COMP4005 Information Systems Theory, Methodology and Architecture (17-18 Semester 1)

Assignment 3 – Object Oriented Methodology

By
GROVER Karan Yogesh
16227018

Table of Contents

Problem Definition

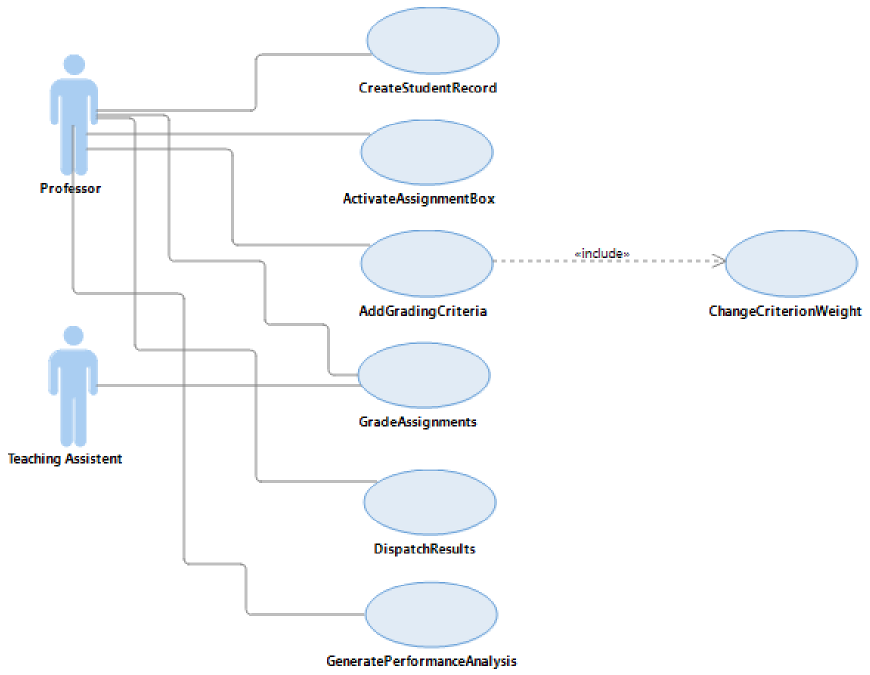
University teaching staff carries out most grading and evaluation of student submissions manually and the task involves several teaching assistants to complete on time. Submissions are either received online or in paper. Upon the completion of this process, the grades are sent back to the students through the moodle platform or by email.

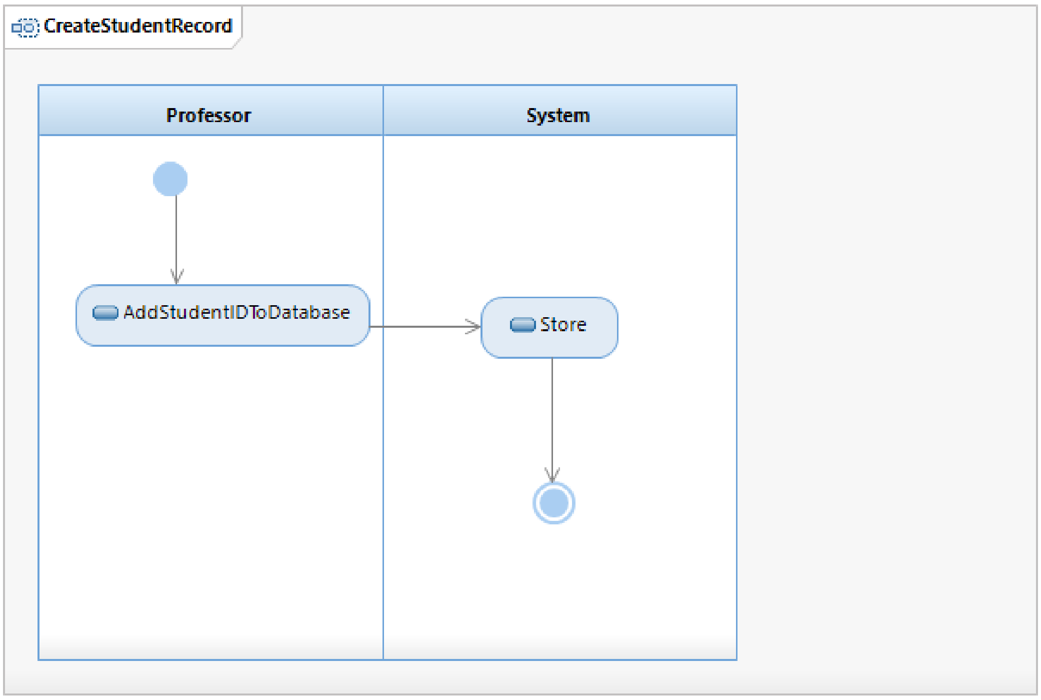
Any data analysis required to assess the performance of the class of students in a certain area of assessment must also be carried out manually with the accepted risk of human error in calculation. Also, if the weight of a certain part of the assessment has to adjusted, usually the whole grading process would have to be repeated or recalculated in a manner which is inefficient and time consuming.

