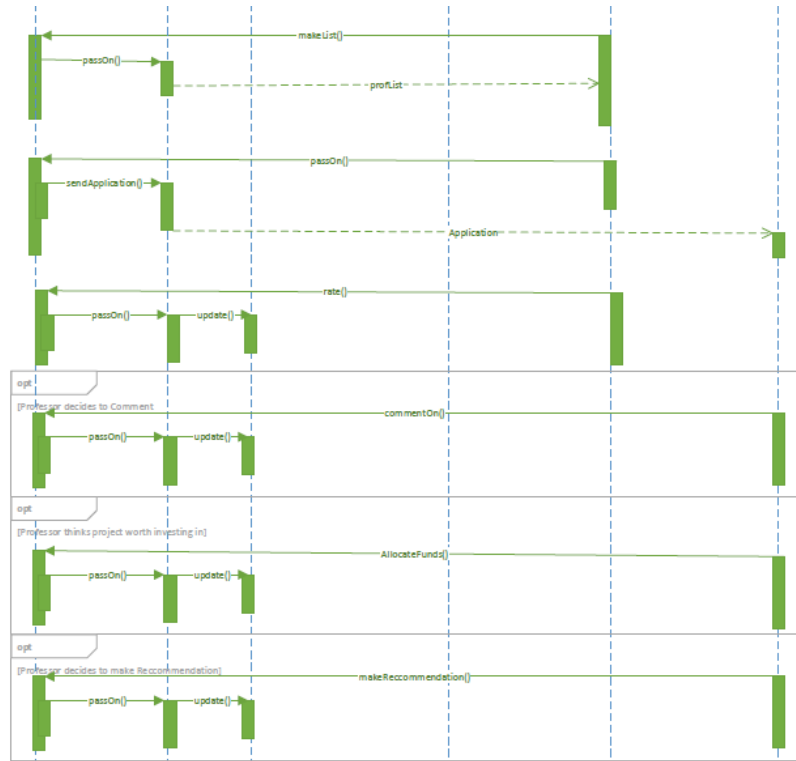
- Hacking – applicant information may be subject to theft depending on the security of the system, can cause severe issues for applicant and the credibility of the University.
- System Error – In the case of a system error, the system needs to come back up immediately or it will inconvenience the actors that require the system. If system is down for long period of time, may cause deadline issues for actors

