Associate Architect, 2022-A1
Comprehensive Evaluation Project (CEP)

SAMSUNG

Architecture Description

Personalized Learning Assistant

Software System for Adaptive and Personalized Learning

January 2022

Revision History

Version	Date	Author	Revisions Made
0.8	1/20(Thu) 9pm	박규준	(Interim) Defining the Skeleton Architecture
0.9	2/06(Sun) 9pm	박규준	(Pre-final) Designing Architecture with Viewpoints
1.0	2/21(Mon) 9pm	박규준	(Final) Architecture with All Activities Applied

삼성전자 첨단기술연수소

TABLE OF CONTENTS

Personalized Learning Assistant

1. Introduction

1.1. Purpose of the Document

The purpose of this document is to specify the architecture design for the target system. It describes all the essential architectural aspects of the target system including its structure, functional components, data components, their relationships, runtime behavior, and deployment.

1.2. System of Interest

Personalized Learning Assistant is a software system which provided by cloud service. The system services the comprehensive set of functionalities for personalized learning for trainees. The system has enough services for staffs, instructors and Teaching Assistant as well.

❑ Functionality for Program Managers

The system provides the functionality of registering training programs with essential information to run. Program manager can review all the registered program and re-organize program by changing its profile.

❑ Functionality for Course Managers

The system provides the functionality of registering courses with essential information. Course manager can browse all the courses and update the profile to cover a cohesive set of detailed topics.

❑ Functionality for Offering Managers

The system provides the functionality of registering offerings for occurrence of training program. Offering manager can browse and review all the registered training program and offering. Offering manager can manage offering schedule by changing registered offering's profile.

❑ Functionality for Instructors

The system provides the functionality of managing lectures for instructors.

❑ Functionality for Teaching Assistants

The system provides the functionality of helping 'lectures and learning activities' for Teaching Assistants.

❑ Functionality for Trainees

The system provides the functionality of personalized learning for trainees. The system provides learning strategy for individual trainee, which is aimed to maximize the effectiveness of learning. Each trainee can have desired level of achievement by recommended learning strategy.

1.3. Definitions, Acronyms, and Abbreviations

❑ PLA

A PLA is acronym for the Personal Learning Assistant. The PLA denotes the system which is described throughout this document.

❑ Personal Learning Recommendation Model (PLRM)

PLRM is acronym for Personal Learning Recommendation Model. The Model is recommends set of learning activities that might improve effectiveness of learning. The PLRM is gradually evolving model while individual trainee's achievement result provided. The goal of the PLRM is providing very effective guide line for each trainee.

❑ Personalization

A Personalization is the key feature of this system. The personalization is for individual trainee

❑ Course

A course is a cohesive set of detailed topics.

❑ Training Program

A training program is set of courses which is able to be offered or already in offering.

❑ Offering

An offering is a behavior that opens a training program to trainees.

❑ Learning Activities

Learning Activities are set of activities for the given lecture. All activities are evaluated without exception and graded results are utilized for following set of Learning Activities' optimization.

❑ In-class lecture

An in-class lecture is the legacy way of teaching done by an instructor. The in-class lecture is one-way from an instructor to all trainees. All trainee has same content, process and progress. The in-class lecture has lack of detailed assessment of each trainee and limited reflection on individual achievement level.

❑ HCI

An HCI is acronym for Human–Computer Interaction which focuses on the interfaces between people (users) and computers.

1.4. References

[Kim 22a] Soo Dong Kim, Associate Architect Program, 2022-A1, CEP Specification of Personalized Learning Assistant (PLA), Version 0.9, 삼성전자 첨단기술연수소, January 2022.

[Kim 22b] Soo Dong Kim, Associate Architect Program, 2022-A1, CEP Template for Personalized Learning Assistant (PLA), 삼성전자 첨단기술연수소, January 2022.

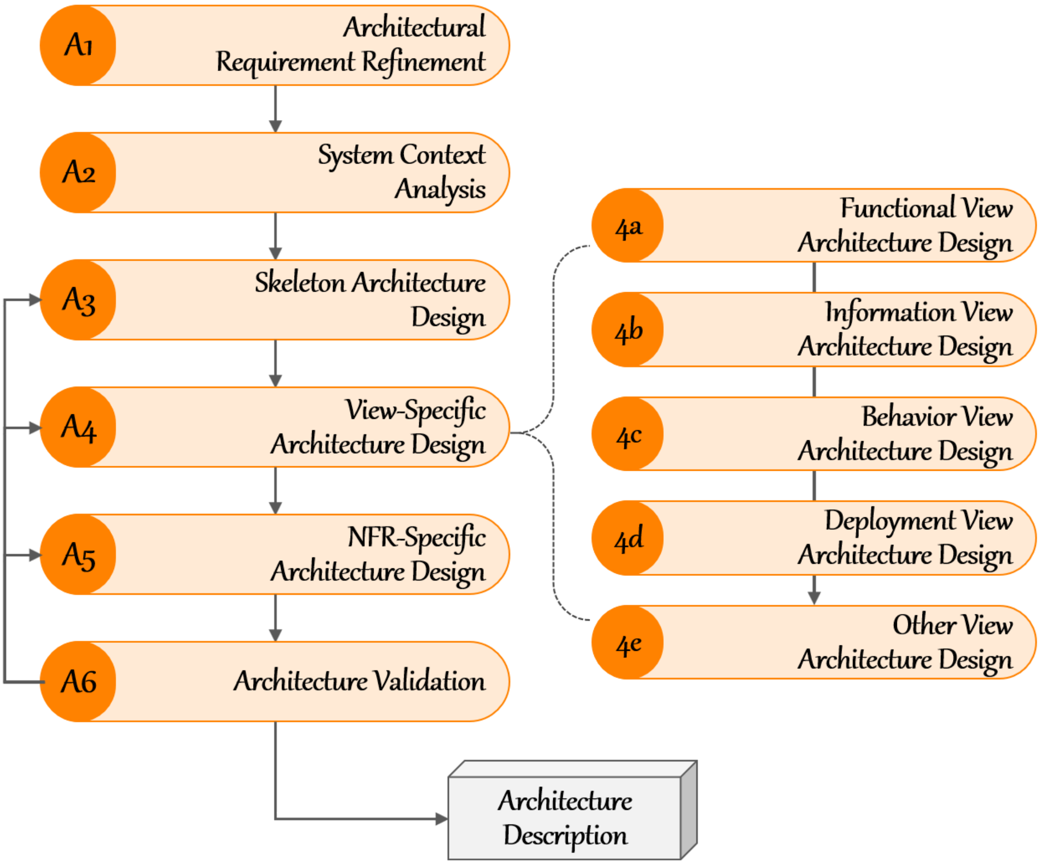
[Kim 22c] Soo Dong Kim, Architecture Design Process & Instructions, 2022-A1, Lecture Note #1, 삼성전자 첨단기술연수소, January 2022.

[Kim 22d] Soo Dong Kim, 2022-A1, CEP Specification of Personalized Learning Assistant (PLA), Version 1.0, 삼성전자 첨단기술연수소, January 2022.

[ISO 42010] ISO/IEC/IEEE, *Systems and software engineering - Architecture description*, pp. 46, Dec. 2011.

1.5. Process applied to Architecture Design

The process applied to designing software architecture in this sample solution is given [KIM 21c]. It consists of the following six activities.


A1 Architectural Requirement Refinement
A2 System Context Analysis
A3 Skeleton Architecture Design
A4 View-Specific Architecture Design
A5 NFR-Specific Architecture Design
A6 Architecture Validation
4a Functional View Architecture Design
4b Information View Architecture Design
4c Behavior View Architecture Design
4d Deployment View Architecture Design
4e Other View Architecture Design
Architecture Description

- ❑ Activity 1. Architectural Requirement Refinement

  This activity is to refine the given requirements for developing a target system.

  Software Requirement Specification should be refined before designing the architecture of the target system. The principles of requirement engineering can be well applied in this activity.

- ❑ Activity 2. System Context Analysis

  This activity is to analyze the given requirements for comprehending the target system before making any architectural decisions. The initial comprehension on the target system is specified in the context model of the target system.

- ❑ Activity 3. Skeleton Architecture Design

  This activity is to design the initial and high-level architecture of the target system, called skeleton architecture. The skeleton architecture mainly specifies the structural aspect of the target system and becomes the stable basis for making additional architectural decisions and defining more detailed architectural elements accordingly.

  The skeleton architecture can effectively be derived by utilizing architectural styles.

- ❑ Activity 4. View-specific Architecture Design

  This activity is to specify more detailed architectural elements for different views. It is advantageous to separate architecture design activities by view, backed by the principle of separate of concerns. Essential Views of Software Architecture are Functional view, Information view, Behavior view, and Deployment view. Utilize viewpoints.

- ❑ Activity 5. NFR-specific Architecture Design

  This activity is to refine the architecture with additional architectural decisions for each NFR item. Each NFR is thoroughly analyzed and effective architectural tactics are defined to fulfill the NFR. Then, the existing architecture is refined with the defined architectural tactics.

- ❑ Activity 6. Architecture Validation

  This activity is to validate the resulting architecture design of the target system for both functional and non-functional aspects.

