ADD Iteration 2: Identifying structures to support primary functionality

A. Step 2 : Establish Iteration Goal by Selecting Drivers (Primary Usecases)

The goal of this iteration is to address the general architecture concern of identifying structures to support primary functionality.
Primary use cases: UC-5,19,20,21,22,31

B. Step 3:Choose One or More Elements of the System to Refine

1. In the previous iteration the element that was the entire university system. The refinement process was to decompose the system to different elements (User Workstation, Database Server, and Secondary University System.) In order to refine the elements of the system again, we would need to refine the Database server, user workstation and secondary university by decomposing it further.

C. Step 4:Choose One or More Design Concepts That Satisfy the Selected Drivers

Design decisions and location	Relation and Assumptions
Create a Domain Model	Before decomposition identify the major entities and their relationships
Identify Domain Objects	Each element of the application needs encapsulation in a self-contained block (i.e. a domain object)
Decompose Domain Objects into components	Domain objects are sets of functionality supported by elements within the layers. (i.e. the components and modules)
Use Web framework such as Django	Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. This framework was selected over the alternative spring framework due to the versatile and efficient nature of the framework.

D. Step 5 : Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

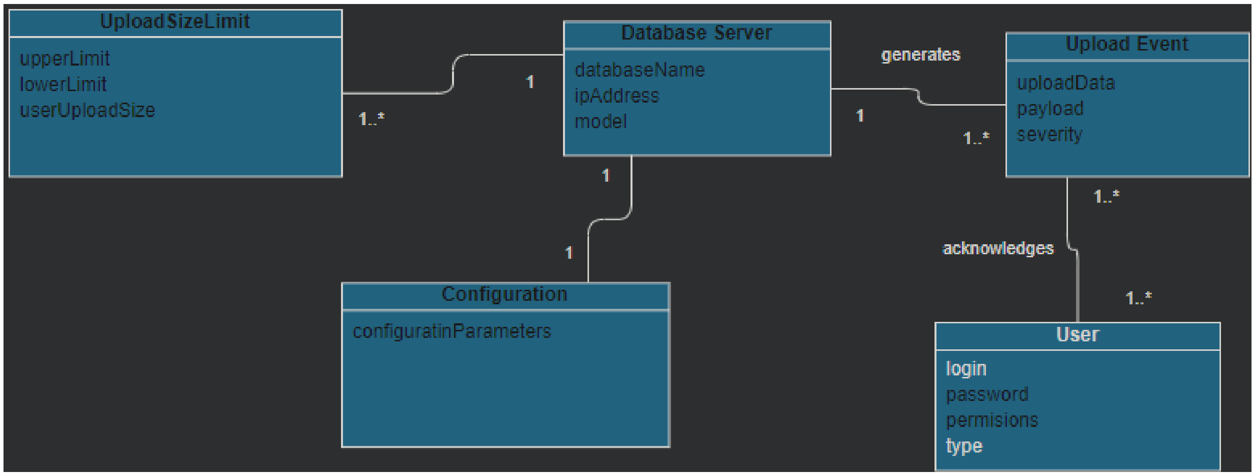
Design decisions and location	Rationale
Create a initial Domain Model	Initial domain model is created to help speed up the phase of design and helps model and identify the entities that identify the main use cases
Map use cases to domain objects	Instantiation of domain objects are created through analysis of the use cases. To address CRN-8
Decompose Domain Objects	-This techniques serves to ensure that the modules that support the