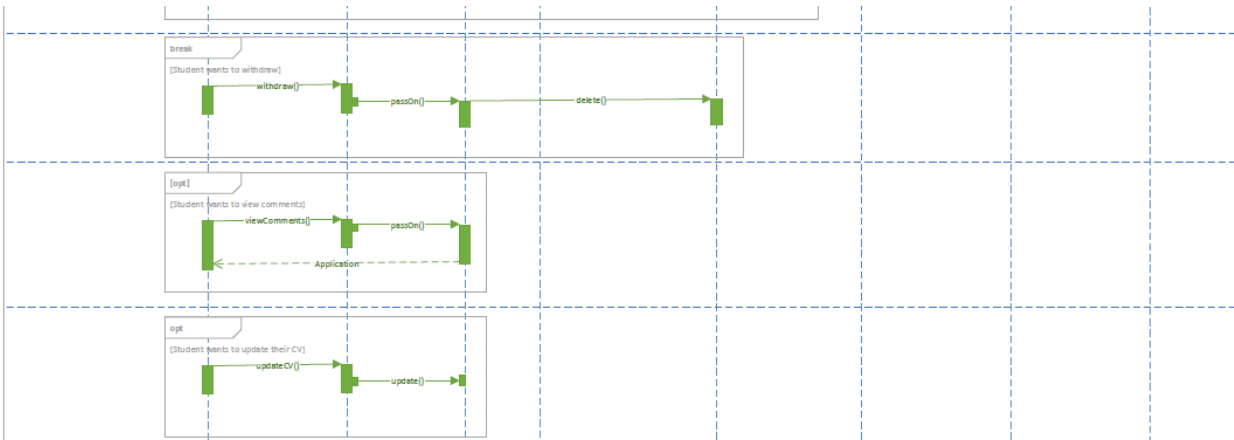
6.2 Open Issues

- Forgetting – Admissions Office, Professors, or Associate Chair forget to do what needs to be done to get an applications to the next step and applicant misses deadline. Some kind of a notification system might need to be implemented.
- Incomplete Application – Applicant fails to fully finish application and submit. Perhaps implement another type of notification system for applicants via their email.
- Storage – how information will be stored to reduce loss of data and security risks (as mentioned above). Perhaps backing up data?
- Traffic – System should be able to handle periods of high volume and ensure no loss of data and timely response to requests by actors.

7. UML Sequence Diagram for Use Case Apply





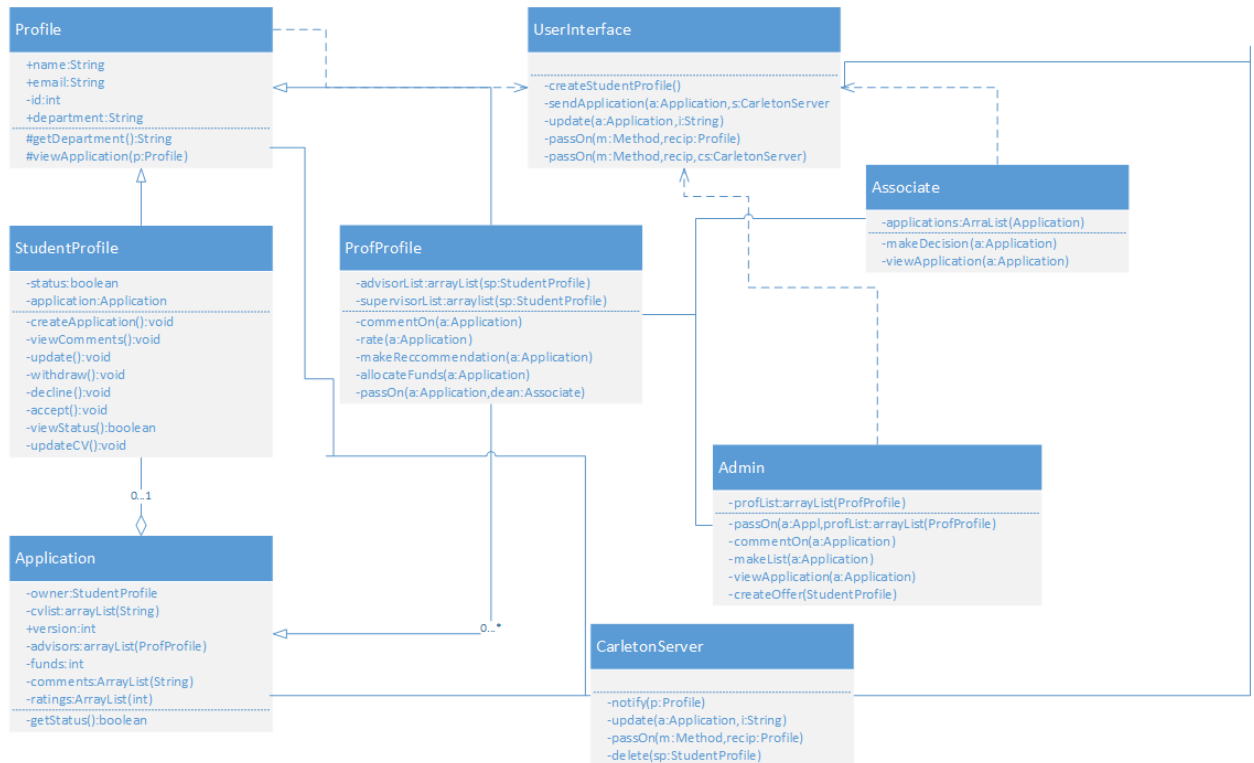


*Note, due to the massive size of the image it was necessary to split up into several smaller files, as a fully zoomed in image would render the size too big to allow for use in word processors.