  Architecture description becomes a concrete baseline document on which detailed system design are made for implementation. Hence, this activity is essential to confirm the fulfillment of both the functional and non-functional requirements.

1.6. Template used for Architecture Description

The template used for writing this architect description is given in [Kim 22b].

2. Activity 1. Architectural Requirement Refinement

This chapter describes the refinements made over the initial requirements of the target system.

2.1. [Step 1] Identify Stakeholders

A stakeholder can be an individual, a group, or an organization. Stakeholders have interests on the target system and concerns that are used as key drivers for designing architecture.

❑ Stakeholder 1. Client (as Software Distributor)

This represents the organization that will sponsor the development of the system and distribute the system to trainees and staffs.

❑ Stakeholder 2. Manager

This represents the staff users who manage the overall training program of PLA system.

❑ Stakeholder 3. Trainee

This represents the users who registered and takes personalized lectures on PLA system.

❑ Stakeholder 4. Instructor

This represents the instructors and teaching assistants who gives group-level lecture and helps learning activities on PLA system.

❑ Stakeholder 5. Lawyer

This represents the law specialist who provide the law- and policy-related advice for developing the system.

Describe the information of each stakeholder.

Stakeholder Group	Representative Name	Contact Information	Availability
Client	Linda Johnson	251-546-9442 Gulf Shores, AL	After 5pm, only on W, Phone Only
Manager	James Brown	415-546-4478 San Francisco, CA	Before Noon, MWF, Phone Only
Trainee	George Washington	415-546-4478 Irvine, CA	After 10am, only on WF, Phone Only
Instructor	Susan Tayler	945-679-7143 San Diego, CA	10-Noon, T. Th, Office Visits
Lawyer	David Harris	408-925-1352 San Jose, CA	All Day, M-F, Phone Only

2.2. [Step 2] Refining Functional Requirements (Revise)

Utilize the *SRS Refinement Table* to document the results of requirement refinement.

❑ Deficiency #1

Deficiency ID	FR.DEF.01	**Deficiency Type**	ambiguity	**Location**	SRS 4.5
Original Context	Step 1. Conduct a group or a personalized lecture on specific topic(s). Step 2. Apply personalized learning activities for the given lecture.				
Questioning	Learning activities have in-class exercise and laboratory. Then, is it needed to provide learning activities beginning of lecture or while instructor giving lecture?				
Refined Context	Recommendation implies lecture and activities. It is generated before conducting lecture. Learning activities are given at the start of lecture.				

❑ Deficiency #2

Deficiency ID	FR.DEF.02	**Deficiency Type**	incompleteness	**Location**	SRS 4.2
Original Context	“A training program consists of one or more courses to teach. Each course in a training program covers a cohesive set of detailed topics.”				
Questioning	A course is a set of topics. Does the topics consist of one or more lectures?				
Refined Context	“A training program consists of one or more courses to teach. Each course in a training program covers a cohesive set of detailed topics. Each topic in a course gives one or more lectures to trainee”				

❑ Deficiency #3

Deficiency ID	FR.DEF.03	**Deficiency Type**	ambiguity	**Location**	SRS 4.1, 4.3
Original Context	4.1. A Training program includes offering conditions. 4.3. An offering includes program status.				
Questioning	What is the meaning of offering conditions for an offering? What is the meaning of program status for an offering?				
Refined Context	4.1 “The offering conditions are minimal requirement to open a program. So that program may not be opened its scheduled date” added. 4.3 “The program status represents the opening and running status of offering. Offering manager can manage Offering by reviewing program status.” added.				

❑ Deficiency #4

Deficiency ID	FR.DEF.04	**Deficiency Type**	incompleteness	**Location**	SRS 4.5
Original Context	Step 5. Recommend Personalize the further subject content to study and learning activities to apply.”				
Questioning	Can the trainee refuse the recommended personalization?				
Refined Context	Step 5. Recommend Personalize the further subject content to study and learning activities to apply. Trainee can refuse Recommended Personalization, but cannot modify current personalization with his or her expectation. When Trainee refuse to apply, preset of lecture and activities are given and done. The evaluated result is also given to Machine Learning Model to generate next Personalization.				

2.3. [Step 3] Architectural Concerns

An architectural concern is a feature or a characteristic of the target system that are raised and defied by stakeholder(s). Hence, architectural concerns represent the stakeholders’ view on the target system and its architecture. Consequently, architectural concerns are expressed in the application domain language, rather than technology languages.

Many of the architectural concerns are requirements and expectations about the target system. And, in fact, many of the concerns in a target system may already be represented in the SRS of the target system in the forms of functional and non-functional requirement items.

The following concerns are acquired from the stakeholders. (revise)

❑ Concern-1. High Effectiveness of Personalizing the Learning Contents and Activities

The system should be designed to provide the high effectiveness of personalizing the learning contents and activities.

❑ Concern-2. High QoS of the Cloud Service

The system should be designed to provide a high QoS as a cloud service on the reliability, availability, scalability and performance.

❑ Concern-3. User Friendly GUI on User Client

The system should be designed and deployed with user friendly GUI.

The two concerns are mapped to the existing NFR items. One new NFR item is derived from the concern #3.

Merge newly derived NFR items and the NFR items of the SRS. We now have 3 NFR items.

❑ Concern-1. High Effectiveness → NFR-1. Effectiveness of Personalized Learning

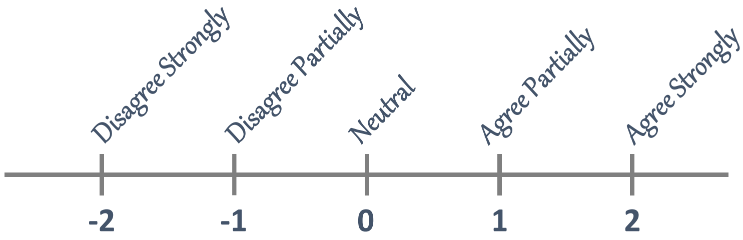
❑ Concern-2. High QoS → NFR-2. High QoS of Cloud Service

❑ Concern-3. User friendly GUI → NFR-3 Intuitive GUI of Client program

List all the NFR items in *NFR-to-Stakeholder Mapping* table.

NFR Items	**Relevance to Stakeholders**					**Average Relevance**	**Standard Deviation**	**Selection (Y/N)**
	Client	Manager	Trainee	Instructor	Lawyer			
NFR-1	2	0	2	2	1	1.4	0.91	Y
NFR-2	2	0	1	2	0	1	1	Y
NFR-3	1	-1	2	0	0	0.4	1.14	N

We use the following scoring systems for filling in the table.



We apply the following guidelines for choosing NFR items

- ❑ Case 1) High Average Relevance & Low Standard Deviation ⇒ Choose!
- ❑ Case 2) Medium Average Relevance & Low Standard Deviation ⇒ May choose with justification!
- ❑ Case 3) Medium Average Relevance & High Standard Deviation ⇒ May not choose with justification!
- ❑ Case 4) Low Average Relevance ⇒ Do not choose!
- ❑ Case 5) High Average Relevance & High Standard Deviation ⇒ Would not occur.
- ❑ Case 6) Low Average Relevance & High Standard Deviation ⇒ Would not occur.

As the result of quantitative assessment on NFR items, we choose NFR-1 and NFR-2.

2.4. [Step 4] Refine Non-Functional Requirements

Utilize the *SRS Refinement Table* to document the results of requirement refinement.

- ❑ Deficiency #1

Deficiency ID	NFR.DEF.01	**Deficiency Type**	Ambiguity	**Location**	SRS 5.1
Original Context	"the system should consider all the achievement-related measures of each trainee and reflect them on the personalization. Especially, the learning effectiveness of a given learning activity such as a laboratory should be measured for each trainee and reflected in further personalization."				

Questioning	Does the Learning Activities consist of measurable activities only? Or it can include non-measurable activities.
Refined Context	"But not all activity is measurable format." Added.

❑ Deficiency #2

Deficiency ID	NFR.DEF.02	**Deficiency Type**	Incompleteness	**Location**	SRS 5.2
Original Context	"The system should be scalable for potentially volatile invocation loads by users."				
Questioning	How much user invocation loads are expected in peak time?				
Refined Context	"The system should be scalable for potentially volatile invocation loads by users. The system should allow one million simultaneous access as peak performance."				

The resulting SRS now becomes more complete and well-aligned with stakeholders' concerns.

2.5. [Step 5] Write Refined Software Requirement Specification

The revised SRS is available here [KIM 22d].

3. Activity 2. System Context Analysis

This chapter specifies the context of the target system in terms of;

- ❍ Target System and Its Boundary
- ❍ Functionality provided by the system
- ❍ Information manipulated in the system
- ❍ Runtime behavior of the system

Additional type of the context can be described.

3.1. [Step 1] System Boundary Context

The target system may interact with external systems or other sources in the operational environment of the system. *System Boundary Context* describes the boundary of the system and elements in the environment which interact with the target system. This helps architect and developers to clearly understand the scope of the system.

3.1.1. Context Diagram

We use *Context Diagram*, i.e. Level 0 of Data Flow Diagram (DFD), which shows each tier of the target system as a process and relationships with its environment.

❑ Level 0 DFD for Context View

The SRS implies 2 tiers and accordingly the context diagram includes 2 processes, as shown in Figure 1.

