Assignment 3

Name: Amr Shurman

Student ID: 100996742

Email: amrshurman@cmail.carleton.ca

GitHub repository: https://github.com/amrshurman/A3

Video Link: https://youtu.be/9F81sARojvw

Note:

- **In src/cucumber: You will find all valid path tests.**
- **In src/cucumber2: You will find all invalid path tests.**
- **In src/cucumber3: You will find testing for 2 students registering at the same time for a course with 1 available spot only.**

Corrections made to the system:

- Fixed the time functionality for the term (slowed it down properly)
- Made clerks have distinct property. We can now distinguish each clerk by the other. Each clerk is labeled with their own number, and this is previewed to the user upon running the system. For eg: (Welcome clerk no.1)
- Fixed the cap size functionality. When cap size is full, student cannot register in course.
- Fixed the drop course mechanic to actually be able to drop courses after registration period ends and before the term ends.
- Added a course fulfilment functionality (UC13). After the term ends, all students that finished the term while registered to a course get listed with their course. If they were registered to multiple courses, then all their registered courses will be outputted as well.

Note: I included all diagrams from this document in the folder “Diagrams” just in case.

Step 1:

UC0	
Title	The system sets up for the coming term
Participating actors	System
Pre-condition	Upon starting the system and before registration starts. Before any student or clerk actions can be performed.
Parameters	Input & output: Void
Steps	- The system should set up for coming term upon running the server. - All timers for deadline dates are set and start

UC1	
Title	The clerk logs into the ATC
Participating actors	Clerk
Pre-condition	After the system (server) is run.
Parameters	Input Password: String
Steps	- The system asks the user to enter their password. - 1) Clerk enters password ‘admin’ to successfully access the system or 2) enters wrong password and gets prompted to enter password again.

UC2	
Title	The clerk logs out of the ATC
Participating actors	Clerk
Pre-condition	After the clerk is logged in. Can be done at any time when prompted with the menu.
Parameters	Input logout: String
Steps	- Enter ‘log out’ in the input stream in order to exit the system

UC3	
Title	The clerk creates a course for the current term
Participating actors	Clerk
Pre-condition	Before the registration period starts
Parameters	Input courseDetails (title, course code, capsize, enforce prereqs? no. of midterms, no. of assignments, has final? is project course?): String
Steps	- The clerk chooses the create course functionality from the menu. - The system asks the clerk to input the course details (title, course code, capsize, enforce prereqs? no. of midterms, no. of assignments, has final? is project course?) that they want created. - Enter course details to create a course. - 1) Course is created if success input is provided. 2) Re-prompted if incorrect input given.

UC4	
Title	The clerk deletes a course for the current term
Participating actors	Clerk
Pre-condition	Before the registration period starts
Parameters	Input ‘course code’: String
Steps	- The clerk chooses the delete course functionality from the menu. - The system asks the clerk to input the course they want to delete. - 1) Enter course code of existing course to delete a course. 2) If course ID doesn’t exist in current list of courses for the term then re-prompt.

UC5	
Title	The clerk cancels a course for the current term
Participating actors	Clerk
Pre-condition	After registration period ends
Parameters	Input ‘course code’: String
Steps	- The clerk chooses the cancel course functionality from the menu. - The system asks the clerk to input the course they want to cancel. - 1) Enter course ID of existing course to cancel a course. 2) If course ID doesn’t exist in current list of courses for the term then re-prompt.

UC6	
Title	The clerk creates a student for the current term
Participating actors	Clerk
Pre-condition	Before registration period starts
Parameters	Input studentDetails (ID, name, fulltime/part time status): String
Steps	- The clerk chooses the create student functionality from the menu. - The system asks the clerk to input the student’s details that they want created. - Enter student details to create a course. 1) Student is created if success input is provided. 2) Re-prompted if incorrect input given.

UC7	
Title	The clerk deletes a student for the current term
Participating actors	Clerk
Pre-condition	Before registration period starts
Parameters	Input studentID: String
Steps	- The clerk chooses the delete student functionality from the menu. - The system asks the clerk to input the student they want to delete. - Enter student ID to delete existing student. 1) Student is deleted if existing student ID is provided. 2) Re-prompted if incorrect input given.

UC8	
Title	a student logs into the ATC
Participating actors	Student
Pre-condition	After the system is run.
Parameters	Input studentID & password: String
Steps	- The system asks the user to input their password.

	- Student enters their student ID and password in order to access the system. 1) Successful login if correct details. 2) Re-prompt if not.

UC9	
Title	a student logs out of the ATC
Participating actors	Student
Pre-condition	After the system is run. Can be done at any time when prompted with the menu.
Parameters	Input ‘log out’: String
Steps	- Student enters ‘log out’ to the system in order to be logged out.

UC10	
Title	A student selects and registers in one course of the current term
Participating actors	Student
Pre-condition	For selecting a course: Before registration period ends. For registering a course: After registration period starts and before registration period ends.
Parameters	Input course code: String
Steps	- The student chooses the select course functionality from the menu. - The system asks the student to input the course they want to register in. Student enters course code to select course. 1) If course code exists, then the course gets added to a list of possible choices of courses that can be registered. 2) If course code doesn’t exist, then system re-prompts. - Student then selects the register course functionality from the menu. - Student then enters course code again to register for the course. 1) If course code exists, then the registration is successful. 2) If course code doesn’t exist, then system re-prompts.

UC11	
Title	A student selects and deregisters from a course
Participating actors	Student
Pre-condition	For deregistering a course: After registration period starts and before registration period ends.
Parameters	Input course code: String
Steps	- Student selects the deregister course functionality from the menu. - Student then enters course code to deregister for the course. 1) If course code exists, then deregistration is successful. 2) If course code doesn’t exist, then system re-prompts.