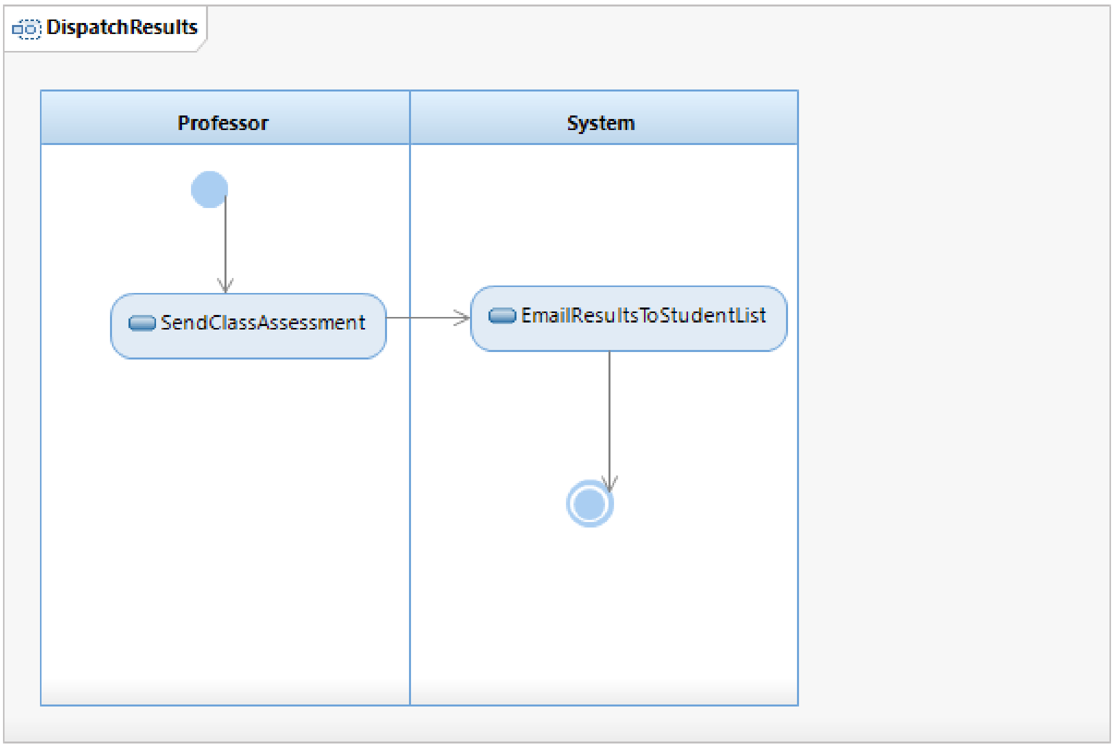
The purpose of developing the Enhanced Academic Learning and Analysis System (EALAS) system, is to address the shortcomings of the current arrangement discussed above.
one of the purposes of this system is to receive submissions like the ones discussed and automate certain tasks of grading in order to expedite the process and delivery of results.