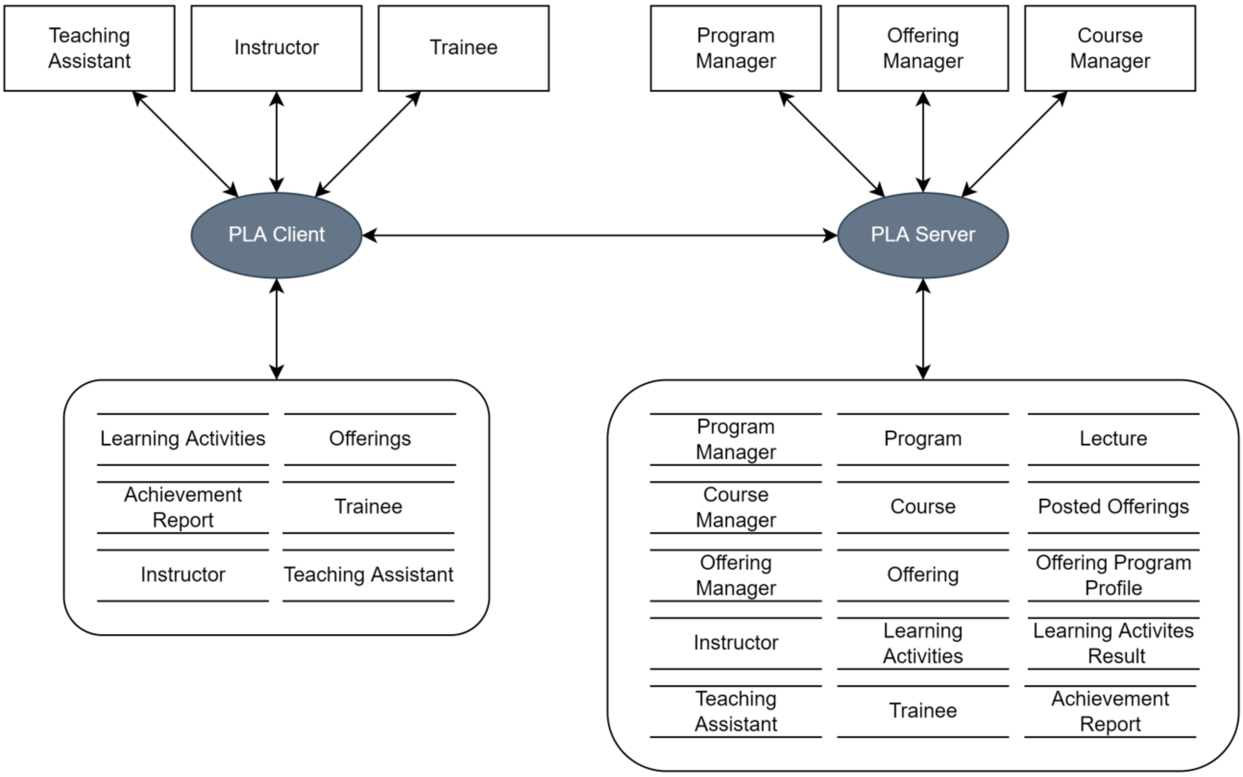


Figure 1. Context Diagram of PLA System

3.1.2. Description of Context Diagram

❑ Terminals

A terminal represents a source providing inputs to the system and/or a destination consuming outputs from the system.

❍ Program manager

This represents the manager who manages training program profile.

❍ Course manager

This represents the manager who manages course profile including lectures.

❍ Offering manager

This represents the manager who manages training program profile.

❍ Trainee

This represents the trainee of a PLA system who enrolls offering and takes recommended learning contents.

❍ Instructor

This represents the instructor who gives a group-level lecture.

❍ Teaching Assistant

This represents the teaching assistant who helps instructors and trainees' learning activities.

- ❑ Processes, i.e. Tiers

  The system consists of 2 tiers and each tier is represented as a process of DFD.

  - ❍ PLA Client
  - ❍ PLA Server

- ❑ Store

  PLA system maintains all the essential datasets, and most of them are also archived on the PLA Server.

- ❑ Data Flow

  Each arrow in the diagram indicates a flow of data, and the names of the data on arrows are omitted in the diagram.

3.2. [Step 2] Functional Context

3.2.1. Use Case Diagram for the System

The functional context of the target system can be well described with a use case diagram and descriptions of the use cases. A use case diagram shows the whole functionality of the target system. It is specified with Include actors, use cases, and their relationships.

The following use case diagram shows the whole functionality of the target system. It is specified with Include actors, use cases, and their relationships.

We do not attempt to show the control flow in the use case diagram; rather show only the use cases and invocation relationships. We do not consider tiers; rather we consider the whole system.

- ❑ Functional Groups

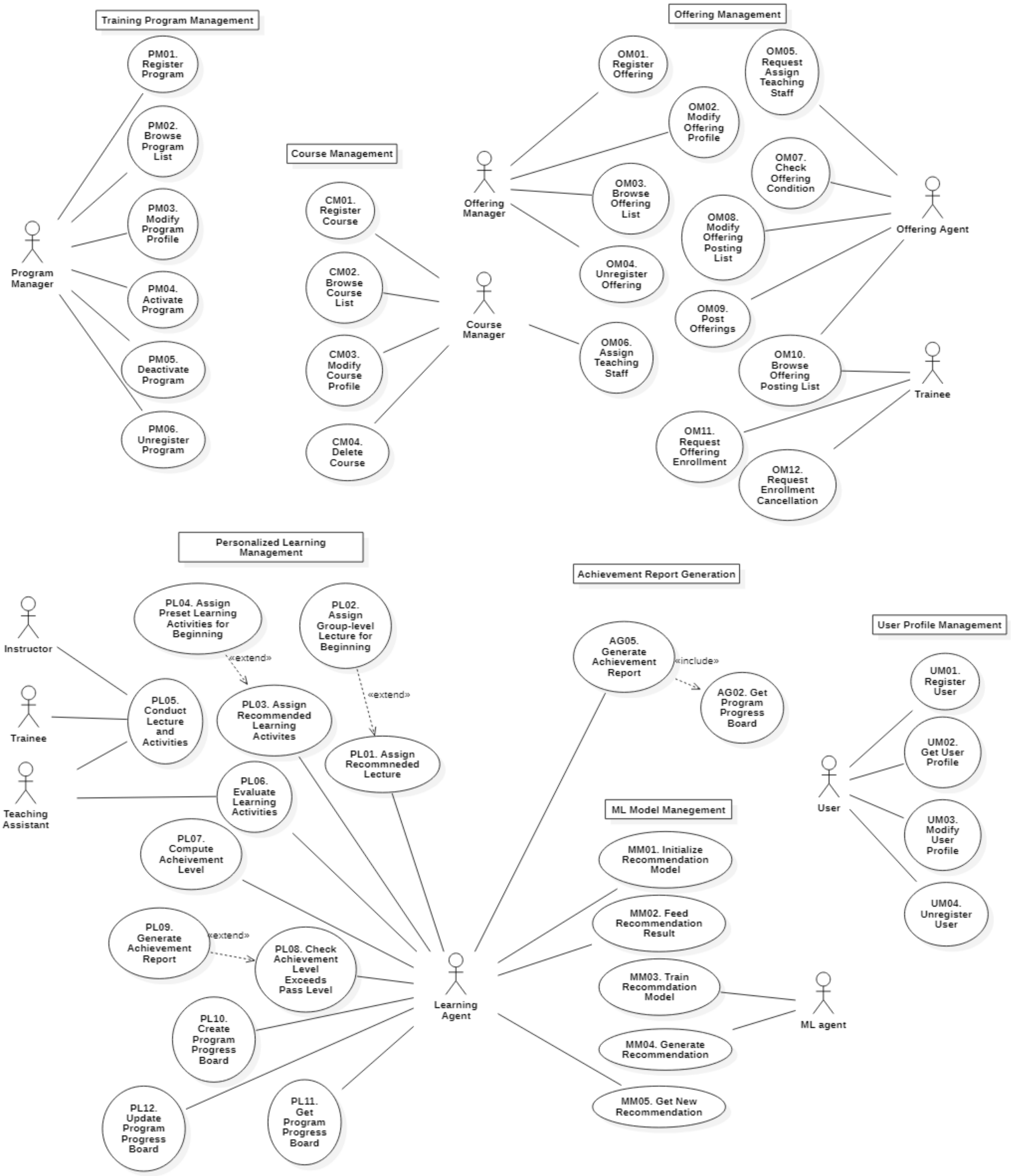
  We apply a scheme for numbering the use cases by considering functional groups. A functional group is a collection of *closely related* use cases. And, we assign a two-character prefix to each functional group. A use case diagram with use case identification numbers becomes easier to comprehend and to manage.

  Based on the given SRS, we identify the following functional groups and their prefixes.

  - ❍ User Profile Management -> UM
  - ❍ Training Program Management -> PM
  - ❍ Course Management -> CM
  - ❍ Offering Management -> OM
  - ❍ Personal Learning Management -> PL
  - ❍ Machine Learning Model Management -> MM
  - ❍ Achievement Report Generation -> AG

- ❑ Use Case Diagram

The following use case diagram shows all the use cases of the target system.