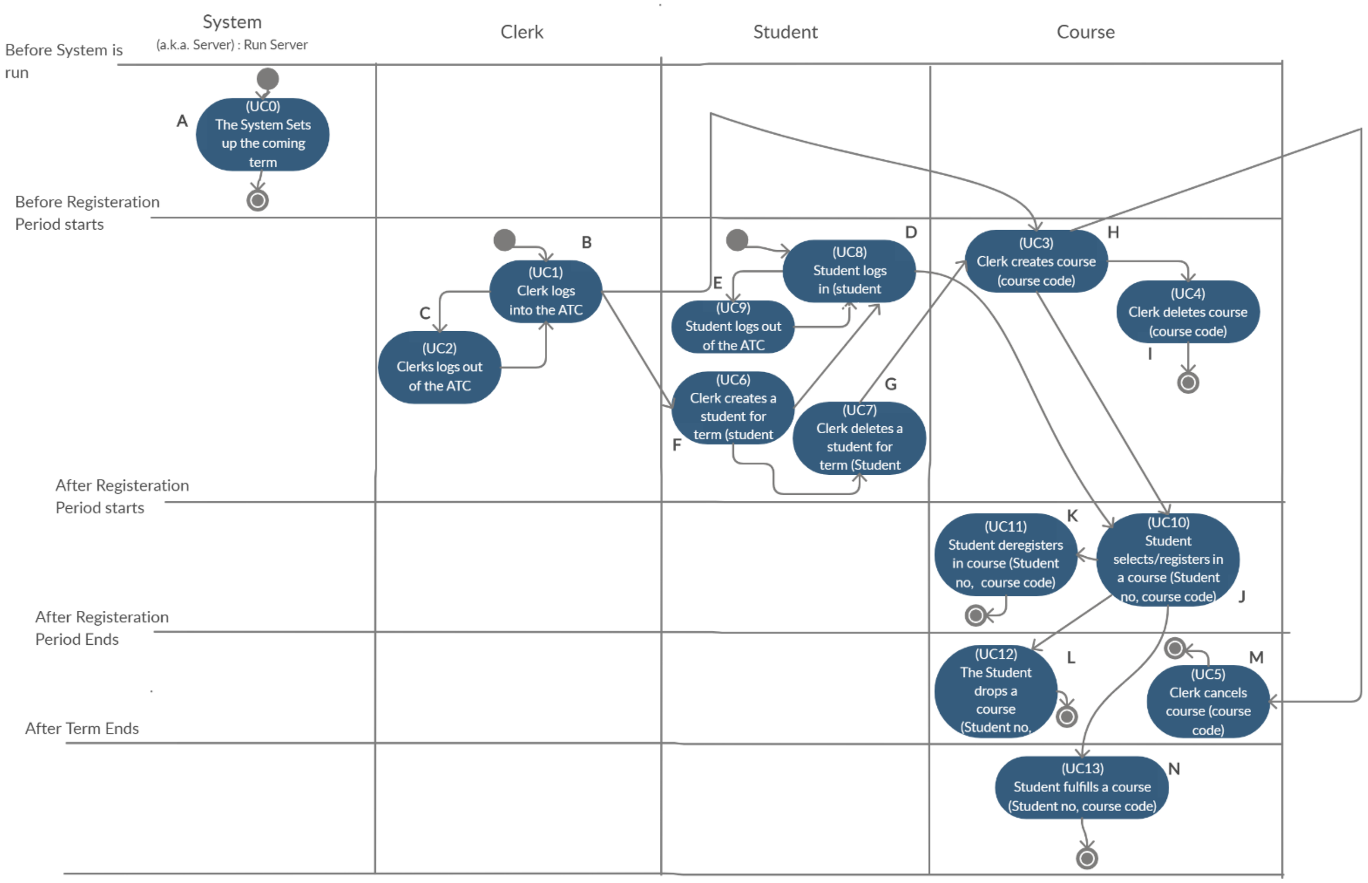
UC12	
Title	A student selects and drops a course
Participating actors	Student
Pre-condition	For dropping a course: After registration period starts and before registration period ends.
Parameters	Input course code: String
Steps	- Student selects the drop course functionality from the menu. - Student then enters course code to drop the course. 1) If course code exists, then the course dropping is successful. 2) If course code doesn’t exist, then system re-prompts.

UC13	
Title	A student fulfills a course in which they are registered to
Participating actors	Student
Pre-condition	After the term is done while still registered to course(s).
Parameters	Void
Steps	- Student finishes all of the days required in order to finish a term while registered to a course.

UC14 and UC15 are not supported by the system and do not require completion.

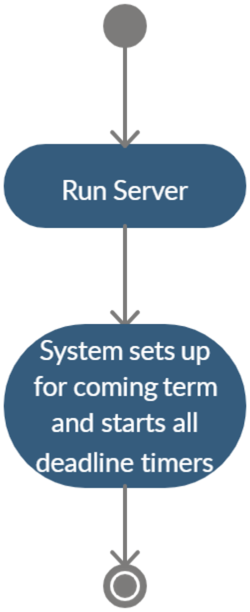
Step 2:

Totem Paper Activity Diagram

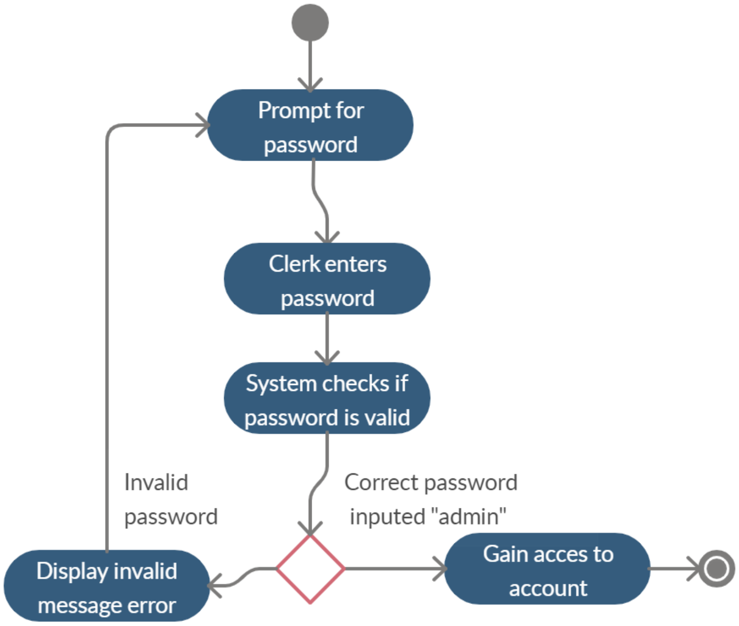

System
(a.k.a. Server) : Run Server
Clerk
Student
Course
Before System is run
Before Registeration Period starts
After Registeration Period starts
After Registeration Period Ends
After Term Ends
A
(UC0) The System Sets up the coming term
B
(UC1) Clerk logs into the ATC
C
(UC2) Clerks logs out of the ATC
D
(UC8) Student logs in (student
E
(UC9) Student logs out of the ATC
F
(UC6) Clerk creates a student for term (student
G
(UC7) Clerk deletes a student for term (Student
H
(UC3) Clerk creates course (course code)
(UC4) Clerk deletes course (course code)
I
K
(UC11) Student deregisters in course (Student no, course code)
(UC10) Student selects/registers in a course (Student no, course code)
J
(UC12) The Student drops a course (Student no,
L
M
(UC5) Clerk cancels course (course code)
(UC13) Student fulfills a course (Student no, course code)
N

Step3:

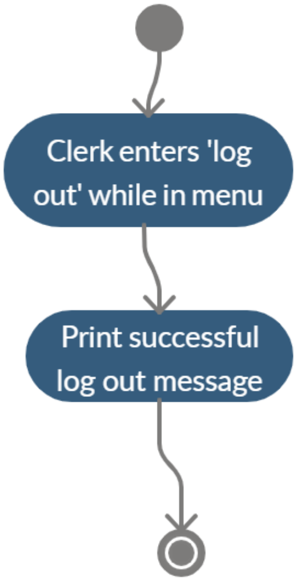
UC0:


Run Server
System sets up for coming term and starts all deadline timers

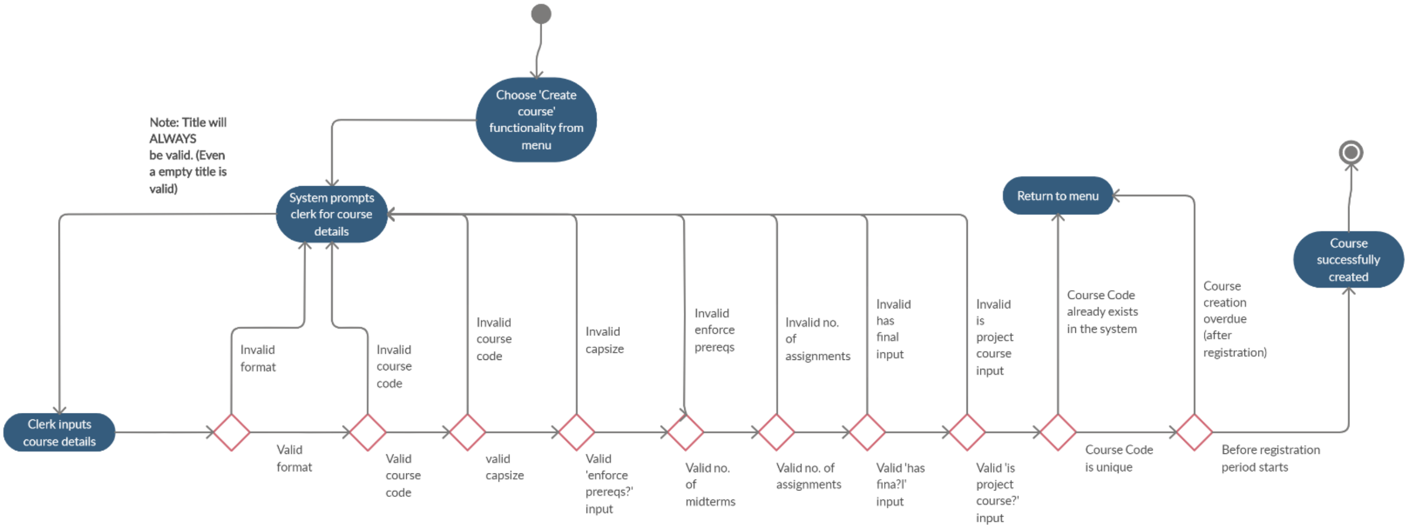
UC1:


Prompt for password
Clerk enters password
System checks if password is valid
Invalid password
Correct password inputed "admin"
Display invalid message error
Gain acces to account

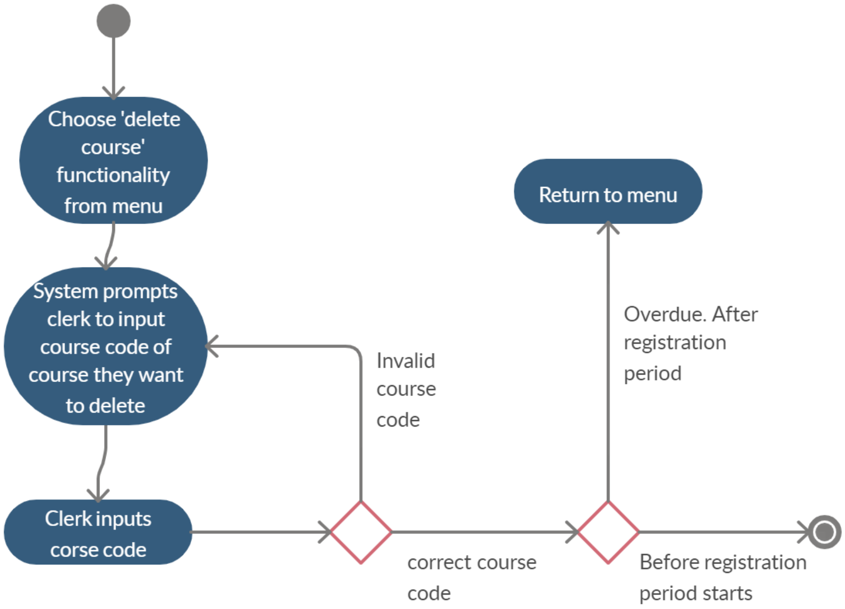
UC2:


Clerk enters 'log out' while in menu
Print successful log out message

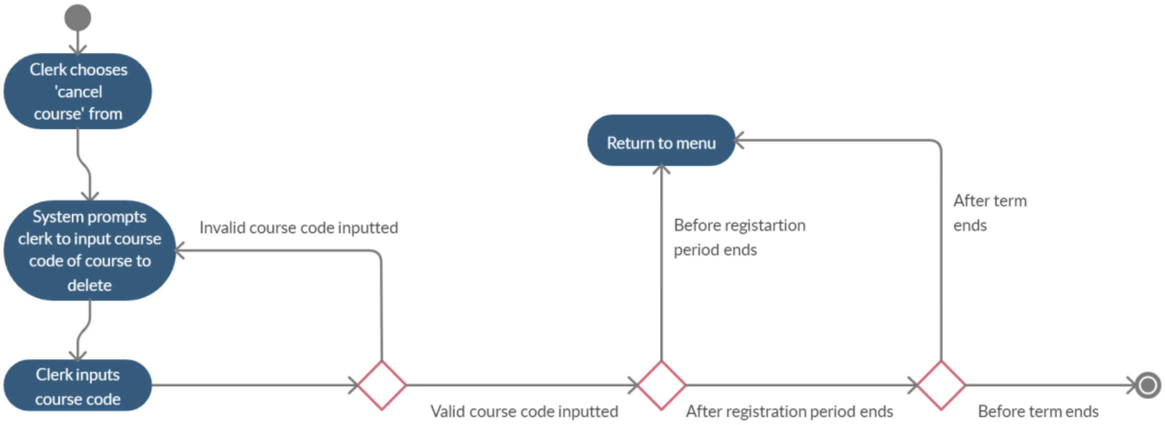
UC3


Choose 'Create course' functionality from menu
Note: Title will ALWAYS be valid. (Even a empty title is valid)
System prompts clerk for course details
Clerk inputs course details
Invalid format
Valid format
Invalid course code
Valid course code
Invalid course code
valid capsize
Invalid capsize
Valid 'enforce prereqs?' input
Invalid enforce prereqs
Valid no. of midterms
Invalid no. of assignments
Valid no. of assignments
Invalid has final input
Valid 'has fina?l' input
Invalid is project course input
Valid 'is project course?' input
Return to menu
Course Code already exists in the system
Course Code is unique
Course creation overdue (after registration)
Before registration period starts
Course successfully created

