FACULTY OF ENGINEERING AND APPLIED SCIENCE

SOFE 3650U – Software Design & Architecture

Final Project

CRN: 43509

Prof: Ramiro

Date: December 5, 2018

Group # 1

MEMBER			
#	Last Name:	First Name:	Student ID:
3	Patel	Harsh	100580778
4	Bhavsar	Karn	100557957

4.3.3 Iteration 2: Identifying Structures to Support Primary Functionality

This section presents the results of the activities that are performed in each of the steps of ADD in the second iteration of the design process for the CMS system. In this iteration, we move on from the generic description of functionality used in iteration 1 to more detailed decisions that will drive implementation and hence the formation of development teams. We cannot design everything up front. After the first iteration was established an overall system structure. For this second iteration is to reason about the units of implementation, which affects team information, interfaces and the means by which development tasks may be distributed, outsourced and implemented in sprints.

Step 2: Iteration Goal by Selecting Drivers

The goal of this iteration is to address the general architectural concern of the management system and data server. Understanding these elements is useful not only for understanding how functionality is supported but also working on how to carry out a database.

The primary focus is:

- UC-1 Core Management system
- UC-2 Course information
- UC-4 File Info
- UC-7 Database

Step 3: Choose Elements of the System to Refine

The elements that will be refined in this iteration are the design architectures and diagrams will be used to explain the element. Details of each modules in the layer is described and outlined which will help explaining the system.

Step 4: Choose Design Concepts that Satisfy Selected Drivers

In this iteration, several design concept and patterns to provide functionality to the system. The web application will be a 4-tiers.

Pattern	**Rationale and Assumptions**
Create Domain Model	- It is very important to initialize our domain for the CMS. - Need to identify entities - Once the entities are identified you will need to relate it to each other - You must create a domain for both functionality and data
Identify Domain Model	- This element is building in the domain object - There are risks with the requirements
Decompose into general and specialized components	- It will represent sets of functionality - It will have supported by sub layers within the system - It will all be related to their corresponding layer

Step 5: Instantiate Architectural Elements, Allocate Responsibilities, Define Interfaces

The instantiate design decision in this iteration are summarized in the following tale:

Design Decisions and Location	**Rationale and Assumptions**
Create an initial domain model	- Need to create an initial domain - Only need to identify our primary drivers
Map the system	- Using a Case Diagram will be the best way to map the system
Decompose the domain objects across the specified layers	- Need to ensure all the functionalities are identified - Do not test all the modules only the ones that are used the most often should be tested
Connect components	- Broken down components allow us to use different aspects

Step 6: Sketch Views and Record Design Decisions

As a result of the decisions made in step 5, several diagrams are created.

- Figure 1 shows an initial domain model for the system
- Figure 2 shows the domain objects that are instantiated for the use case model
- Figure 3 shows a sketch of a model view with modules that are derived from the business objects and associated with the primary use cases. Note that explicit interfaces are not shown but their existence is assumed.
- Figure 4: UC-1 shows how the content is retrieved
- Figure 5: UC-2 shows how the message system is working
- Figure 6: UC-3 shows how the file is retrieved