Training Program Management
PM01. Register Program
PM02. Browse Program List
PM03. Modify Program Profile
PM04. Activate Program
PM05. Deactivate Program
PM06. Unregister Program
Program Manager
Course Management
CM01. Register Course
CM02. Browse Course List
CM03. Modify Course Profile
CM04. Delete Course
Course Manager
Offering Manager
Offering Management
OM01. Register Offering
OM02. Modify Offering Profile
OM03. Browse Offering List
OM04. Unregister Offering
OM05. Request Assign Teaching Staff
OM06. Assign Teaching Staff
OM07. Check Offering Condition
OM08. Modify Offering Posting List
OM09. Post Offerings
OM10. Browse Offering Posting List
OM11. Request Offering Enrollment
OM12. Request Enrollment Cancellation
Offering Agent
Trainee
Personalized Learning Management
Instructor
Trainee
Teaching Assistant
PL04. Assign Preset Learning Activities for Beginning
PL02. Assign Group-level Lecture for Beginning
«extend»
«extend»
PL05. Conduct Lecture and Activities
PL03. Assign Recommended Learning Activites
PL01. Assign Recommneded Lecture
PL06. Evaluate Learning Activities
PL07. Compute Acheivement Level
PL09. Generate Achievement Report
«extend»
PL08. Check Achievement Level Exceeds Pass Level
PL10. Create Program Progress Board
PL11. Get Program Progress Board
PL12. Update Program Progress Board
Learning Agent
Achievement Report Generation
AG05. Generate Achievement Report
«include»
AG02. Get Program Progress Board
ML Model Manegement
MM01. Initialize Recommendation Model
MM02. Feed Recommendation Result
MM03. Train Recommdation Model
MM04. Generate Recommendation
MM05. Get New Recommendation
ML agent
User Profile Management
UM01. Register User
UM02. Get User Profile
UM03. Modify User Profile
UM04. Unregister User
User

Description of Use Case Diagram

❑ Actors as Human Users

There are several actors in the diagram that represent human users who use the system;

❍ User, Program Manager, Course Manager, Offering Manager, Trainee, Instructor, Teaching Assistant

❑ Actors as Software Agents

There are several actors in the diagram that represent software components which run in background mode. They can be implemented as daemon processes/threads.

❑ Use Cases

The use cases in the diagram are directly derived from the functional requirement of SRS. The name of each use case begins with 2 character-long prefix which indicates the functional group it belongs to.

The use cases in a functional group are placed together in the diagram. This will make easier to identify functional components during applying the functional viewpoint.

3.3. [Step 3] Information Context

3.3.1. Class Diagram for the System

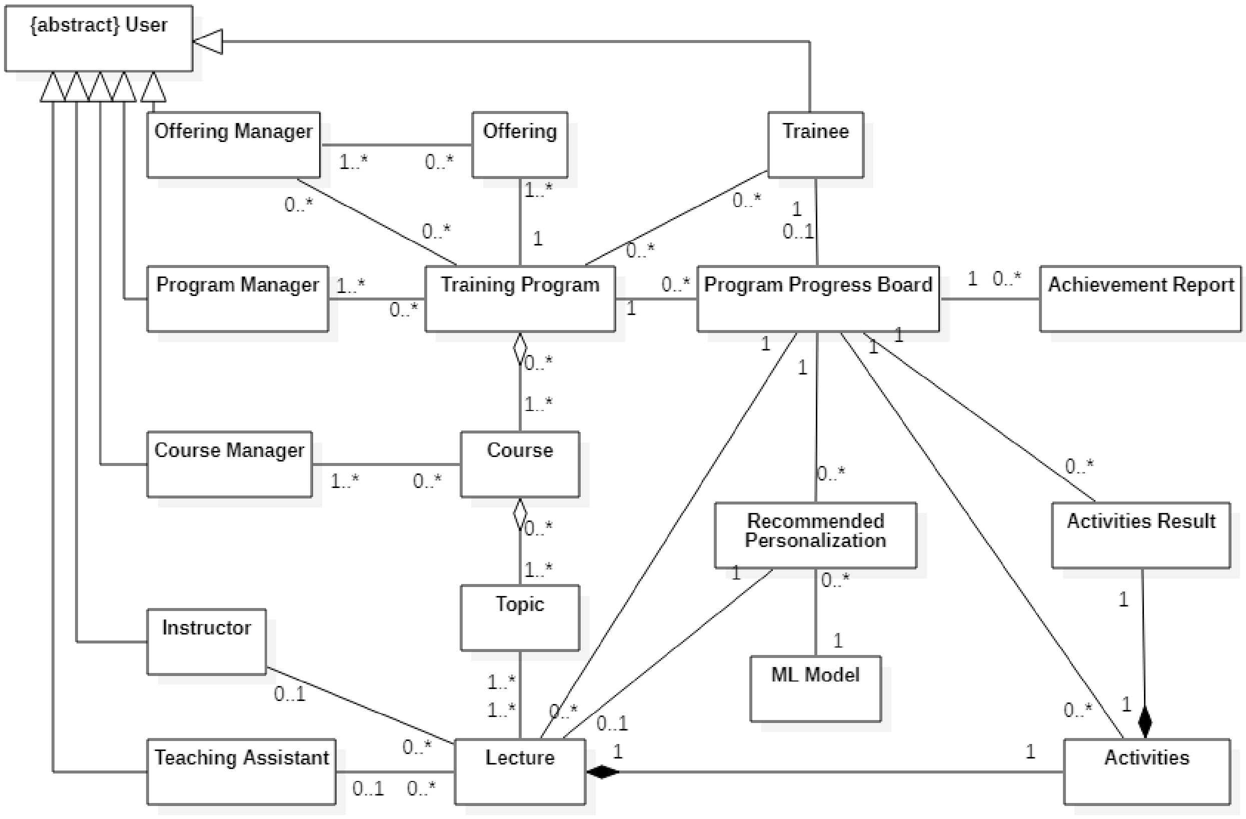
The information context of the system shows the datasets manipulated by the system. Class diagram can be effectively used to capture the information context.

For context modeling, show only the entity-type classes, their relationships, cardinalities on the relationships, and a few key attributes. Do not attempt to define methods for each class at this stage.

❑ Class Diagram

The class diagram for acquiring the information context consists of only classes and their relationships. A relationship is defined with cardinalities.

The context-level class diagram of the target system is shown in the following figure.



3.3.2. Description of Classes

❑ User Profile-related Classes

User is the superclass which captures the common property of its subclasses.

❍ *Manager* captures the information of the manager of a PLA system.

❍ *Instructor and Teaching Assistant* captures the information of the teaching staffs working on PLA system.

❍ *Trainee* captures the information of trainees who enrolling training program on PLA system.

❑ Program Progress Board

Program Progress Board captures the information of training program progress of each trainee. Every conducted lecture and activities information and evaluated result are stored this Board.

❑ Achievement Report

This class captures the information of evaluated achievement result of whole lecture in training program.

❑ Offering

This class captures the information of program offering.

❑ Training Program

This class captures the information of training program profile.

- ❑ Course

  This class captures the information of Course

- ❑ Topic

  This class captures the information of Topic

- ❑ Lecture

  This class captures the information of program offering

- ❑ Activities

  This class captures the content of Learning Activities.

- ❑ Activities Result

  This class captures the information of evaluation result of Learning Activities.

- ❑ Recommended Personalization

  This class captures the information of lecture and activity recommendation.

- ❑ ML Model

  This class captures the information of generated and/or deployed machine learning models.