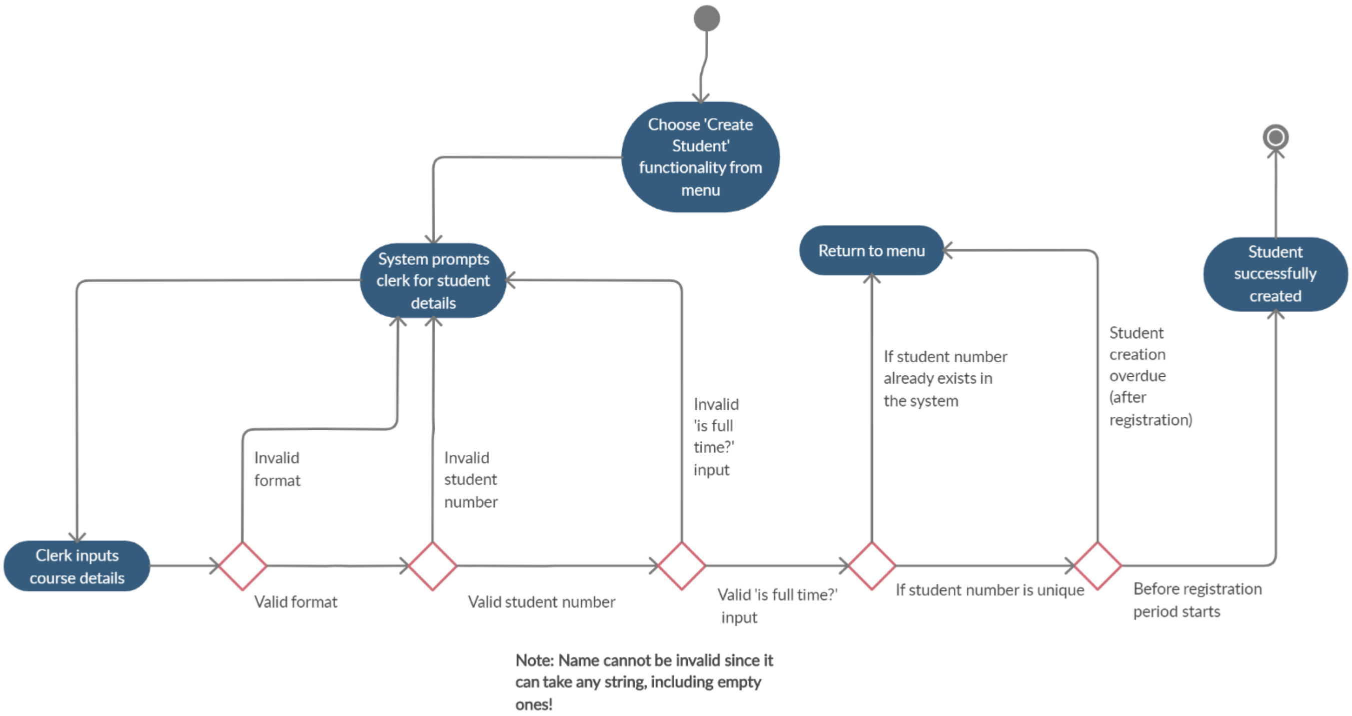
UC4:


Choose 'delete course' functionality from menu
System prompts clerk to input course code of course they want to delete
Clerk inputs corse code
Invalid course code
correct course code
Return to menu
Overdue. After registration period
Before registration period starts

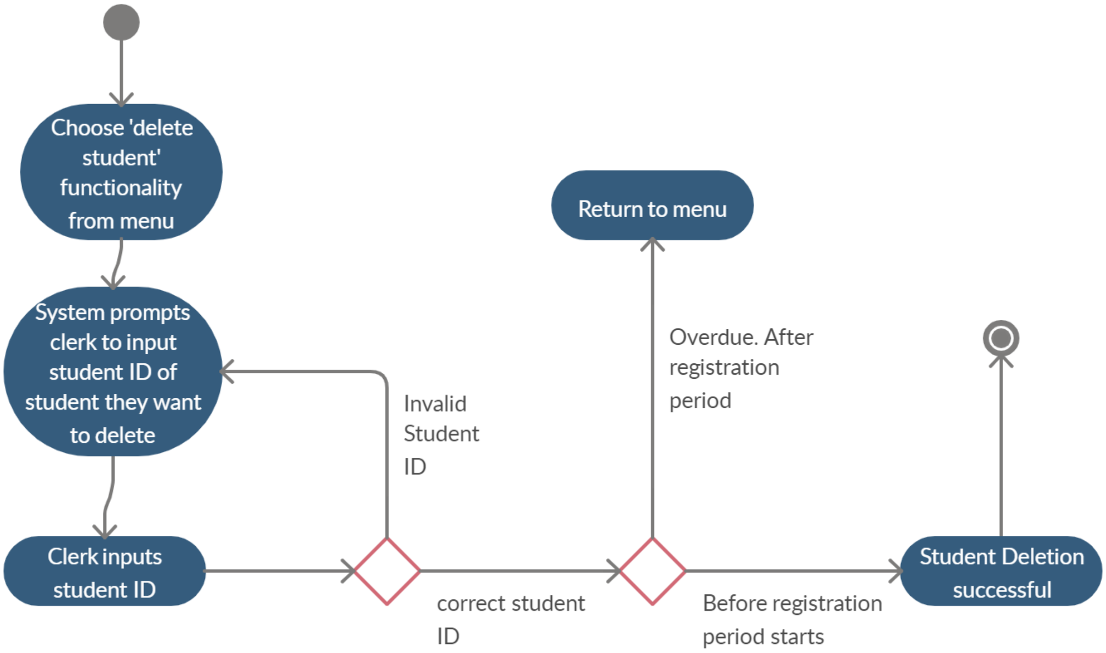
UC5:


Clerk chooses 'cancel course' from
System prompts clerk to input course code of course to delete
Clerk inputs course code
Invalid course code inputted
Valid course code inputted
Return to menu
Before registartion period ends
After registration period ends
After term ends
Before term ends