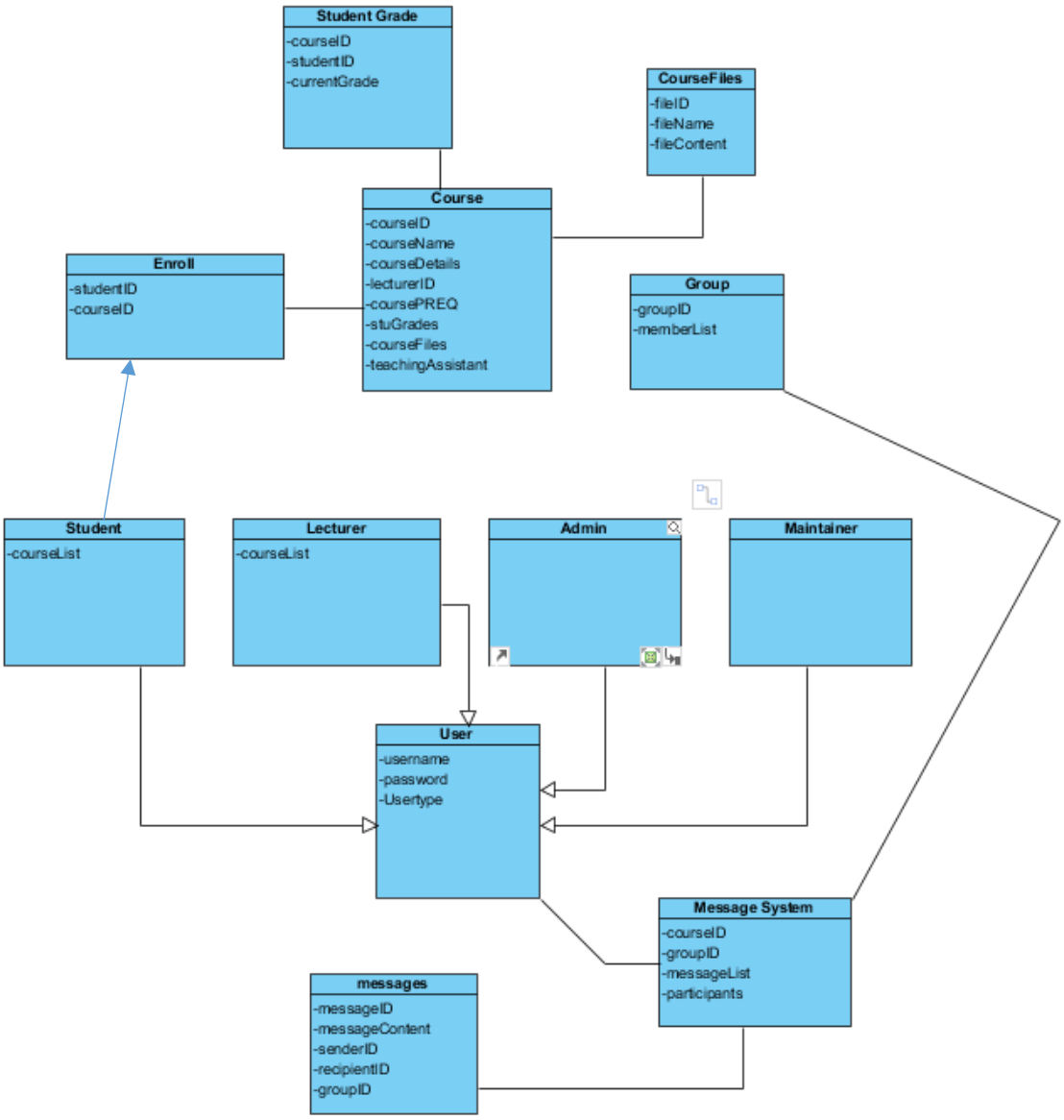

Figure 1: Figure 1 shows an initial domain model for the system

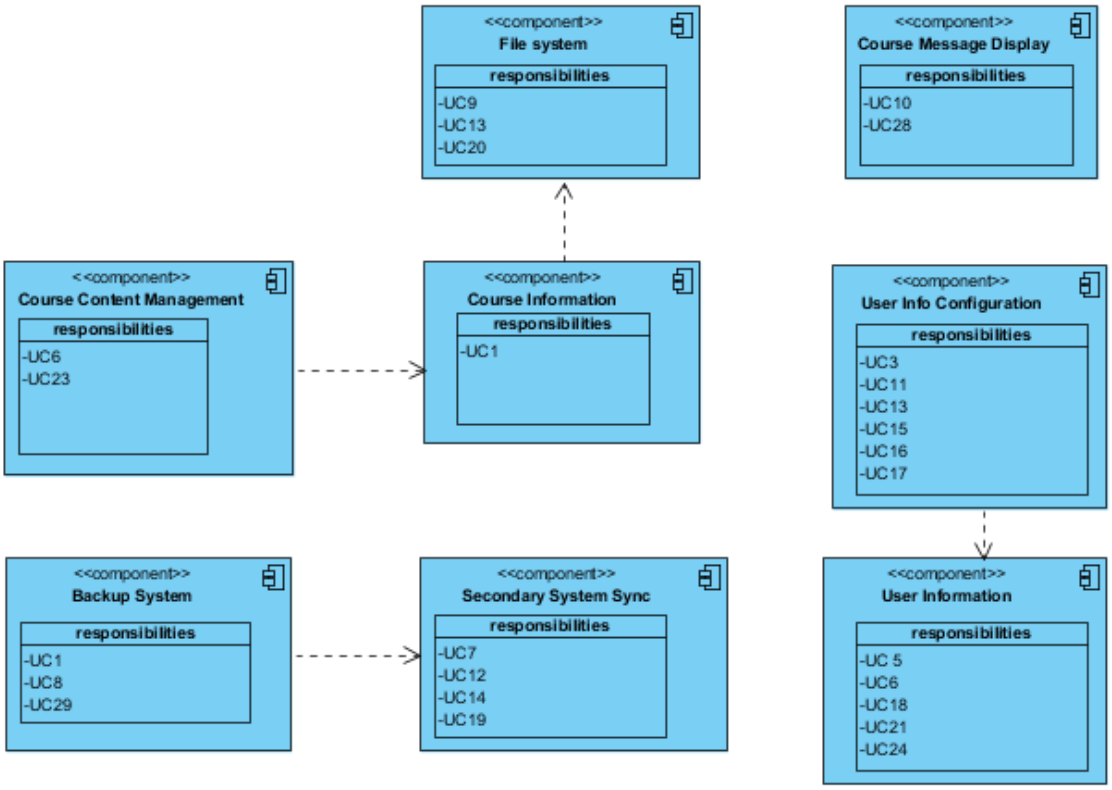

Figure 2: shows the domain objects that are instantiated for the use case model

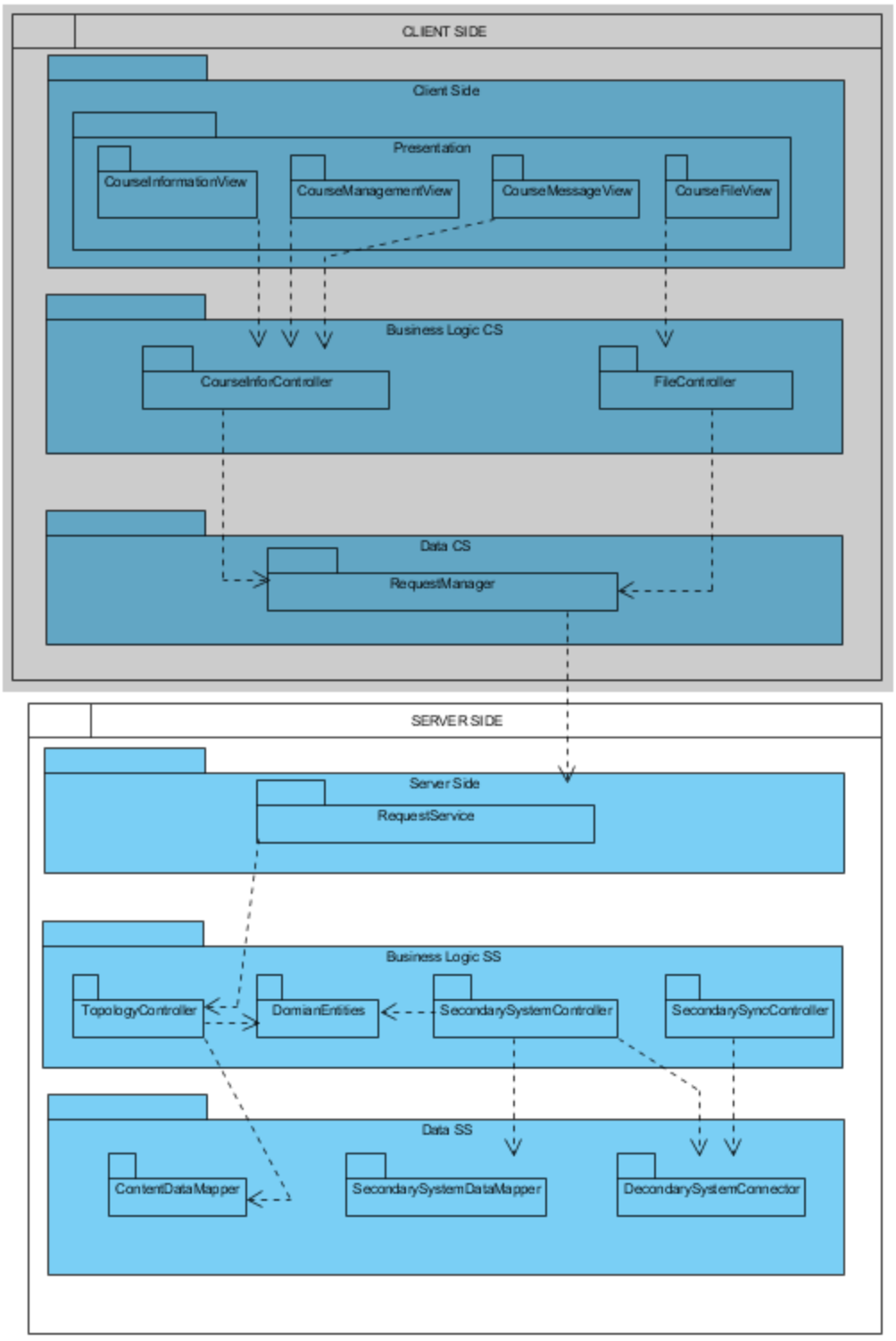

Figure 3: Modular view that support use cases

Element	Responsibility
CourseInformationView	Displays the course information from events present
CourseManagementView	Displays the course management system
MessageView	Displays the messages from the system
CourseFile	Displays the needed files for the courses taken
CourseInfoController	This is where all the course related material will be found
MessageController	This is where it will figure out if it's a group message or a private message
FileController	Provided the files needed for the course
StudentGrades	Updates the grade for the student from ID
Enroll	Checks the student enrollment for the course