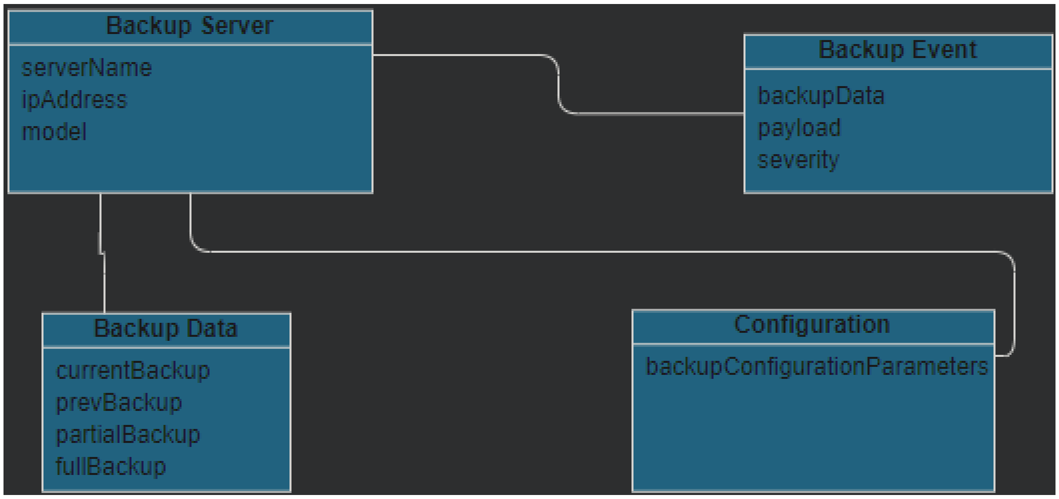
across layers to create layer specific modules with an explicit interface	functionalities are identified -We do this for the primary use cases so that we can also address CRN-8 in which we will make other team members address the rest of the modules -CRN-9: finding a way to reliably back up the server
Connect component associated with modules using Django	This framework gives the ability to support both the database and backup server

Element	**Responsibility**
NetworkStatusView	Displays the network representation and updates it when events are received.
RequestManager	Responsible with communication with server side logic
RequestService	Provides a facade that receives requests from the client