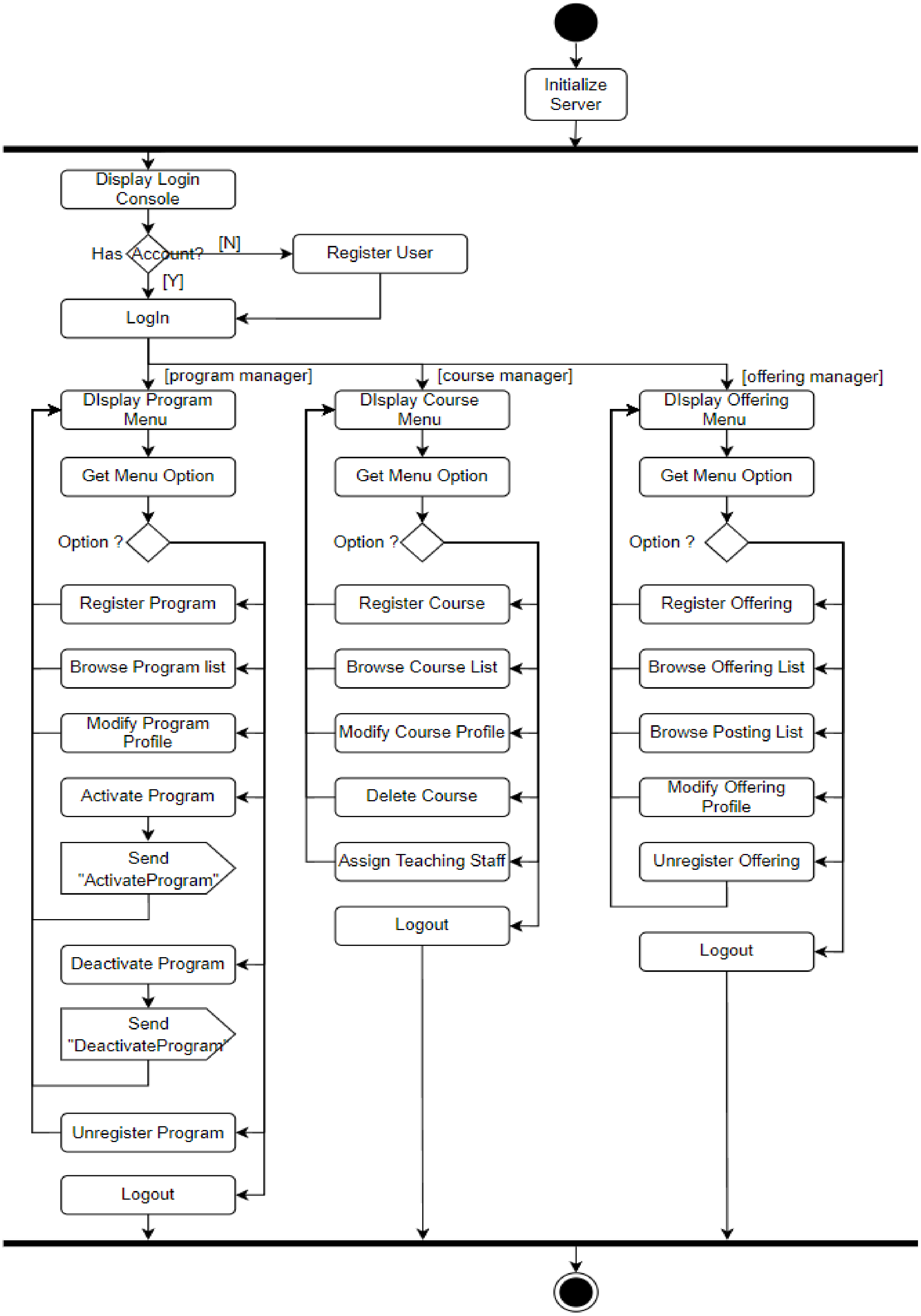
3.4. [Step 4] Behavioral Context

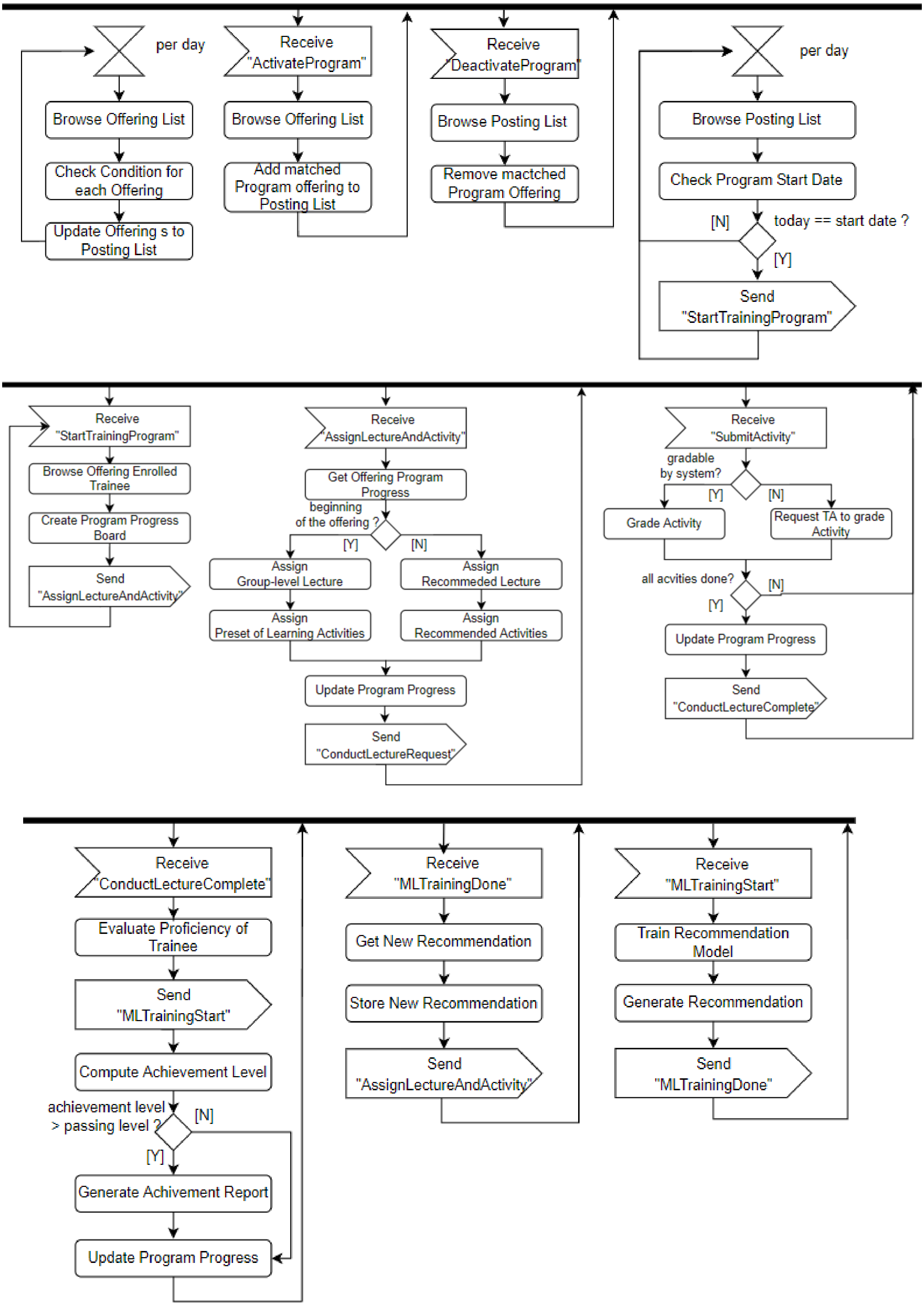
Behavioral context of the system shows the execution and control flow at runtime. Behavioral context may be more important for systems with complex workflows, parallel processing, and timing constraints.

3.4.1. Activity Diagram

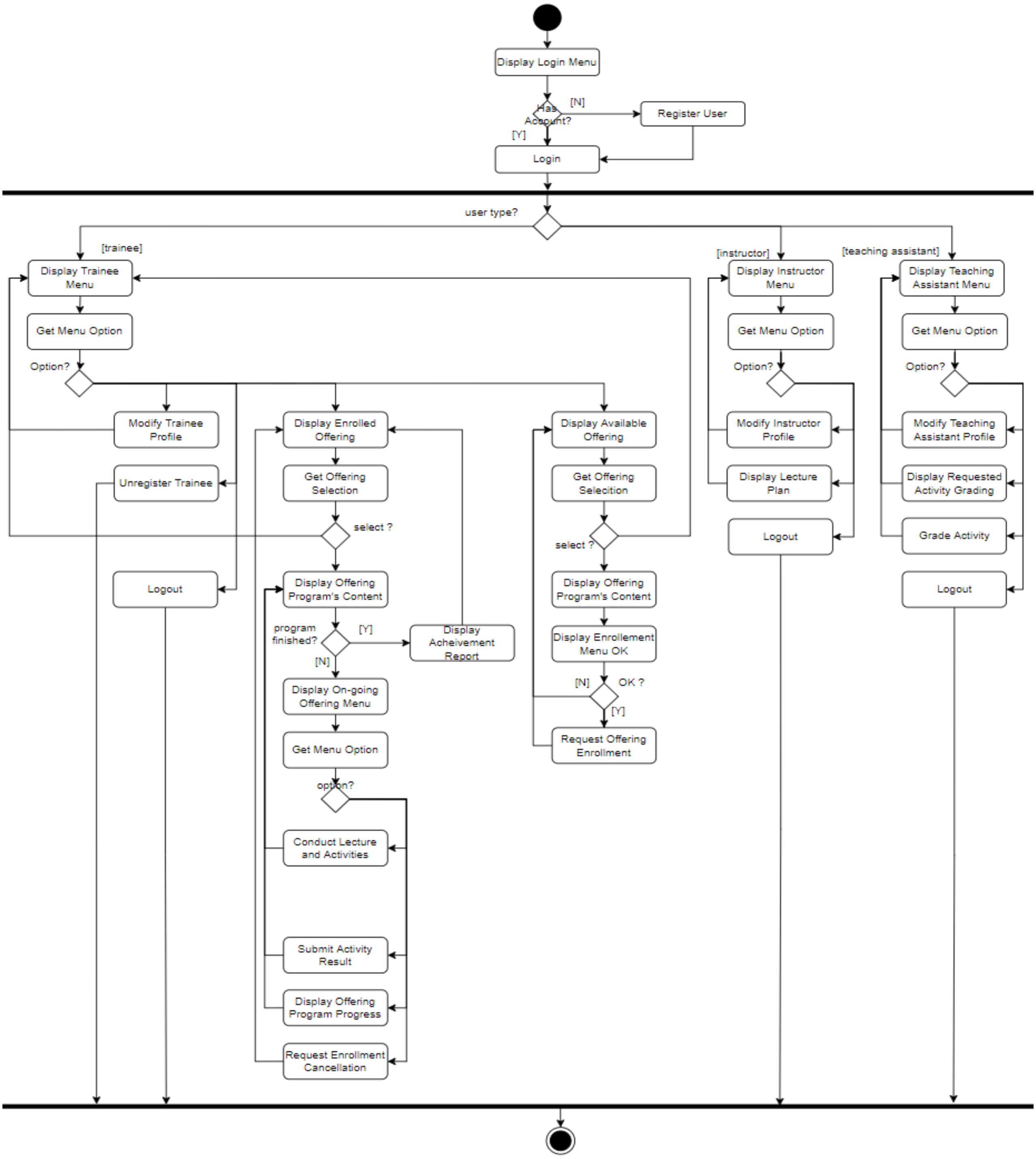
Activity Diagram can be used to define the overall control of the target system. There are Client and Server tiers in the target system, and we specify the context-level control flows of the all 2 tiers.