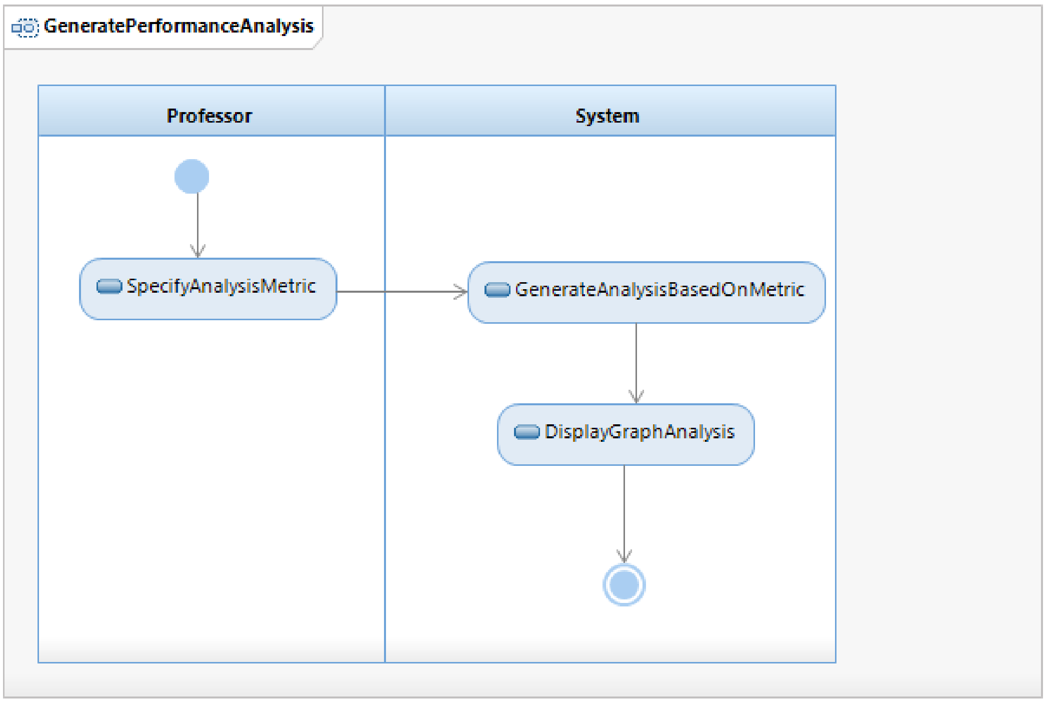
OO development steps

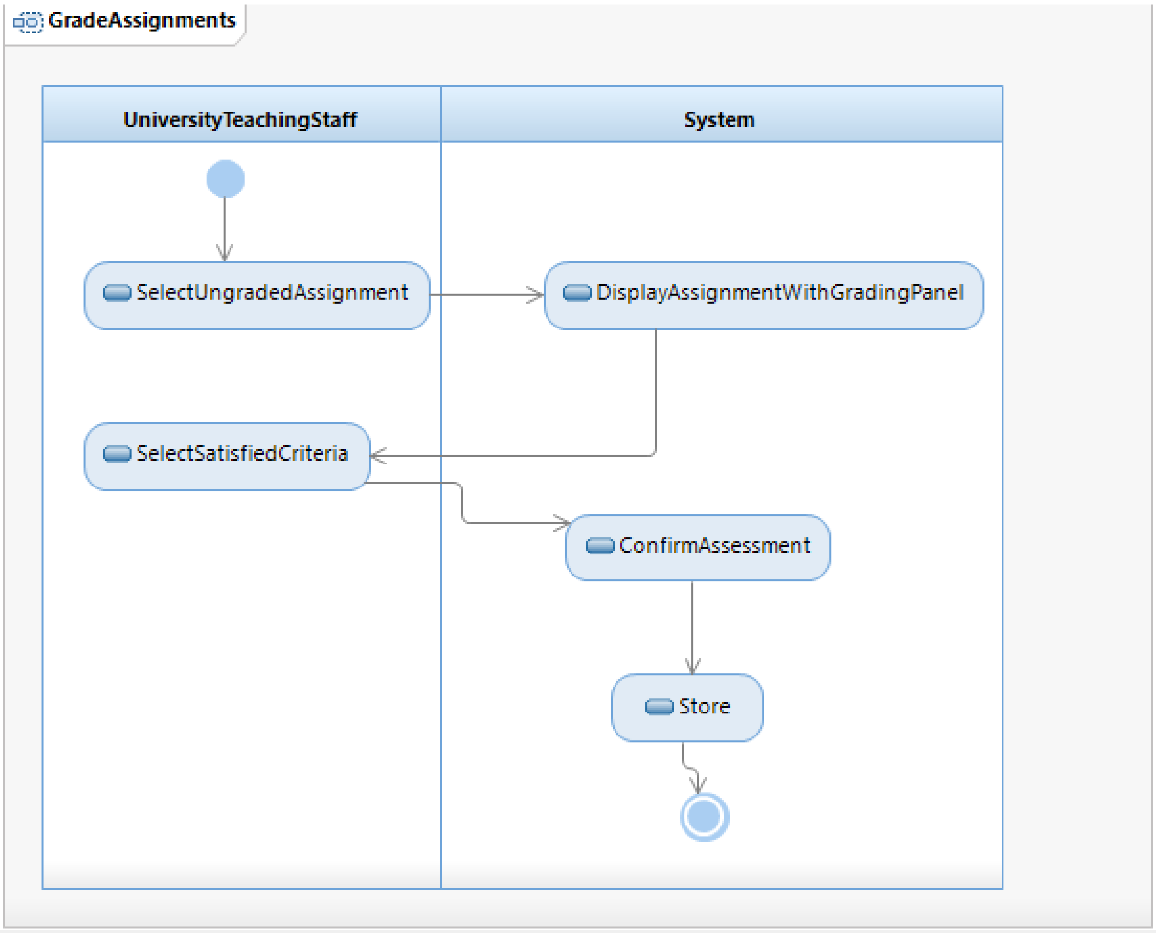
When adopting a Model Driven Architecture for developing the EALAS and using the Model Driven Development approach in the scope of Object-Oriented software development, UML is extensively used. The process begins by the development of conceptual UML models known as Platform Independent Models (PIMs) in Object Management Group (OMG) terminology. Such models include User case diagrams. The next stage involves Platform specific models, which usually dependent on the technology used and allow mapping and conversion to software code. Finally, the last stage, known as the Code Model, is used to generate code using implementation specific models. Examples of such models made using Rational Software Architect and Infosphere Data Architect can be found below