The sequence diagram for the use case "Apply", where a graduating student applies for graduation, uses a variety of actors. The boundary class is the UserInterface, which allows the Student(Actor) to create their requests to be sent to the system. The control class is the CarletonServer, which relays all Student commands from the UI to the entity classes. In this fashion, the Student could pass commands such as withdraw(), and updateCV(). The other actors involved, which were Associate, Admin, and Professor, also used these classes to interact with the entities.

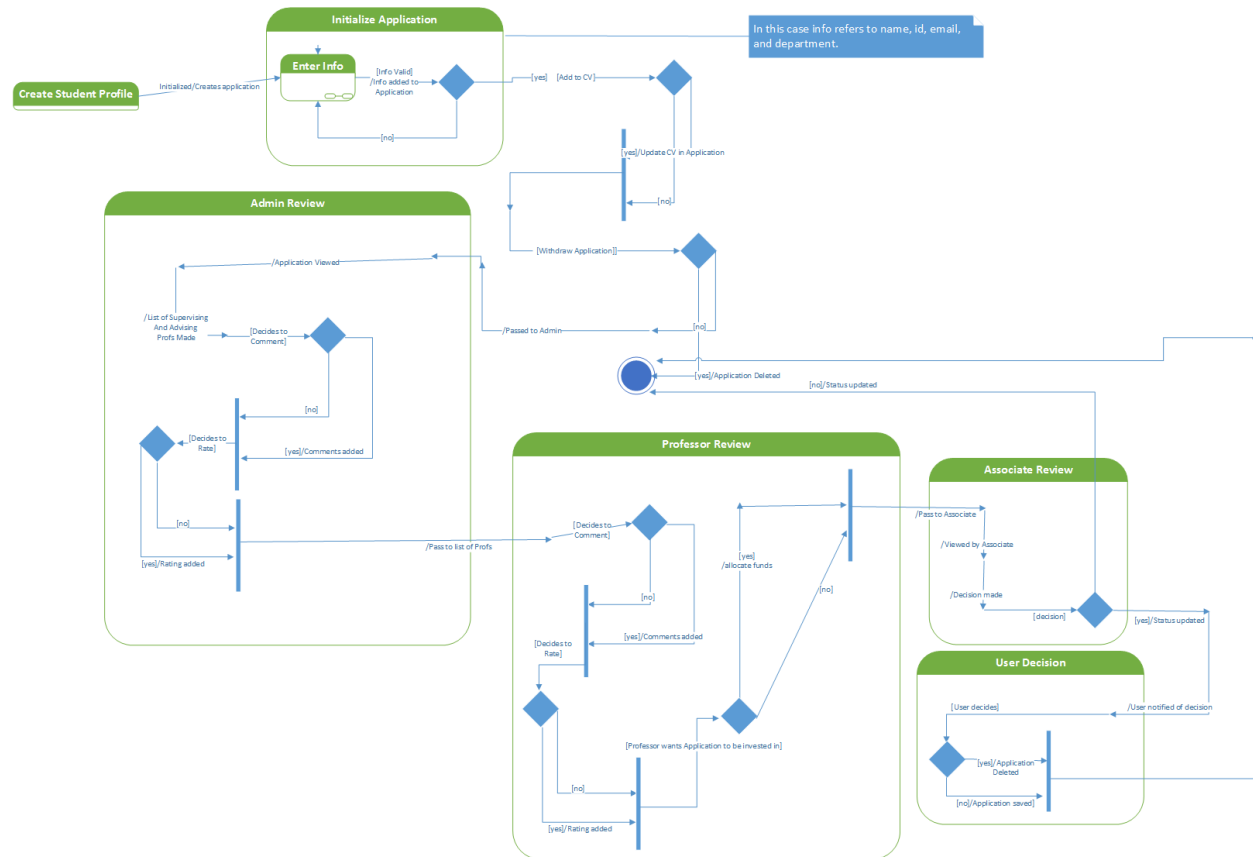
Note that the passOn() functions for the entity classes do not do the same thing as the same function in both control and boundary classes. For the entities, the Application is passed along to a recipient. For the other classes, the passOn() method actually passes the previously recieved message to a recipient. The control and boundary classes also make use of functions such as update(), and notify().