UploadController	Contains business logic related to the upload information
DomainEntities	Contains entities from the domain model
BackupEventController	Contains business logic related to the management of backup type events
DataCollectionController	Contains logic to perform data collection and storage of backups
FileDataController	Contains logic related to the upload file storage on the system
BackupDataMapper	Responsible for mapping backup data to a location in the server
EventDataMapper	Responsible for mapping the Event data to a location on the server
BackupServerConnector	Responsible for communication with the backup server.

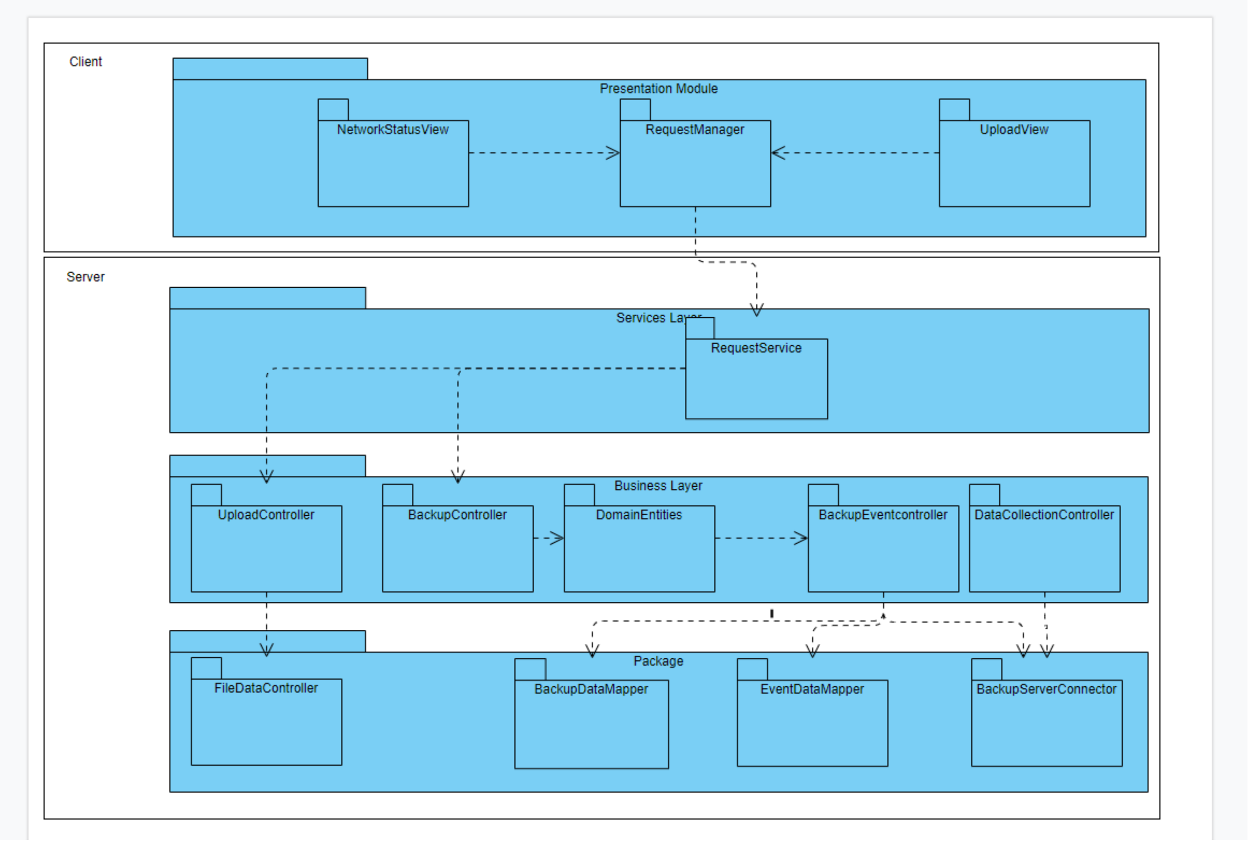
E. Step 6 : Sketch views and record design decisions