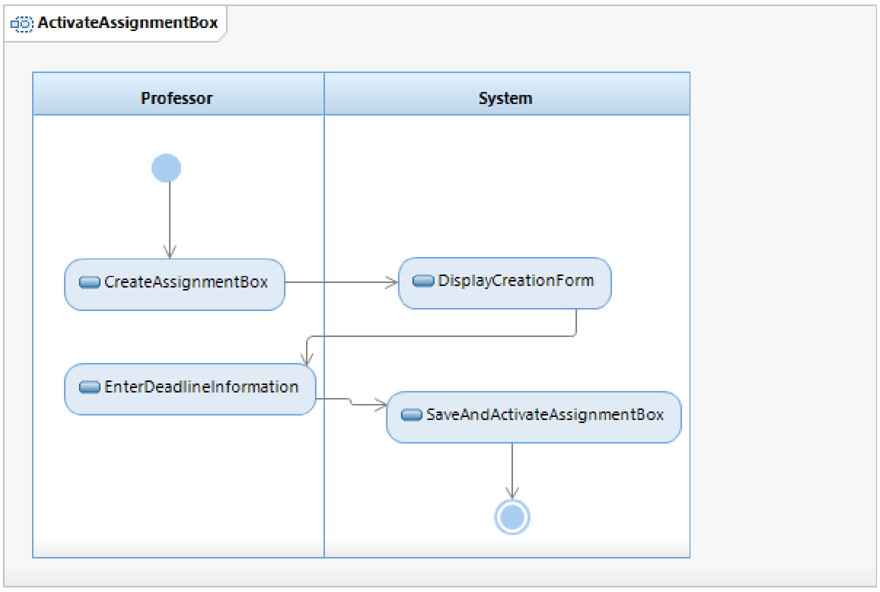



CreateStudentRecord
Professor
System
AddStudentIDToDatabase
Store

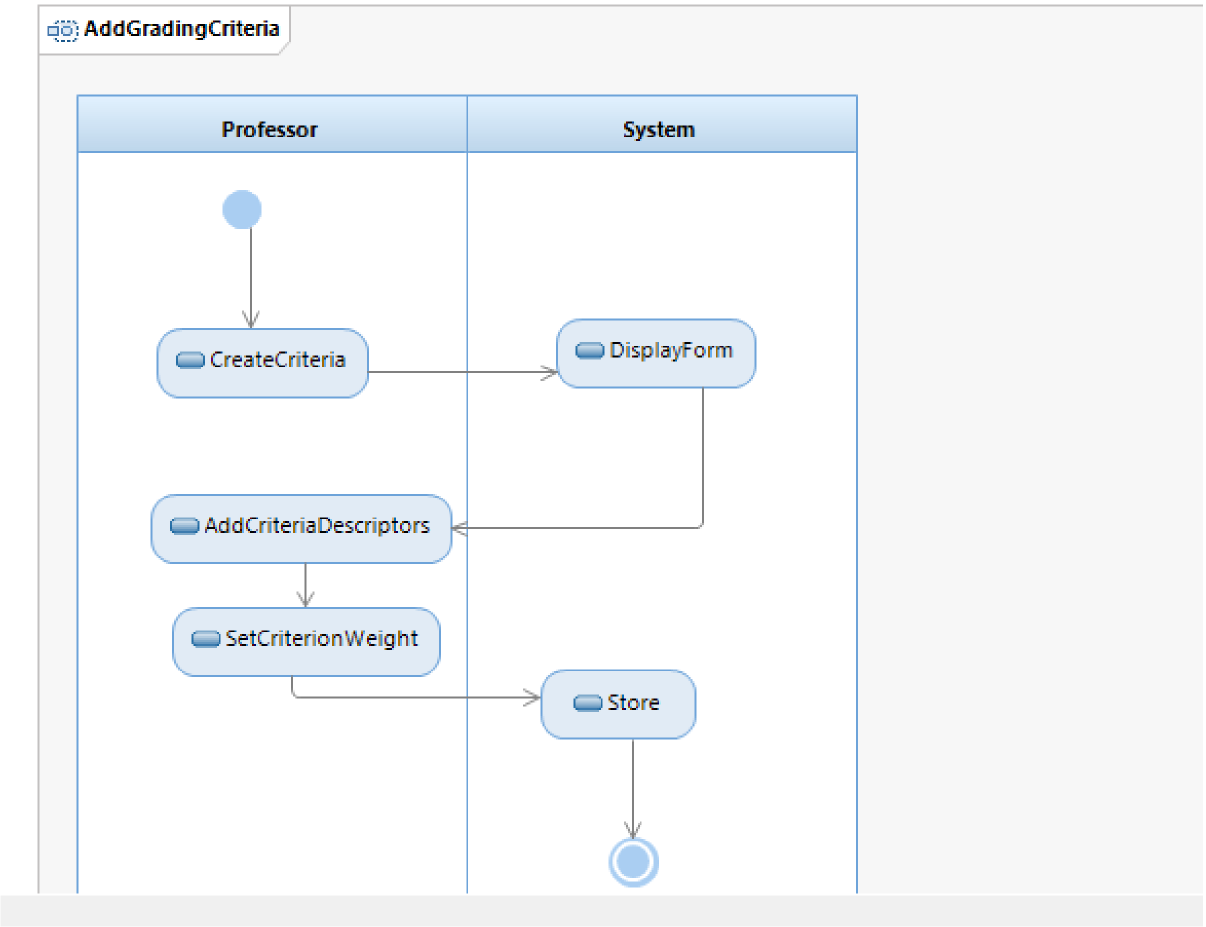

DispatchResults
Professor
System
SendClassAssessment
EmailResultsToStudentList


GeneratePerformanceAnalysis
Professor