- Control Flow of PLA Server




per day
Browse Offering List
Check Condition for each Offering
Update Offering s to Posting List
Receive "ActivateProgram"
Browse Offering List
Add matched Program offering to Posting List
Receive "DeactivateProgram"
Browse Posting List
Remove mactched Program Offering
per day
Browse Posting List
Check Program Start Date
[N]
today == start date ?
[Y]
Send "StartTrainingProgram"
Receive "StartTrainingProgram"
Browse Offering Enrolled Trainee
Create Program Progress Board
Send "AssignLectureAndActivity"
Receive "AssignLectureAndActivity"
Get Offering Program Progress
beginning of the offering ?
[Y]
[N]
Assign Group-level Lecture
Assign Recommeded Lecture
Assign Preset of Learning Activities
Assign Recommended Activities
Update Program Progress
Send "ConductLectureRequest"
Receive "SubmitActivity"
gradable by system?
[Y]
[N]
Grade Activity
Request TA to grade Activity
all acvities done?
[N]
[Y]
Update Program Progress
Send "ConductLectureComplete"
Receive "ConductLectureComplete"
Evaluate Proficiency of Trainee
Send "MLTrainingStart"
Compute Achievement Level
achievement level > passing level ?
[N]
[Y]
Generate Achivement Report
Update Program Progress
Receive "MLTrainingDone"
Get New Recommendation
Store New Recommendation
Send "AssignLectureAndActivity"
Receive "MLTrainingStart"
Train Recommendation Model
Generate Recommendation
Send "MLTrainingDone"

- ❑ Control Flows of PLA Client


Display Login Menu
Has Account?
[N]
Register User
[Y]
Login
user type?
[trainee]
[instructor]
[teaching assistant]
Display Trainee Menu
Get Menu Option
Option?
Modify Trainee Profile
Unregister Trainee
Logout
Display Enrolled Offering
Get Offering Selection
select ?
Display Offering Program's Content
program finished?
[Y]
[N]
Display Acheivement Report
Display On-going Offering Menu
Get Menu Option
option?
Conduct Lecture and Activities
Submit Activity Result
Display Offering Program Progress
Request Enrollment Cancellation
Display Available Offering
Get Offering Selection
select ?
Display Offering Program's Content
Display Enrollement Menu OK
[N]
OK ?
[Y]
Request Offering Enrollment
Display Instructor Menu
Get Menu Option
Option?
Modify Instructor Profile
Display Lecture Plan
Logout
Display Teaching Assistant Menu
Get Menu Option
Option?
Modify Teaching Assistant Profile
Display Requested Activity Grading
Grade Activity
Logout

3.4.2. Description of the Diagram

- ❑ PLA Server

The control flow for PLA Server has two parts; menu-driven invocation-based behavior and event-driven behavior.

The invocation-based behavior includes the actions, i.e. use cases, that are explicitly invoked by users, and they run in synchronous manner.

The event-driven behavior waits the signal while the thread/process sleeping. It repeats the sequence of waiting and handling the signals.

❑ PLA Client

The control flow for PLA Client runs an invocation-based behavior mainly for interacting with trainees

3.5. [Step 5] Additional Contexts

Any additional contexts of the target system can be described.

❑ None

4. Activity 3. Skeleton Architecture Design

A skeleton architecture is a description of the structural aspect of the target system without fully describing the key components and their properties. It can be effectively derived by applying architectural style(s). Each architectural style is a named collection of architectural decisions that are applicable in a given development context.

4.1. [Step 1] Observe Architectural Characteristics

We make the following observation on the target system as a preparation for choosing candidate architecture styles.

❑ Consisting with Client Program and Server Tiers → Client-Server Architecture Style

The number of tiers for the target system is implied by the SRS as already described in the boundary context. PLA is consisted of PLA-Client and PLA-Server.

❑ Considering Event-based Delegation on Server → Event-Driven Architecture Style

To make loosely-coupled Event Handling, PLA server handles asynchronous and synchronous events with several thread. To give

❑ Conventional MIS Application → MVC Architecture Style

Both PLA Client and Server have 3 distinct roles of presentation for HCI, business logic, and data persistency management.

❑ High QoS of the Cloud Service → Dispatcher Architecture Style

PLA system provides High QoS with reliability, availability, scalability and performance. Dispatcher provides optimal server for PLA Client invocation.

4.2. [Step 2] Candidate Architectural Styles

With the architectural observation on the target system, we propose the following architectural styles;

❑ Client-Server architecture style

❑ Event-Driven architecture style

❑ MVC architecture style

❑ Dispatcher architecture style

4.3. [Step 3a] Applicability of 'Candidate 1. Client-Server Architecture Style'

The system consists of only two tiers; client tier and server tier. The server tier provides a common service to the client tier, and the client tier interacts with users. Multiple client applications are components, may share the functionality provided by the server tier.

4.3.1. Match on the Applicable Situation

Applicable Situations	Match	Demands on the System
Consist of only two tiers	❍	PLA server for managing the system and PLA client for invoking server functionality
Server tier provide common service	❍	PLA server provides common services for personalized learning
Multiple client applications	❍	Multiple trainee access to server via Client

4.3.2. Applicable Benefits

Advantages of the Style	Match	Benefits Applicable to the System
Allocate functionality to independent Client and server tier.	❍	Each PLA client and server can focus on its functionality and it gives reliability
Maintenance server (upgrade, and expand) is easy	❍	For increasing usage of client, server resource can be easily upgraded.

4.3.3. Handling Drawbacks

Cons of the Style	Match	Handling the Drawbacks
Client is not working without server. Server failure will propagate to Client directly.	❍	Server replication or load-balancing can solve this issue. With load-balancing system, PLA system can address this.
Communication Overhead between Client and Server	❍	With segmented Data transaction and Client cache could solve this.

4.3.4. Summary of the Applicability

Client Server architecture style is well applicable to the target system according to the justification. There is no significant issue which prevents the application of this style.

4.4. [Step 3b] Applicability of 'Candidate 2. Event-driven Architecture Style'

The target system can be well configured with the Event-driven architecture style. This is applied PLA server's Learning course work.

4.4.1. Match on the Applicable Situation

Applicable Situations	Match	Demands on the System
The system consists of event emitters and event sinks. Occurrence of events invoke specific functionality of events.	❍	PLA system use various event signal to service specific functionality.
The communication between event emitter and event sinks is asynchronous.	❍	PLA system uses asynchronous event handling to have parallelism, modularity and extendibility. The communication between event emitter and event sinks is asynchronous.

4.4.2. Potential to benefit the *Pros*

Advantages of the Style	Match	Benefits Applicable to the System
Decoupling between system, only interact with Event message	❍	Handler is separated as a part and the part can be independently implemented.
Scalability is served through independent and decoupled event process.	❍	Additional request of event, System can easily add new event handler and provide event handling in parallel

4.4.3. Potential to handle the *Cons*

Cons of the Style	Match	Handling on Cons on the System
Hard to detect and trace error state	X	PLA system does not have monitoring service for health. It is hard to address problem immediately.
Difficulty of understanding whole system flow	❍	PLA system does not complex event algorithm. The number of events is understandable.

4.4.4. Summary of the Applicability

Event-driven Architecture style is well applicable to the target system according to the justification. There is no significant issue which prevents the application of this style.

4.5. [Step 3c] Applicability of 'Candidate 3. MVC Architecture Style'

The target system can be well configured with the three layers of MVC architecture style. This is applied to both Client and Server.

4.5.1. Match on the Applicable Situation

Applicable Situations	Match	Demands on the System

The system can ideally be divided into 3 distinct roles of View, Control and Model.	❍	PLA Client and Server has View for multiple types of User, Control for its business logic and Model data.
This system is used for presenting Model Data as number of different types of View	❍	PLA system has multiple type of User and each user interact with system with different view.
View and Control are frequently changed.	❍	PLA system provide recommended lecture content. By changing recommendation, view and Control frequently notified.

4.5.2. Potential to benefit the *Pros*

Advantages of the Style	Match	Benefits Applicable to the System
Various View are implemented with one Model.	❍	PLA system provides different View for each types of Users with one Model.
Guarantee Synchronization of all independent View and Control	❍	Model changes immediately propagated to all Views and Controls and represented.