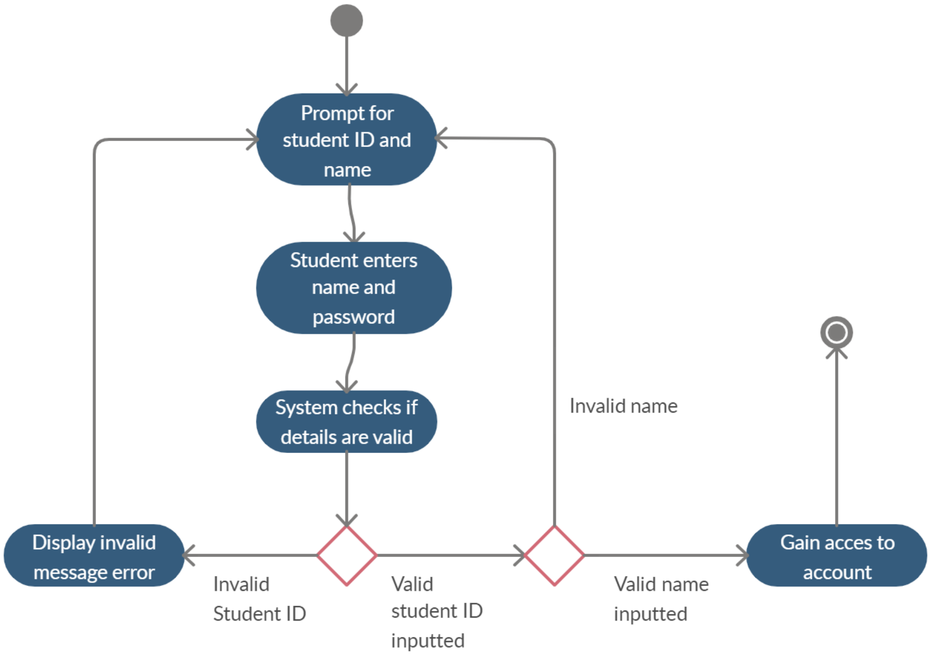
UC6:


Choose 'Create Student' functionality from menu
System prompts clerk for student details
Clerk inputs course details
Invalid format
Valid format
Invalid student number
Valid student number
Invalid 'is full time?' input
Valid 'is full time?' input
Return to menu
If student number already exists in the system
If student number is unique
Student creation overdue (after registration)
Before registration period starts
Student successfully created
Note: Name cannot be invalid since it can take any string, including empty ones!

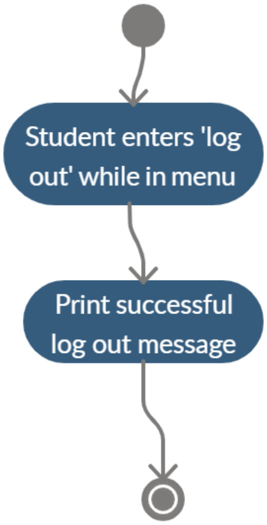
UC7:


Choose 'delete student' functionality from menu
System prompts clerk to input student ID of student they want to delete
Clerk inputs student ID
Invalid Student ID
correct student ID
Return to menu
Overdue. After registration period
Before registration period starts
Student Deletion successful

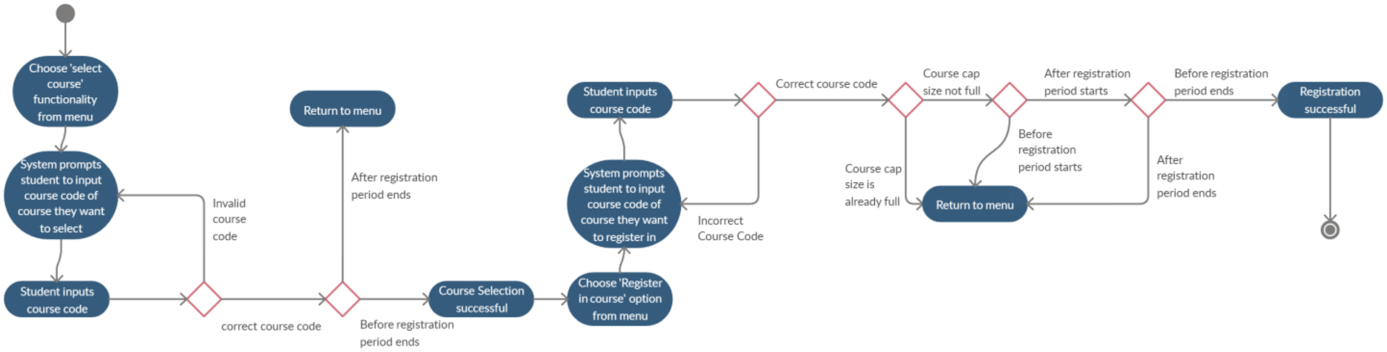
UC8:


Prompt for student ID and name
Student enters name and password
System checks if details are valid
Invalid name
Display invalid message error
Invalid Student ID
Valid student ID inputted
Valid name inputted
Gain acces to account

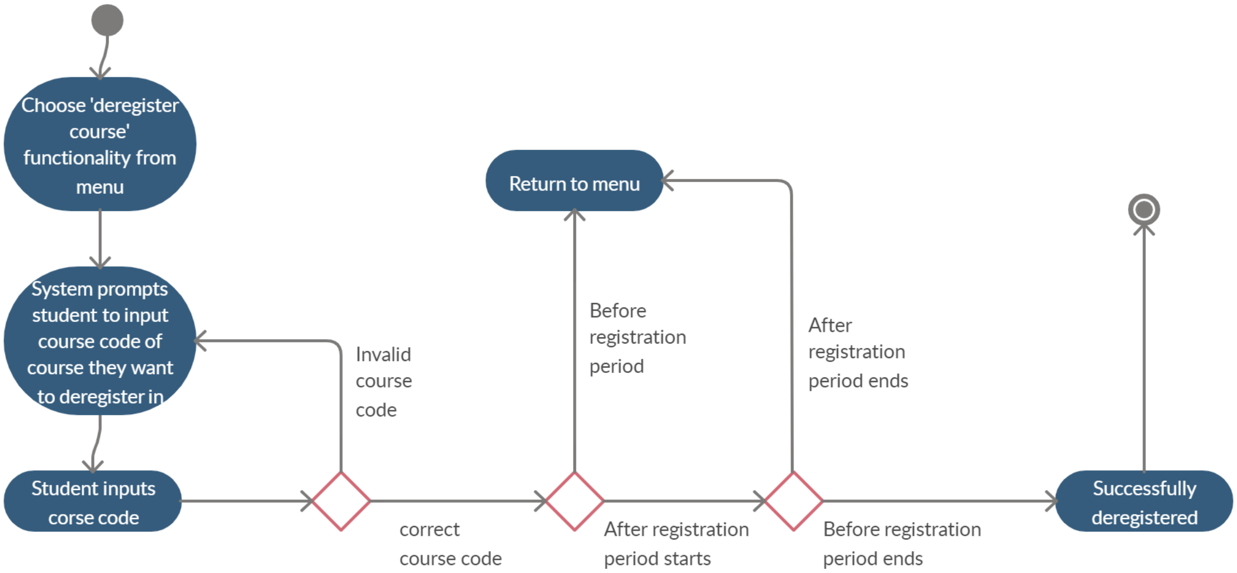
UC9:


Student enters 'log out' while in menu
Print successful log out message

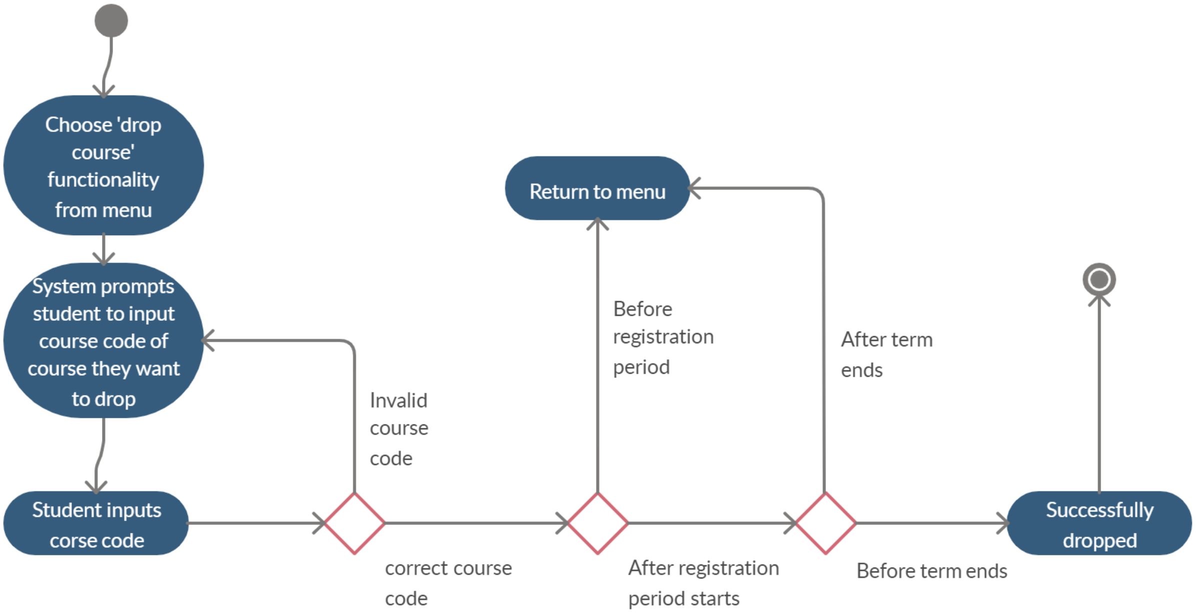
UC10:


Choose 'select course' functionality from menu
System prompts student to input course code of course they want to select
Student inputs course code
Invalid course code
correct course code
Return to menu
After registration period ends
Before registration period ends
Course Selection successful
Choose 'Register in course' option from menu
System prompts student to input course code of course they want to register in
Student inputs course code
Incorrect Course Code
Correct course code
Course cap size not full
Course cap size is already full
Return to menu
After registration period starts
Before registration period starts
Before registration period ends
After registration period ends
Registration successful

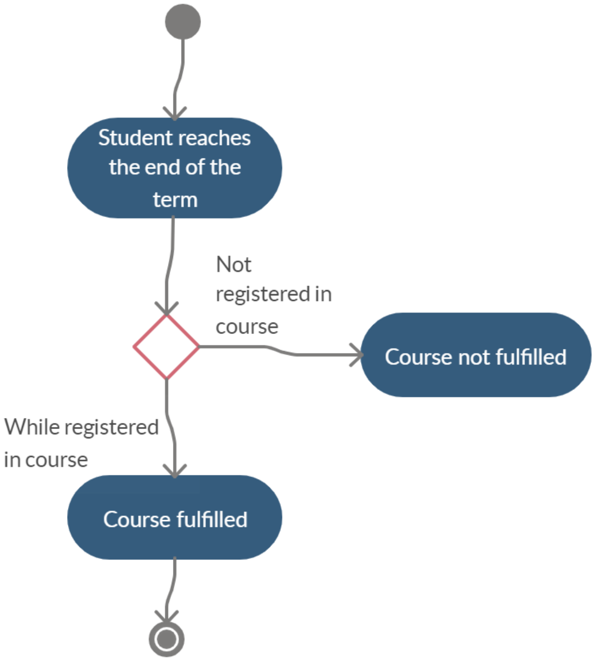
UC11:


Choose 'deregister course' functionality from menu
System prompts student to input course code of course they want to deregister in
Student inputs corse code
Invalid course code
correct course code
Return to menu
Before registration period
After registration period starts
After registration period ends
Before registration period ends
Successfully deregistered

UC12:


Choose 'drop course' functionality from menu
System prompts student to input course code of course they want to drop
Student inputs corse code
Invalid course code
correct course code
Return to menu
Before registration period
After registration period starts
After term ends
Before term ends
Successfully dropped

UC13:


Student reaches the end of the term
Not registered in course
Course not fulfilled
While registered in course
Course fulfilled

Step 5:

UC0			
	Path 1: Run the server		
UC1			
	Path 1: - Enter Password correctly. - Get access to account.		
		Operational variable: Password	Password = “admin”
	Path 2: - Enter Password incorrectly - Don’t get access to account. - get re-prompted for password		
		Operational variable: Password	Password != “admin”
UC2			
	Path 1: Enter Input in the menu in order to be logged out of the system		
		Operational variable: String x	X = “log out”
UC3			
	Path 1: - The clerk chooses the create course functionality from the menu before registration starts. - The system asks the clerk to input the course details (title, course code, capsize, enforce prereqs? no. of midterms, no. of assignments, has final? is project course?) that they want created. - Enter correct course details to create a		

	course. Course created successfully.		
		Operational variable 1: Title	Title = anything
		Operational variable 2: Course code	Course code = any 6 digited number that does not already exist
		Operational variable 3: capsize	Capsize = any number
		Operational variable 4: enforce prereqs?	Either “y” or “n”
		Operational variable 5: no. of midterms	Any number between 0 and 2
		Operational variable 6: has final?	Either “y” or “n”
		Operational variable 7: is project course?	Either “y” or “n”
		Operational variable 8: Before registration starts	True
	Path 2: - The clerk chooses the create course functionality from the menu after registration starts. - The system asks the clerk to input the course details (title, course code, capsize, enforce prereqs? no. of midterms, no. of assignments, has final? is project course?) that they want created. - **Enter incorrect course details to create a course**. Course created unsuccessfully.		
		Operational variable 1: Title	Title = anything (This variable will always be valid), even if empty.
		Operational variable 2: Course code	- Number that is not 6 digited. - Or input that is not a number.