8. UML Class Diagram for System



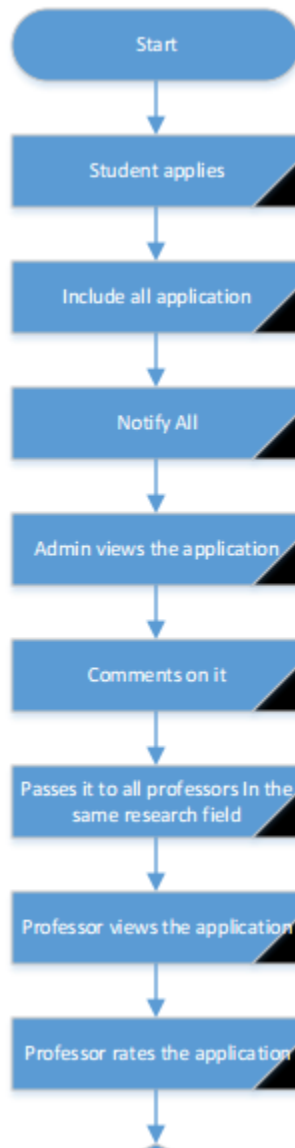
In this diagram, the actors (in this case Associate, Admin, StudentProfile, and ProfProfile), are each associated with the necessary boundary classes UserInterface and CarletonServer. Although both professors and students can have profiles, admins and associates can't, for several reasons. First, it was unnecessary for them to have a profile, as personal information is of no relevance to anyone but themselves. Secondly, it would be inadvisable for a public email address to be available, as they would likely be bombarded by requests from students, which is not desirable. Upon the construction of a StudentProfile through the UserInterface, a new Application is instantly created, which notifies the Admin, who can in turn use passOn() to get it to the Professor's profile, who uses the same method to get it to the Associate. All requests and inter-profile functions are performed using the CarletonServer, which keeps a database of each profile.

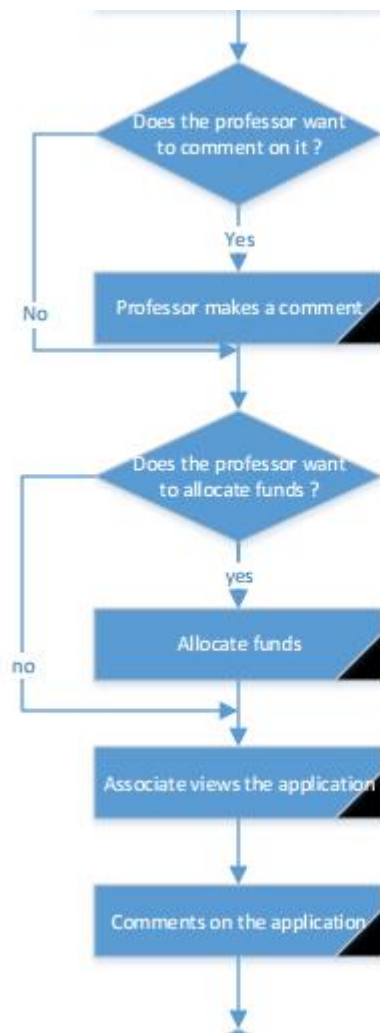
9. UML StateGraph Diagram for Application Object