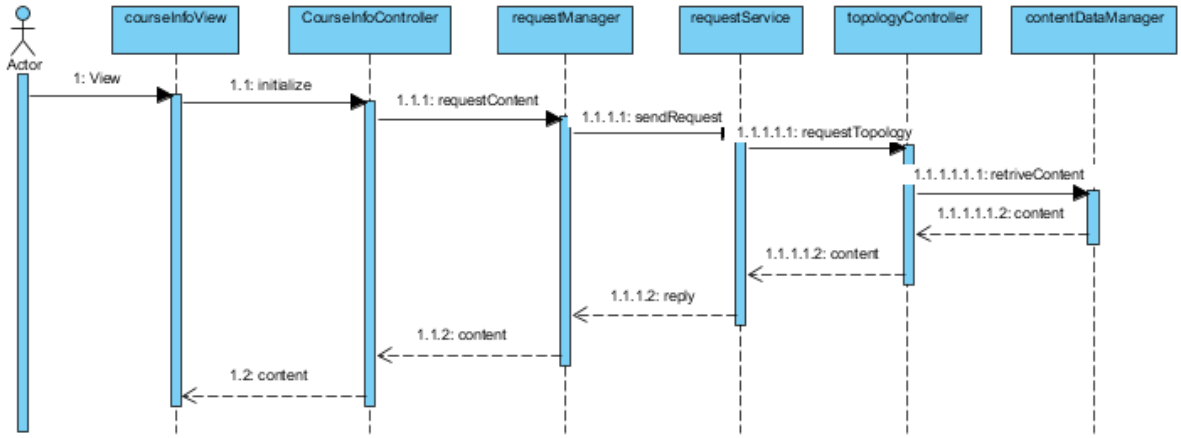

Figure 4: UC-1 Content system

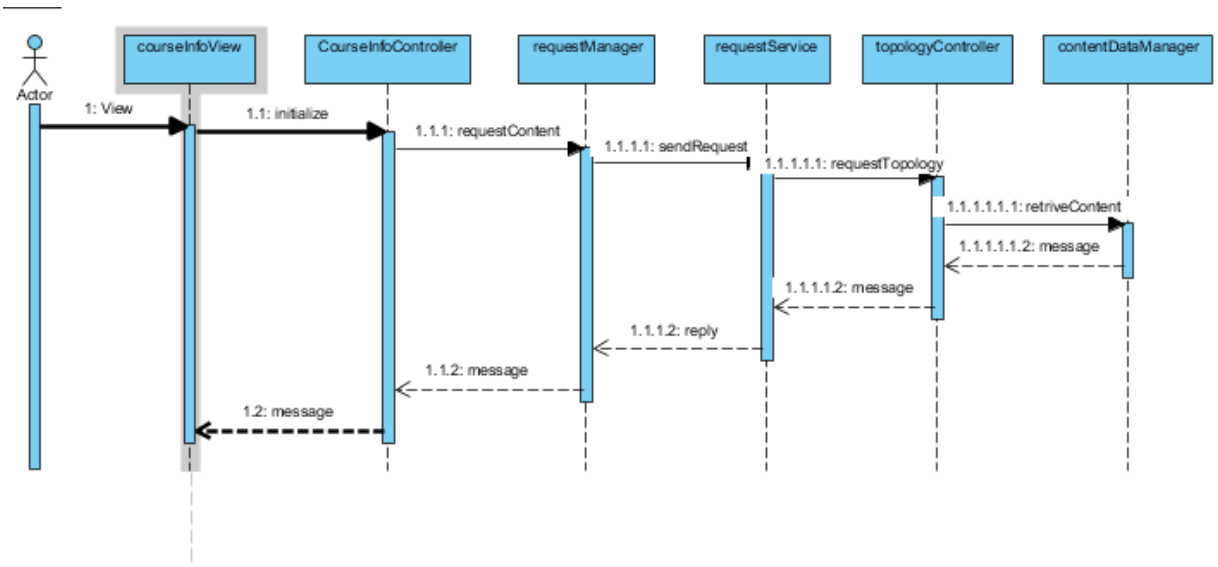

Figure 5: UC-2 messaging system

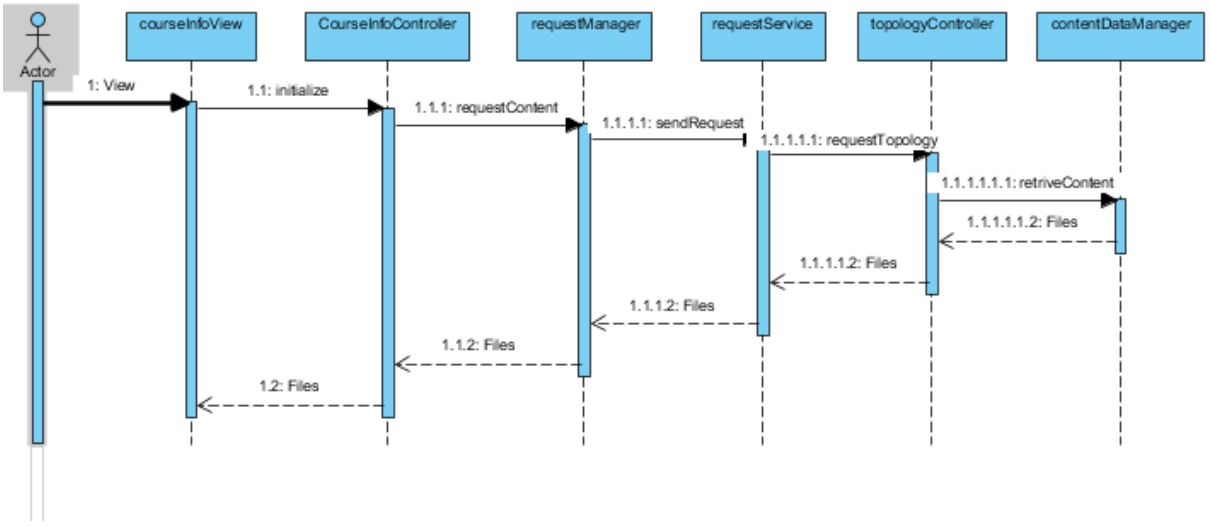

Figure 6: UC-2 files system

Step 7: Perform Analysis of Current Design and Review Iteration Goal

The decision made in this iteration an initial understanding of how functionality is supported in the system. The modules associated with the primary use cases aware identified by the architect, and the models associated with the rest of the functionality were identified by another team member. Also, as part of the module identification, a new architectural concern was identify and added to the Kanban board. Drivers that were completely addressed in the previous iteration are removed from the table.

Not Addressed	Partially Addressed	Completely Addressed	Design Decisions during the iteration
		UC-1	Modules across the layers and preliminary interfaces to support this use case have been identified
		UC-2	Modules across the layers and preliminary interfaces to support this use case have been identified
	UC-4		Discussion board provides for functionality but mechanism not created
		UC-7	Course content needs to be created
	QA-1		Each user can make their own changes
	QA-2		No design decision
		QA-4	Architecture for the web application
		QA-10	Web applications database
		CON-1	No relevant decision were made
	CON-2		Cloud server
		CON-3	No relevant decision were made
	CON-4		Physically structure the application using the 3 tier deployment pattern and isolate the data base by providing components in the data layer of the application server.
		CON-5	3 tier deployment pattern and isolate the data base by providing components in the data layer of application server
	CRN-1		Relational database is designed
	CRN-2		Selection of reference architectures and deployment pattern
	CRN-3		Languages are considered and now can easily be used to create team a team
	CRN-4		Unit testing modules are in places to complete basic database and architecture