4.5.3. Potential to handle the *Cons*

Cons of the Style	Match	Handling on Cons on the System
Difficulty of allocation functionality to appropriate component	❍	PLA system, each components' role is clear. This problem is not expected.
Each View can be notified unexpected Model changes. Not all View has concern with Model changes.	❍	This is not critical. The majority of Model changes are Trainee related, so most of concern is about View, respectively, and it is essential.

4.5.4. Summary of the Applicability

MVC Architecture style is well applicable to the target system according to the justification. There is no significant issue which prevents the application of this style.

4.6. [Step 3d] Applicability of 'Candidate 3. Dispatcher Architecture Style'

The target system can be well configured with the Dispatcher architecture style. This is applied between Client and Server for high QoS.

4.6.1. Match on the Applicable Situation

Applicable Situations	Match	Demands on the System
The system adopts Dispatcher middle layer between client and server.	❍	PLA Client and Server has performance impact. High QoS needs to meet for Stakeholder's concern. PLA system has high demand of Load-balancing.
For high scalability and availability in Distributed environment, this system maintains multiple servers those are considered as One Server at Client.	❍	PLA system can have multiple Server with Dispatcher.

4.6.2. Potential to benefit the *Pros*

Advantages of the Style	Match	Benefits Applicable to the System
This system gives high availability	❍	PLA system provides 24/7 available system with client-dispatcher-servers. It is endurable to server failure.
This system provides transparency of Server location.	❍	This system is easily expanded. PLA system gets maintenance benefit under multiple server environment.

4.6.3. Potential to handle the *Cons*

Cons of the Style	Match	Handling on Cons on the System
Difficulty of changing dispatcher interface	Δ	The interface of PLA system would be fixed before deployment. When the change is happened, we can apply design pattern like adapter to fix interface mismatch.
Inefficient connection manner between client-dispatcher-server. It provides an explicit and bypass connection	Δ	This is not critical for PLA system. Even the system gets this inefficiency, we take large advantage with dispatcher architecture style application.

4.6.4. Summary of the Applicability

Dispatcher Architecture style is well applicable to the target system according to the justification. There is no significant issue which prevents the application of this style.

4.7. [Step 4] List of Selected Architecture Styles

All the candidate architecture styles are chosen for defining the skeleton architecture.

❑ Client-Server Architecture Style

❑ Event-driven Architecture Style

❑ MVC Architecture Style

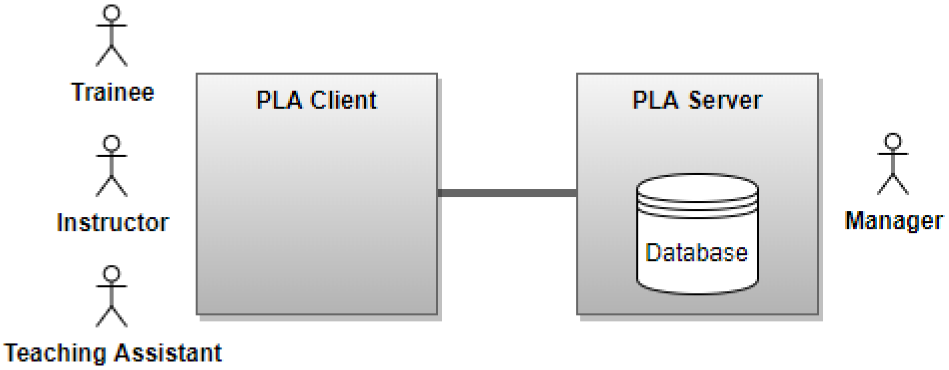
❑ Dispatcher Architecture Style

4.8. [Step 5] Integrating Architecture Styles

We apply the selected architecture styles incrementally.

4.8.1. Applying Client-Server Architecture Style

We define Client and Server tiers for the target system by considering the main-functionalities. A PLA client is user application running on mobile device and various OS platform. Web-browser is not considered as a tier since it is just a software tool for browsing web contents.



- ❑ PLA Client Tier

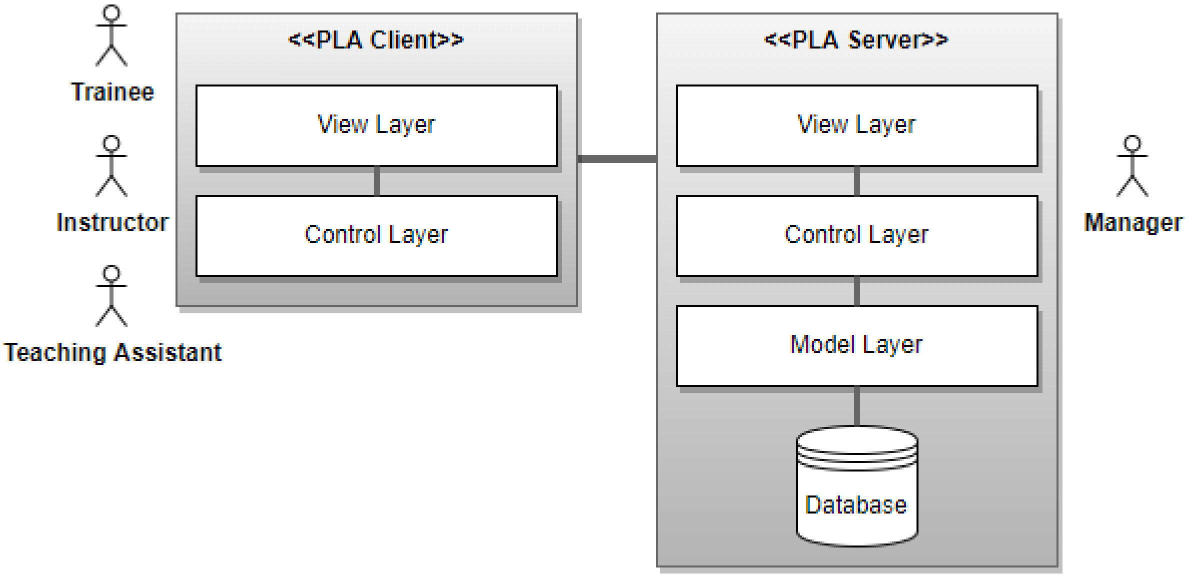
  This tier provides the functionality of PLA system access and running Personalized Training Program for Trainee, Instructor and Teaching Assistant.

- ❑ PLA Server Tier

  This tier provides the functionality of Training program and course management. The Server also provides the functionality of Training Program offering. The Server opens API to allow client access to use the core functionality for Personalized Learning.

4.8.2. Applying MVC Architecture Style

We define that PLA Client and Server could be configured with MVC Architecture Style. We define View and Control for Client and View, Control and Model for Server. PLA Client's Model reside in PLA server.



- ❑ PLA Client Tier

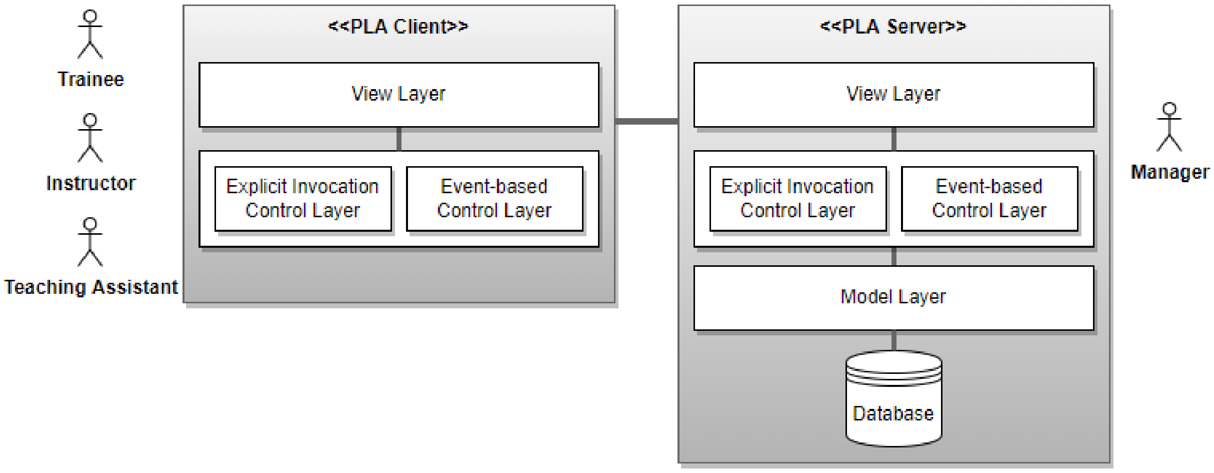
PLA Client is typical application which communicates with User and Server. By applying MVC Architecture Style, we can easily develop new type of application. A Client View interact with User and a Client Control takes User input and communicates with Server. The PLA Client does not have Model layer inside. A Client Control interact with Model in Server via Network Connection

- ❑ PLA Server Tier

  PLA Server provides several Views for Manager types. Thus, we could compose Server with MVC Architecture Style. A Server Control handles core business logic for Personalized Learning including ML task.

4.8.3. Applying Event-driven Architecture Style

For Both PLA Client and Server, Event-driven Architecture Style is applied.