Domain Diagrams:

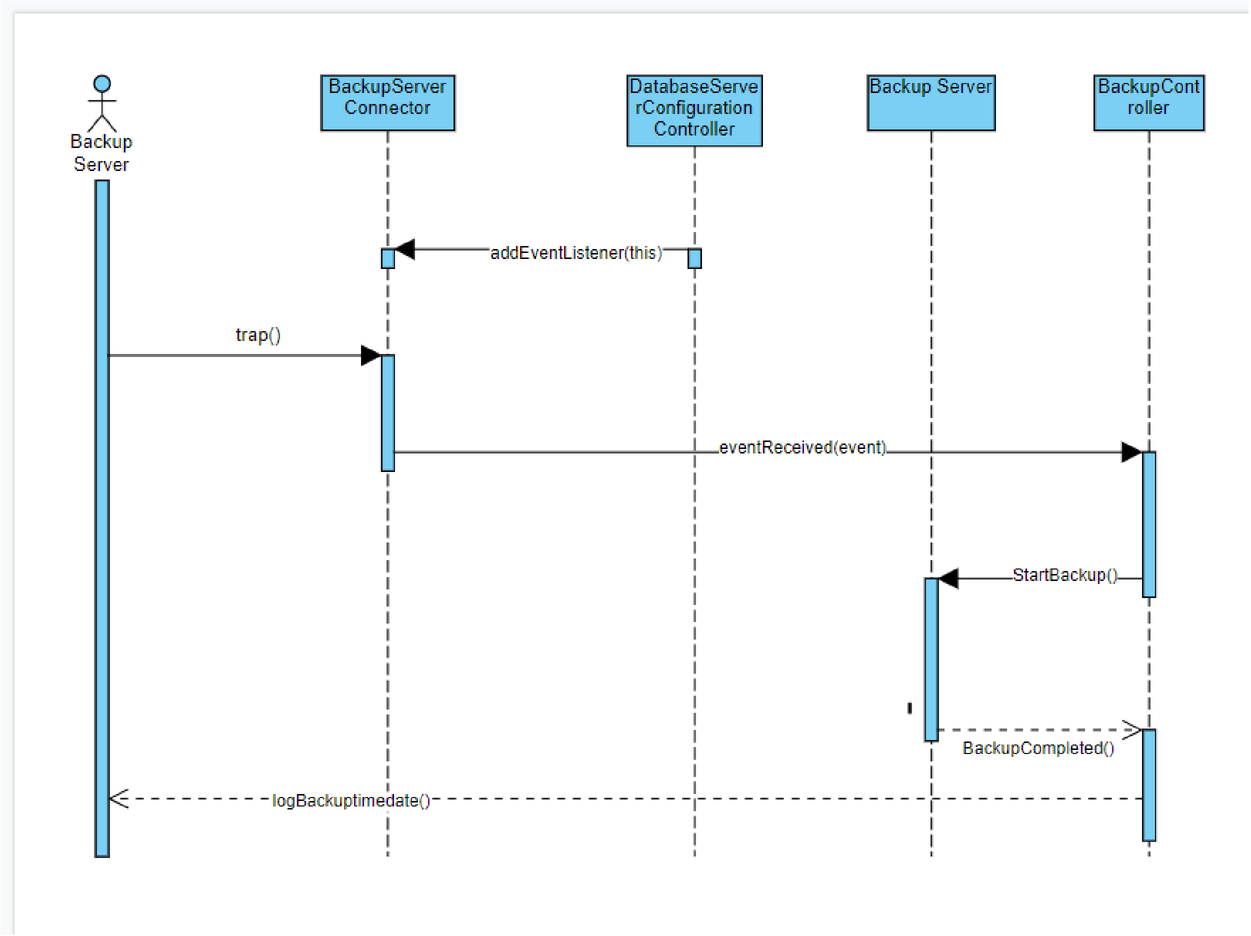




Modules that support primary use cases:

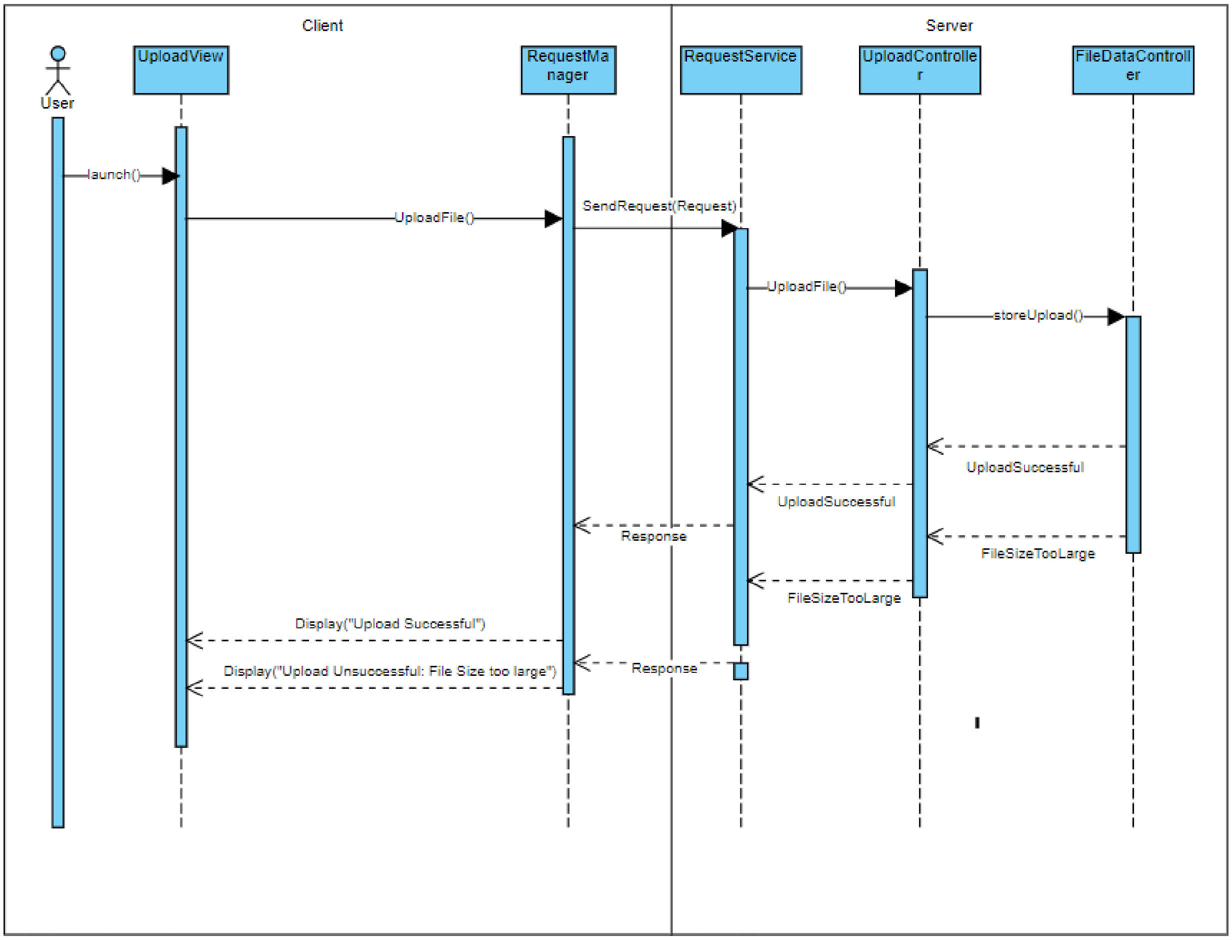

Client
Presentation Module
NetworkStatusView
RequestManager
UploadView
Server
Services Layer
RequestService
Business Layer
UploadController
BackupController
DomainEntities
BackupEventcontroller
DataCollectionController
Package
FileDataController
BackupDataMapper
EventDataMapper
BackupServerConnector

Sequence diagram for UC-19 and UC-313



Method Name	Description
boolean addEventListener(EventListener el)	This method allows components from the business logic to register themselves as listeners to events that received from the database server
Boolean eventReceived(Event evt)	This callback method notifies the clients that a new event has occurred
Boolean StartBackup()	This method starts backing up the system
Boolean backupCompleted()	This method sends a boolean value which indicates weather the backup process has been completed.
String logBackuptimedate()	This method saves the time it took to backup the system and displays it on the screen.

Sequence diagram for UC-20 and UC-21



Method Name0	Description
Boolean initialize()	Opens up the network representation so that users can interact with it
File uploadFile()	Requests server to upload a file
File storeUpload()	Stores the file on the database server

F. Step 7 : Perform Analysis of Current Design and Review Iteration Goal and Achievement of Design Purpose

Note:check concerns and constraints

Not Addressed	Partially Addressed	Completed Addressed	Design decisions made during the iteration
		UC-5	Modules across layers and main interfaces to support this use case was identified
		UC19-22	Modules across layers and main interfaces to support this use case was identified
		UC 31	Modules across layers and main interfaces to support this use case was identified
	QA-2,QA-4,QA-11		Elements thats support uc-5 has been identified
		QA-3	No relevant decisions made
	QA-6		No relevant decisions made
	QA-8		No relevant decisions made
CON-1-6			No relevant decisions made
CON-8-10			No relevant decisions made
	CON-11		Modules responsible for collecting data has been identified
	CON-12		Modules responsible for collecting data has been identified
	CON-13		Modules responsible for collecting data has been identified
	CON-14		Modules responsible for collecting data has been

			identified
CON-15-19			No relevant decisions made
CRN-1			No relevant decisions made
	CRN-2		No relevant decisions made
CRN-3			No relevant decisions made
	CRN-4		Modules responsible for the upload have been addressed
	CRN-5		Selected reference architecture establishes the modules that will some support this functionality
		CRN-6	Elements that support CRN-6 has been identified with Backup server module
		CRN-8	Modules responsible for collecting data has been identified