		Operational variable 3: capsize	Capsize = - Non-number input
		Operational variable 4: enforce prereqs?	Not “y” or “n”
		Operational variable 5: no. of midterms	Non-number input or not 0,1,2
		Operational variable 6: has final?	Not “y” or “n”
		Operational variable 7: is project course?	Not “y” or “n”
		Operational variable 8: Before registration starts	False
	Path 3: - The clerk chooses the create course functionality from the menu after registration starts. - The system asks the clerk to input the course details (title, course code, capsize, enforce prereqs? no. of midterms, no. of assignments, has final? is project course?) that they want created. - **<u>Enter course details with same course code of another course that has been created previously.</u>** Course created unsuccessfully.		
		Operational variable 1: Title	Title = anything (This variable will always be valid), even if empty.
		Operational variable 2: Course code	**Course code = - 6 digited number that already exists.**
		Operational variable 3: capsize	Capsize = - Non-number input
		Operational variable 4: enforce prereqs?	Not “y” or “n”
		Operational variable 5: no. of midterms	Non-number input or not 0, 1, 2

		Operational variable 6: has final?	Not “y” or “n”
		Operational variable 7: is project course?	Not “y” or “n”
		Operational variable 8: Before registration starts	False
UC4			
	Path 1: - The clerk chooses the delete course functionality from the menu before registration starts. - The system asks the clerk to input the course they want to delete. - Enter correct course code of course. Successfully deleted.		
		Operational variable 1: Course code	Course code= Code has to already exist in the registered courses list for the system.
		Operational variable 2: Before registration started	true
	Path 2: - The clerk chooses the delete course functionality from the menu before registration starts. - The system asks the clerk to input the course they want to delete. - **Enter incorrect course code of course**. Failure to delete.		
		Operational variable 1: Course code	Course code = Code does not exist in the registered courses list.

		Operational variable 2: Before registration started	true
	Path 3: - The clerk chooses the delete course functionality from the menu **after registration started.** - The system asks the clerk to input the course they want to delete. - Enter correct course code of course. Failure to delete.		
		Operational variable 1: Course code	Course code = Code has to already exist in the registered courses list for the system.
		Operational variable 2: Before registration started	False
UC5			
	Path 1: - The clerk chooses the cancel course functionality from the menu after registration ends and before term ends. - The system asks the clerk to input the course they want to cancel. - Enter course ID of existing course to cancel a course. Course successfully cancelled.		
		Operational variable 1: Course code	Course code = Code that exists in the registered courses list.
		Operational variable 2: After registration period ends	true

		Operational variable 3: Before term ends	True
	Path 2: - The clerk chooses the cancel course functionality from the menu after registration ends and before term ends. - The system asks the clerk to input the course they want to cancel. - **Enter course ID of course that does not exist in the system.** - Course successfully cancelled.		
		Operational variable 1: Course code	**Course code = Code does not exist in the registered courses list.**
		Operational variable 2: After registration period ends	true
		Operational variable 3: Before term ends	True
	Path 3: - The clerk chooses the cancel course functionality from the menu **before registration period ends.** - The system asks the clerk to input the course they want to cancel. - Enter course ID of existing course to cancel a course. Course not cancelled.		
		Operational variable 1: Course code	Course code = Code that exists in the registered courses list.

		Operational variable 2: After registration period ends	**false**
		Operational variable 3: Before term ends	True
	Path 4: - The clerk chooses the cancel course functionality from the menu after registration period ends and **after term ends.** - The system asks the clerk to input the course they want to cancel. - Enter course ID of existing course to cancel a course. Course not cancelled.		
		Operational variable 1: Course code	Course code = Code does not exist in the registered courses list.
		Operational variable 2: After registration period ends	True
		Operational variable 3: Before term ends	**False**
UC6			
	Path 1: - The clerk chooses the create student functionality from the menu before registration starts. - The system asks the clerk to input the student's details that they want created. - Enter student details to create a student. Successfully created.		
		Operational variable 1: Student number	Unique 9 digit number

		Operational variable 2: Name	Any string
		Operational variable 3: fulltime status	“y” or “n”
		Operational variable 3: Before registration period started	True
	Path 2: - The clerk chooses the create student functionality from the menu **after registration starts.** - The system asks the clerk to input the student’s details that they want created. - Enter student details to create a course. - Student not created.		
		Operational variable 1: Student number	Student no. not 9 digit number
		Operational variable 2: Name	Always true (Can have no name)
		Operational variable 3: fulltime status	“y” or “n”
		Operational variable 3: Before registration period started	False
	Path 3: - The clerk chooses the create student functionality from the menu before registration starts. - The system asks the clerk to input the student’s details that they want created. - **Enter invalid student details.** - Student not created.		
		Operational variable 1: Student number	Student no. not 9 digit number
		Operational variable 2: Name	Always true (Can have no name)

		Operational variable 3: fulltime status	Not “y” or “n”
		Operational variable 3: Before registration period started	True
	Path 4: - The clerk chooses the create student functionality from the menu before registration starts. - The system asks the clerk to input the student’s details that they want created. - **<u>Enter student details with same student id of another student.</u>** - Student not created		
		Operational variable 1: Student number	**Student no. that already exists in the registered courses list.**
		Operational variable 2: Name	Always true (Can have no name)
		Operational variable 3: fulltime status	“y” or “n”
		Operational variable 3: Before registration period started	True
UC7			
	Path 1: - The clerk chooses the delete student functionality from the menu before registration starts. - The system asks the clerk to input the student they want to delete. - Enter student ID to delete existing student. Student is deleted.		
		Operational variable 1:	Student no. exists in the current system

		Student number	
		Operational variable 2: Before registration period started	True
	Path 2: - The clerk chooses the delete student functionality from the menu before registration starts. - The system asks the clerk to input the student they want to delete. - **Enter invalid student ID**. - Student not deleted.		
		Operational variable 1: **Student number**	**Student no. does not exist in the current system**
		Operational variable 2: Before registration period started	True
	Path 3: - The clerk chooses the delete student functionality from the menu **after registration starts.** - The system asks the clerk to input the student they want to delete. - Enter valid student ID to. - Student not deleted.		
		Operational variable 1: Student number	Student no. does not exist in the current system
		Operational variable 2: Before registration period started	**False**
UC8			
	Path 1:		

	- The system asks the user to input their login info. - Student enters their correct student ID and password in order to access the system. - Successful login accomplished		
		Operational variable 1: Student Number	Student number that already exists in the system
		Operational variable 2: Name	Name that exists in the system already
	Path 2: - The system asks the user to input their login info. - **Student enters incorrect student ID/ Name** - login failed		
		Operational variable 1: Student Number	Student number that DOES NOT exist in the system
		Operational variable 2: Name	Name that DOES NOT exist in the system.
UC9			
	Path 1: - Student enters ‘log out’ to the system in order to be logged out.		
		Operation variable 1: String	“Log Out”
UC10			
	Path 1: - The student chooses the select course functionality from the menu before registration period ends and after registration starts.		

	- The system asks the student to input the course they want to register in. - Student enters course code to select course then the course gets added to a list of possible choices of courses that can be registered. - Student then selects the register course functionality from the menu. - Student then enters course code again to register for the course. - Registration success		
		Operational variable 1: Course Code	Course Code that exists in the system (for selecting and registering)
		Operational variable 2: Course cap size not full	True
		Operational variable 3: After registration period started	True
		Operational variable 4: Before registration period ends	True
	Path 2: - The student chooses the select course functionality from the menu before registration period ends and **before registration starts.** - The system asks the student to input the course they want to register in.		

