For every method

**User: (AL)**

**Instantiate User**

Open Textbox’s for username and password

Set both values of variables “username” and “password” to empty strings

Set both values of variables “tuition” and “finAid” to 0.00

Set value of variable “file” to NULL

If “Login button” is pressed

Call User(username, password)

Call login method and verify “username” and “password” are correct

If they’re correct

Display the Main Menu

Set “a True/False condition” to “True”

Return True

Else

Display error message

Set “a True/False condition” to “False”

Return False

End of If Loop

While (“a True/False condition is True”) do

If “Grades Button” is pressed

Instantiate userGrade class,

Call getGrades method

Return grades

Else if “Course Schedule Button” is pressed

Instantiate Course Object,

Instantiate Calendar Object, passing in term, startDate, endDate

Call displayCalendar method on Calendar Object

Display Course Schedule Window

Else if “StudentID Button” is pressed

Instantiate StudentID Object

If “file” = NULL

Ask to upload image of Student ID

Open file menu

Else

Display the studentID file as an image

End of Loop

Else if “Campus Map Button” is pressed

Instantiate Course Class

Call maps constructor from Course Class

Else if “Logout” Button is pressed

Call **Logout** method

End of If Loop

End of While Loop

**User**(userName, password)

**login**(userName, password)

Get username and password

**logout():**

Logout of account specified by the username

**getUserName():** String

Get string from username textbox

**updateUserName**(userName)

Set username to username

**getPassword():**

Get password from password textbox

**updatePassword**(password)

Set password to password

**deleteUser():**

Delete User object and its data

**addCourse**(course)

Adds selected course to schedule

**getGrades():** userGrade

Get userGrade from Course object and put them in a list

**studentID()**: file

Create studentID object

**updateID**(file):

Get new image

Set file to the image

**getTuition()**: tuition

Get tuition value

**updateTuition**(tuition)

Set tuition to tuition

**payTuition**(tuition)

**updateFinAid**(finAid)

Get financial aid information from CCM

Set it equal to finAid

**getFinAid():** finAid

Set finAid to finAid

**calculateCost**(tuition, finAid): decimal

Set netcost equal to tuition minus finAid

return netCost

**viewCalendar():**

Display Calendar Object

**getTermGPA():** currentTermGPA

**calculateCurrentGPA():**

Add the grade value of each course by called getGrade for each course, multiply by the number of credits, then divide

**viewGrades: list of userGrade**

Add getGrades to a list.

Display list

**Assignment DM**

**assignment**(totalPoints, assignmentName): assignment

Set totalPoints to totalPoints

Set assignmentName to assignmentName

Set pointsEarned to 0.

**deleteAssignment**()

Delete assignment object and information related to it

**getName**():string

Get assignmentName

**setName**(assignmentName)

Set assignmentName to assignmentName

**getpointsEarned**():int

Get pointsEarned

**setpointsEarned**(pointsEarned)

Set pointsEarned to pointsEarned

**gettotalPoints**():int

Get totalPoints

**settotalPoints**(totalPoints)

Set totalPoints to totalPoints

**Course: DM**

**course**(startTime, endTime, section, name, startDate, endDate, roomNum, location, term)

Set startTime to startTime

Set entTime to endTime

Set section to section

Set name to name

Set startDate to startDate

setendDate to endDate

Set roomNumber to roomNum

Set location to location

Set term to term

Set grade to 100.00

**getTerm**(): string

Get term

**setTerm**(term)

Set term to term

**getGrade**(): float

Get grade

**calculateGrade**()

Initiate totalpoints

Initiate earned points

For each item in assignments

Add assignment’s total points to total points

Add assignment’s earned points to earned points

Get earned points divided by total points

**getStartTime**():int

Get startTime

**setStartTime**(startTime)

Set startTime to startTime

**getEndTime**():int

Get endTime

**setEndTime**(endTIme)

Set endTime to endTime

**getSection**():int

Get section

**setSection**(section)

Set section to section

**getName**():string

Get name

**setName**(name)

Set name to name

**getStartDate**():date

Get startDate

**setStartDate**(startDate)

Set startDate to startDate

**getEndDate**():date

Get endDate

**setEndDate**(endDate)

Set endDate to endDate

**getRoomNumber**():int

Get roomNumber

**setRoomNumber**(roomNumber)

Set roomNumber to roomNumber

**deleteCourse**()

Delete course object and all of its data

**launchMaps**(roomNumber, gps)

Launch the google maps api

Input the room number into the gps

**addAssignment**(int, String)

Create new assignment with given name and total number of points available

**Calendar:**

**calendar**(term, startDate, endDate)

Set term to term

Set startDate to startDate

Set endDate to endDate

**deleteCalendar**()

Delete calendar object and all of its data

**addEntry**(course)

Create event in google calendar API

Set start time of event to start time of course

Set end time of event to end time of course

Set start date of event to start date of course

Set end date of event to end date of course

**displayCalendar**()

Call Google Calendar API

Display Calendar

**closeCalendar**()

Closes google calendar API

**getTerm**():string

Get term

**setTerm**(term)

Set term to term

**getStartDate**():date

Get startDate

**setStartDate**(startDate)

Set startDate to startDate

**getEndDate**():date

Get endDate

**setEndDate**(endDate)

Set endDate to endDate