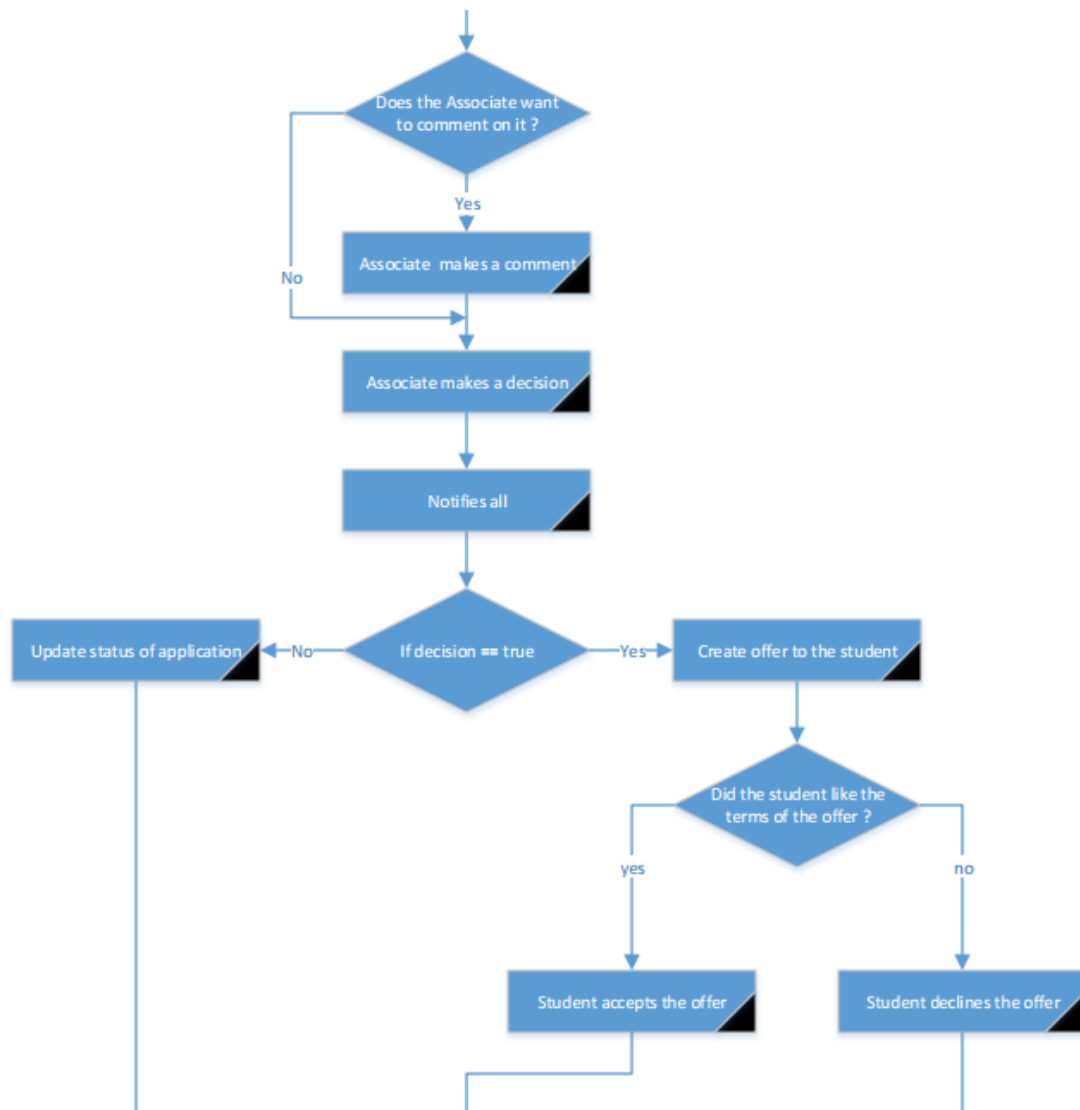


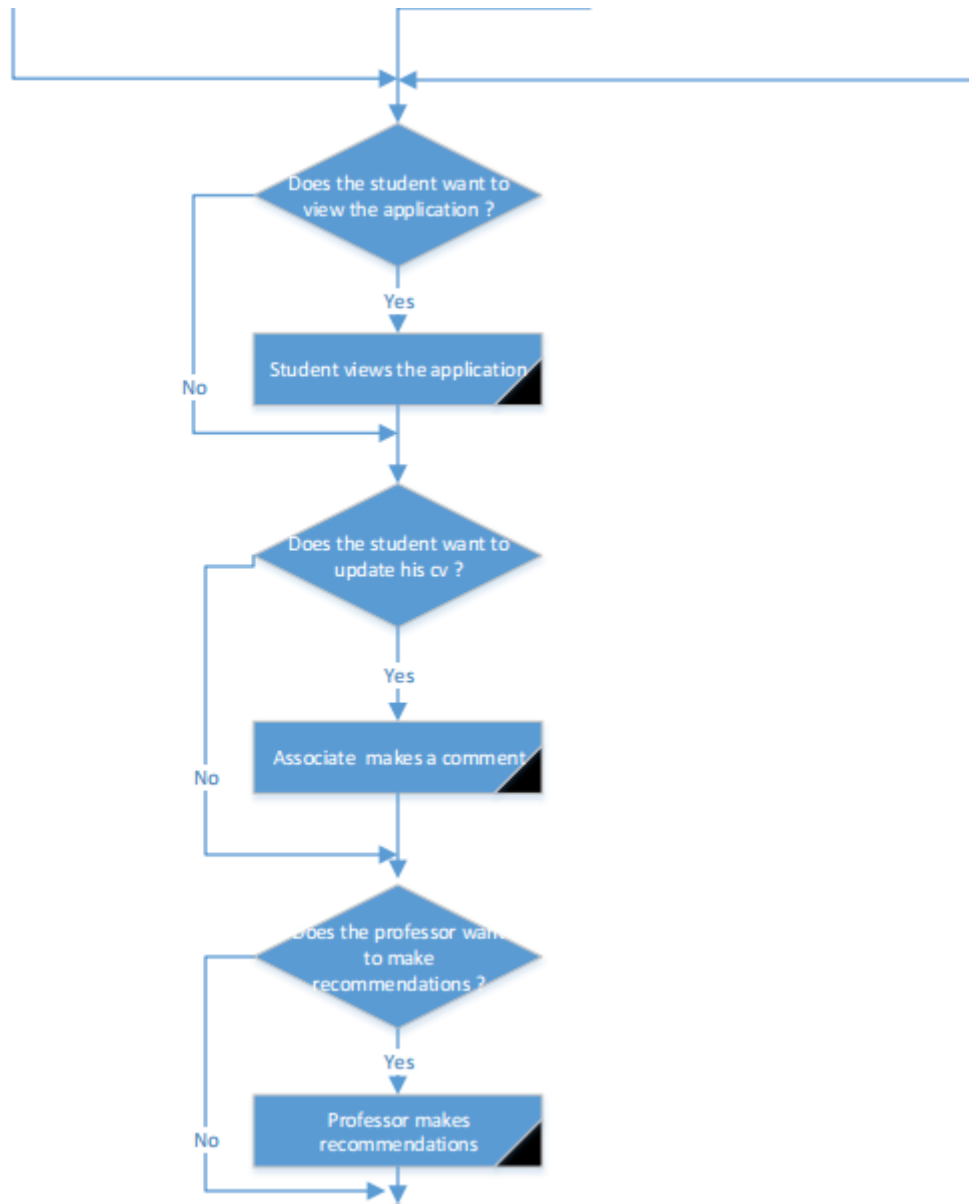
The state graph diagram shows the progression of the state of an Application object through its many possible stages. Each specific section includes a number of conditions and possible outcomes.

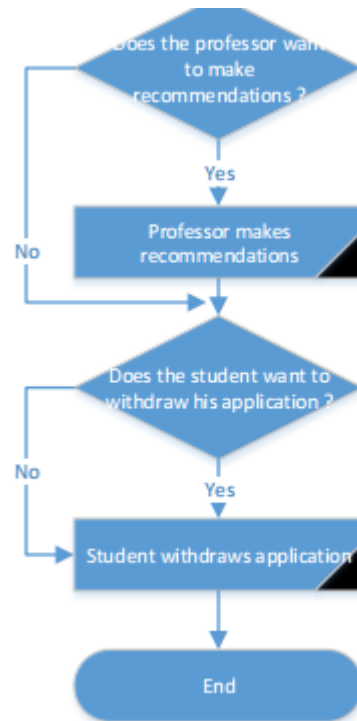
10. UML Activity Diagram for Application Object











The above diagram illustrates a general outline of the activity of an Application object from start to finish, with multiple possible paths.