System
SpecifyAnalysisMetric
GenerateAnalysisBasedOnMetric
DisplayGraphAnalysis

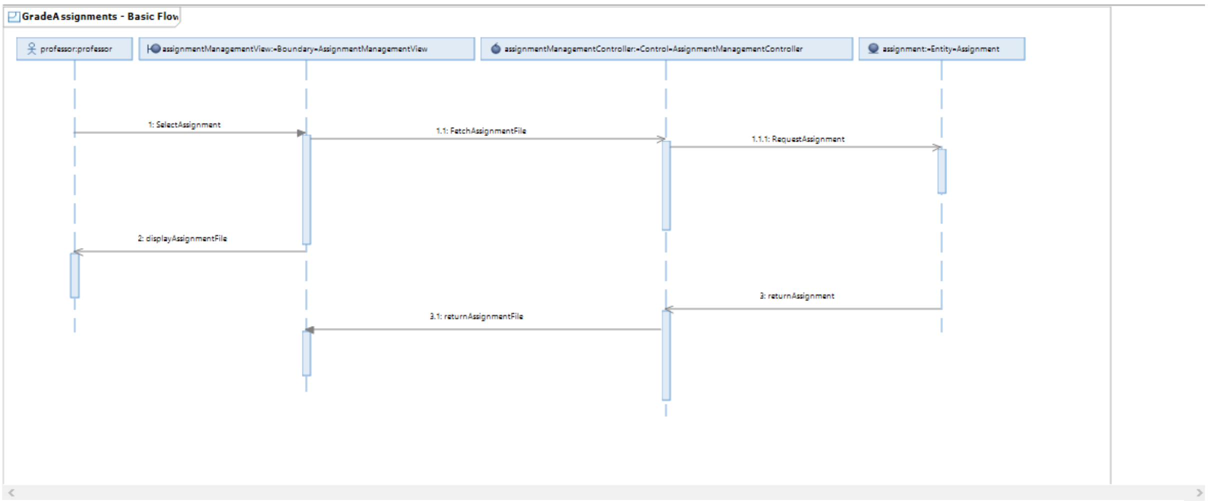

GradeAssignments
UniversityTeachingStaff
System
SelectUngradedAssignment
DisplayAssignmentWithGradingPanel
SelectSatisfiedCriteria
ConfirmAssessment
Store