	- Student enters course code to select course. - Since course code exists, then the course gets added to a list of possible choices of courses to register in. - Student then selects the register course functionality from the menu. - Student then enters course code again to register for the course. - Registration failed.		
		Operational variable 1: Course Code	Course code that exists in the system (for selecting and registering)
		Operational variable 2: Course cap size not full	True
		Operational variable 3: **After registration period started**	**False**
		Operational variable 4: Before registration period ends	True
	Path 3: - The student chooses the select course functionality from the menu after registration period ends and after registration starts and **after registration period ends**. - The system asks the student to input the course they want to register in. - Student enters course code to select course and add to list of selected courses.		

	- Student then selects the register course functionality from the menu. - Student then enters course code again to register for the course. - Registration failed		
		Operational variable 1: Course code	Course code that exists in the system (for selecting and registering)
		Operational variable 2: Course cap size not full	True
		Operational variable 3: After registration period started	True
		Operational variable 4: Before registration period ends	False
	Path 4: - The student chooses the select course functionality from the menu before registration period ends and after registration starts and before registration ends. - The system asks the student to input the course they want to register in. - Student enters course code to select course. - Course gets added to a list of possible choices of courses that can be registered. - Student then selects the register course functionality from the menu.		

	- Student then enters course code of course again to register for the course. - **Registration fails because course is already in the student’s registered courses list.**		
		Operational variable 1: Course code	**Course code that already exists in the system. (for selecting or registering or both)**
		Operational variable 2: Course cap size not full	True
		Operational variable 3: After registration period started	True
		Operational variable 4: Before registration period ends	True
	Path 5: - The student chooses the select course functionality from the menu before registration period ends and after registration starts and before registration ends. - The system asks the student to input the course they want to register in. - Student enters course code to select course. - Course gets added to a list of possible choices of courses that can be registered. - Student then selects the register course		

	functionality from the menu. - Student then enters valid course code again to register for the course. - **The request gets declined because the course cap is FULL.** - Registration failed		
		Operational variable 1: Course Code	Course code that does not exist in the system
		Operational variable 2: Course cap size not full	**False**
		Operational variable 3: After registration period started	True
		Operational variable 4: Before registration period ends	True
UC11			
	Path 1: - Student selects the deregister course functionality from the menu. - Student then enters valid registered course code to deregister for it. - Deregistration is successful.		
		Operational variable 1: Course Code	Course code that exists in the student’s registered courses
		Operational variable 2: Registration period started	True
		Operational variable 3: Before registration period ends	True
	Path 2:		

	- Student selects the deregister course functionality from the menu after registration starts and before registration period ends. - **Student then enters invalid course code.** - Deregistration is unsuccessful.		
		Operational variable 1: Course Code	**Course code that does not exist in the student’s registered courses**
		Operational variable 2: Registration period started	True
		Operational variable 3: Before registration period ends	True
	Path 3: - Student selects the deregister course functionality from the menu after registration starts **and after registration period ends.** - Student then enters course code. - Deregistration is unsuccessful.		
		Operational variable 1: Course Code	Course code that exists in the student’s registered courses
		Operational variable 2: Registration period started	True
		Operational variable 3: **Before registration period ends**	**False**

	Path 4: - Student selects the deregister course functionality from the **menu before registration starts** and before registration period ends. - Student then enters invalid unregistered course code to deregister for it. - Deregistration is unsuccessful.		
		Operational variable 1: Course Code	Course code that does not exist in the student’s registered courses
		Operational variable 2: **Registration period started**	**False**
		Operational variable 3: Before registration period ends	False
UC12			
	Path 1: - Student selects the “drop course” functionality from the menu before the term has ended and after registration period ends. - Student then enters course code to drop the course. - Course dropping is successful.		
		Operational Variable 1: Course Code	Course code that does exist in the registered courses list
		Operational Variable 2: After Registration period ends	True

		Operational Variable 3: Before Term Ends	True
	Path 2: - Student selects the “drop course” functionality from the menu before the term has ended and after registration period ends. - **Student then enters INVALID course code**. - Course dropping is unsuccessful.		
		Operational Variable 1: **Course Code**	**Course code that does not exist in the registered courses list**
		Operational Variable 2: After Registration period ends	True
		Operational Variable 3: Before Term Ends	True
	Path 3: - Student selects the “drop course” functionality from the menu before the term has ended and **before registration period ends.** - Student then enters course code to drop the course. - Course dropping is unsuccessful.		
		Operational Variable 1: Course Code	Course code that does exist in the registered courses list
		Operational Variable 2: **After Registration period ends**	**False**

		Operational Variable 3: Before Term Ends	True
	Path 4: - Student selects the “drop course” functionality from the menu **<u>after the term has ended</u>** and after registration period ends. - Student then enters course code to drop the course. - Course dropping is unsuccessful.		
		Operational Variable 1: Course Code	Course code that does exist in the registered courses list
		Operational Variable 2: After Registration period ends	False
		Operational Variable 3: Before Term Ends	**False**
UC13			
	Path 1: - Student finishes a full term while registered to a course. - Successful.		
		Operational Variable 1: Registered in Course	True
		Operational Variable 2: After Term Ends	True
	Path 2: - Student finishes a full term while **<u>NOT registered to a course.</u>** - Unsuccessful		
		Operational Variable 1: Registered in Course	**False**

		Operational Variable 2: After Term Ends	True
	Path 3: - **<u>Student does NOT finish a full term</u>** while registered to a course. - Unsuccessful		
		Operational Variable 1: Registered in Course	True
		Operational Variable 2: After Term Ends	**False**

<u>From step 5, I derived all the above cases (Valid and Invalid) into test cases that I implement with the help of cucumber.</u>

<u>Step 6:</u>

Parameterized use case sequences	- **B.C** - **B.H(**Course code)**.M(**Course code) - **B.F**(student no.).**G**(Student no.).**H**(Course Code).**I**(Course Code) - **B.F**(Student no).**D.E** - **B.F**(student no.).**G**(Student no.).**H**(Course Code).**J**(Student no, Course code).**N**(Student no, Course code) - **B.H**(Course code).**J**(Student no, Course code).**K**(Student no, Course code) - **B.H**(Course code).**J**(Student no, Course code).**L**(Student no, Course code)
Parameter instances (Symbolic values)	- Clerks (3): c1, c2, c3 - Students (4): s1, s2, s3, s4 - Courses (5): c1, c2, c3, c4, c5

<u>From this, I derive all the VALID test paths I need to take (with the help of cucumber):</u>

- B->C.
- B->H->M
- B->F->D->E
- B->F->G->H->I
- B->F->G->H->J->N
- B->H->J->K
- B->H->J->L