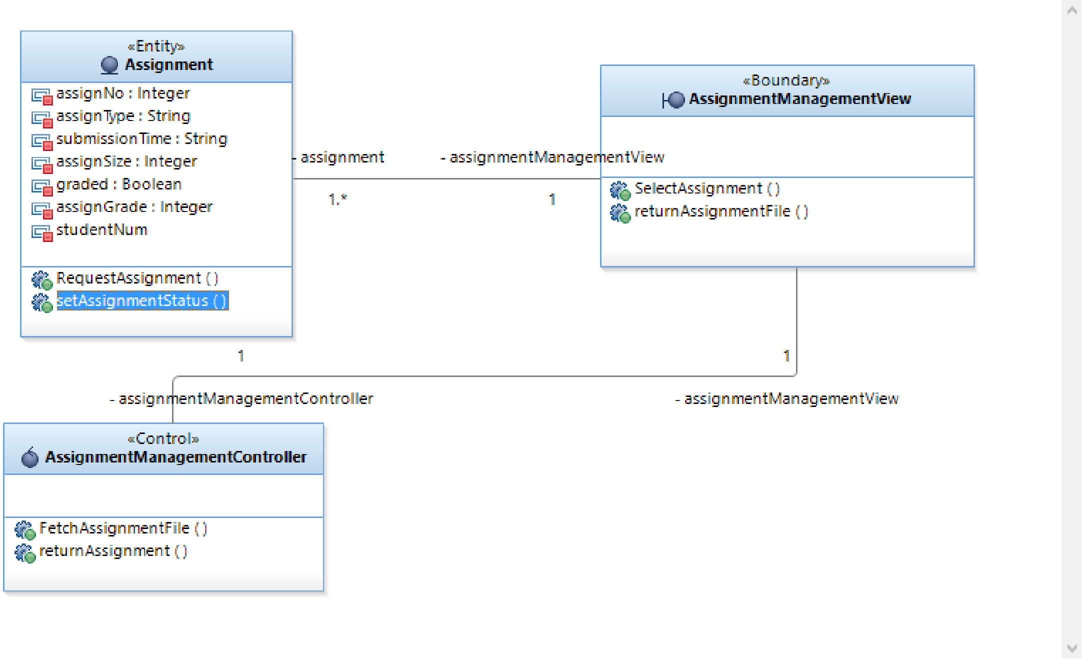


ActivateAssignmentBox
Professor
System
CreateAssignmentBox
DisplayCreationForm
EnterDeadlineInformation
SaveAndActivateAssignmentBox



AddGradingCriteria
Professor
System
CreateCriteria
DisplayForm
AddCriteriaDescriptors
SetCriterionWeight
Store


GradeAssignments - Basic Flow
professor:professor
assignmentManagementView:«Boundary»AssignmentManagementView
assignmentManagementController:«Control»AssignmentManagementController
assignment:«Entity»Assignment
1: SelectAssignment
1.1: FetchAssignmentFile
1.1.1: RequestAssignment
2: displayAssignmentFile
3: returnAssignment
3.1: returnAssignmentFile


«Entity»
Assignment
assignNo : Integer
assignType : String
submissionTime : String
assignSize : Integer
graded : Boolean
assignGrade : Integer
studentNum
RequestAssignment ( )
setAssignmentStatus ( )
«Boundary»
AssignmentManagementView
SelectAssignment ( )
returnAssignmentFile ( )
- assignment
- assignmentManagementView
1..*
1
1
1
- assignmentManagementController
- assignmentManagementView
«Control»
AssignmentManagementController
FetchAssignmentFile ( )
returnAssignment ( )

Critique and Evaluation

While the Model Driven Development approach does allow individuals with non-technical expertise to form functional prototypes of systems based on the requirements specified in models, it can be seen that the generation of complex logic required for complex, real-world systems is not always possible and therefore, software developers will be required to refine the generated code. That said, the Object-Oriented Development approach when coupled with Model Driven Development does give a clear view of the functions of the system and helps software developers understand requirements more effectively than the soft-systems approach.