Trainee
Instructor
Teaching Assistant
<<PLA Client>>
View Layer
Explicit Invocation Control Layer
Event-based Control Layer
<<PLA Server>>
View Layer
Explicit Invocation Control Layer
Event-based Control Layer
Model Layer
Database
Manager

- ❑ PLA Client Tier

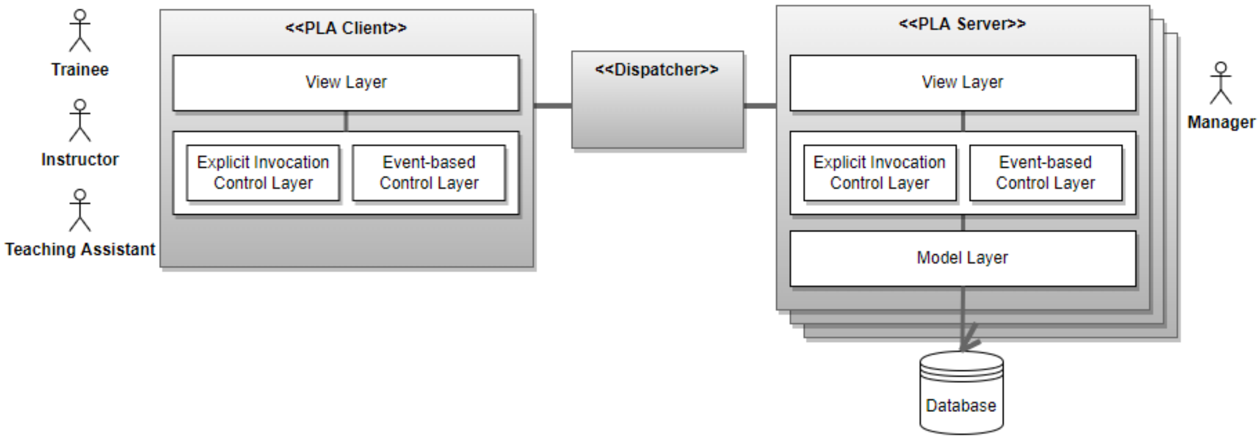
  By running Training Program, each progress on lecture generates output data. The output updated to server and it is used the core logic for Personalized Learning. Throughout this process, specified set of event handling is requested to Server and Client gets the response as event also.

- ❑ PLA Server Tier

  Server Control layer is consisted of several threads to support both synchronous and asynchronous handling of invocation. It enables parallel processing of external Client events and internal Server thread events.

4.8.4. Applying Dispatcher Architecture Style

PLA system needs load-balancing for high QoS. We define Dispatcher between PLA Client and Server for better reliability, scalability, availability and performance.



- ❑ PLA Client Tier

  PLA Client has interface to Dispatcher only. It does not have direct Server connection. Every request to Server is bypass-ed through Dispatcher. PLA Client act like there is only one server in opposite side.

- ❑ Dispatcher

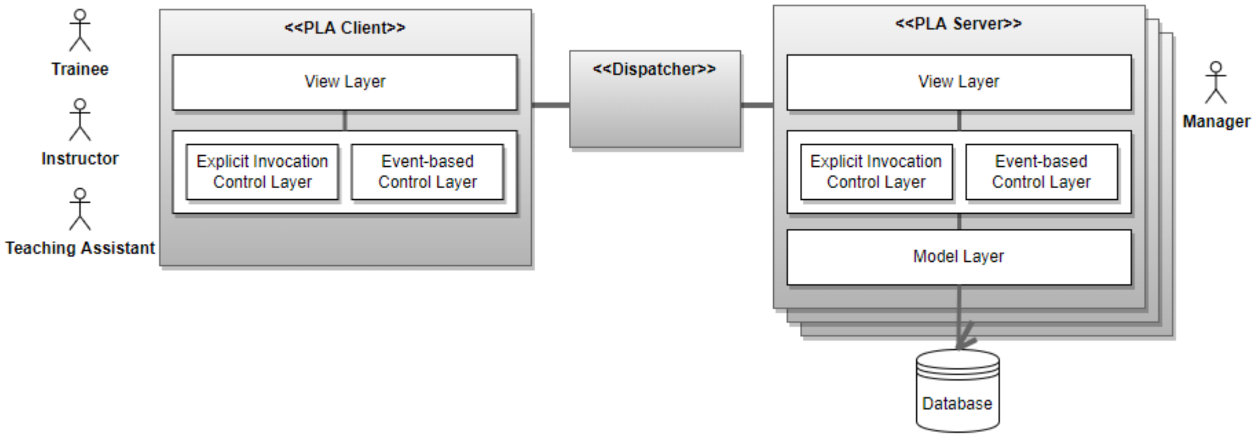
  Dispatcher knows every Server's availability. For Client invocation, Dispatcher establishes connection between Client and Server. Dispatcher chooses the optimal Server for connection.

- ❑ PLA Server Tier

  Server Control layer is consisted of several threads to support both synchronous and asynchronous handling of invocation. It enables parallel processing of external Client events and internal Server thread events.

4.8.5. Resulting Skeleton Architecture

The resulting skeleton architecture of applying all the selected architecture styles is shown in the following figure.



The resulting architecture shows the application of the selected architecture styles. And, it serves a stable basis on which view-specific architectural designs can be appended.

4.8.6. Interactions of the Skeleton Architecture

Define interaction paths among places in the skeleton architecture. An interaction path can be casual dependency or persistent relationship. It provides paths for making function calls or sending messages for communications among components.

❑ Interaction Paths derived from the Styles

All the interaction paths defined in each style are adopted in the skeleton architecture.

❑ Additional Interaction Paths

None

4.9. [Step 6] Strength and Limitations of the Skeleton Architecture

4.9.1. Strengths

Specify the advantages of the proposed skeleton architecture.

❑ Separation of Concern

Each component or layer represent a unique and separate concern. It yields a logically well-defined architecture with high modularity.

❑ Complexity of the System Design and Implementation

Due to the independence of each component or layer, the complexity design is low, and effort to implement the system can be greatly reduced.

❑ High Maintainability

Due to the key principles applied to designing the skeleton architecture, the impact of modification would be minimal.

4.9.2. Drawbacks

Specify the drawbacks and risks of the proposed skeleton architecture.

❑ Not Anticipated.

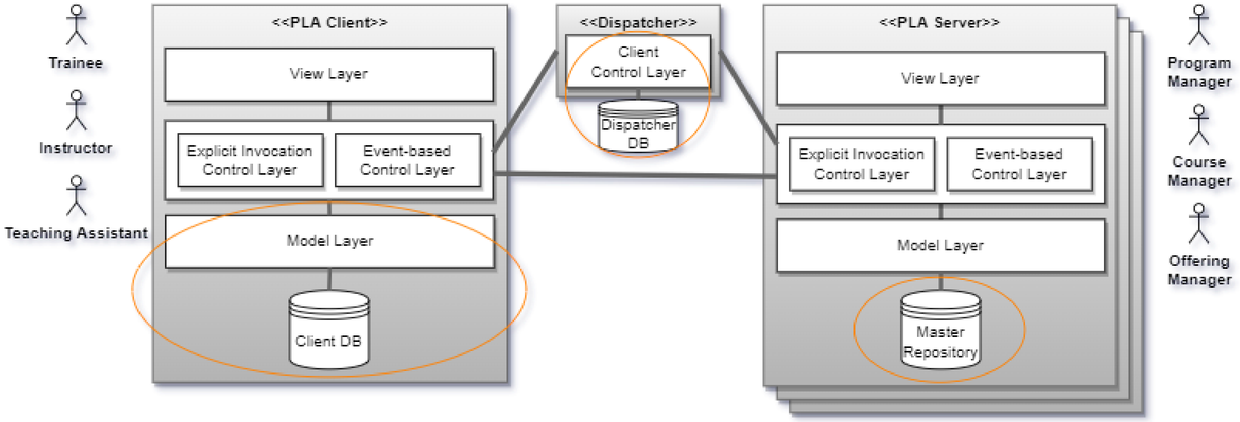
4.10. Revised Skeleton Architecture

Revised skeleton architecture of applying all the selected architecture styles is shown in the following figure.

- ❑ For maintaining frequently updated data on PLA Client, Model layer and Client DB added.

  Database for Dispatcher was omitted, it is added for clarification.

  Server's master repository got into the server because this system does not have additional data base server tier.



5. Activity 4. View-specific Architecture Design

This chapter describes the results of applying essential architecture viewpoints. The skeleton architecture is now refined with additional architectural decisions made with viewpoints.

5.1. Functional View

5.1.1. [Step 1] Observe Functional Characteristics

❑ Functionality of Managing Various Profiles

The target system has various profiles to manage; program manager, course manager, offering manager, instructor, teaching assistant and trainee.

❑ Functionality of Managing Personal Learning

The target system has process for personalized learning.

❑ Functionality of Optimizing Personalized Learning Contents through Machine Learning

The target system has machine learning model to get optimized learning recommendation. It is repeatedly evolved every iteration in process of personalized learning.

❑ Functionality of Generating Achievement Report

The target system has lots of offerings and each trainee of an offering has a report as offering output. It is generated the end of offering course-work

5.1.2. [Step 2] Refine Use Case Diagram

Before identifying functional components, we need to refine the context-level use case diagram with greater details and refinements. We first define the following functional groups for the target system.

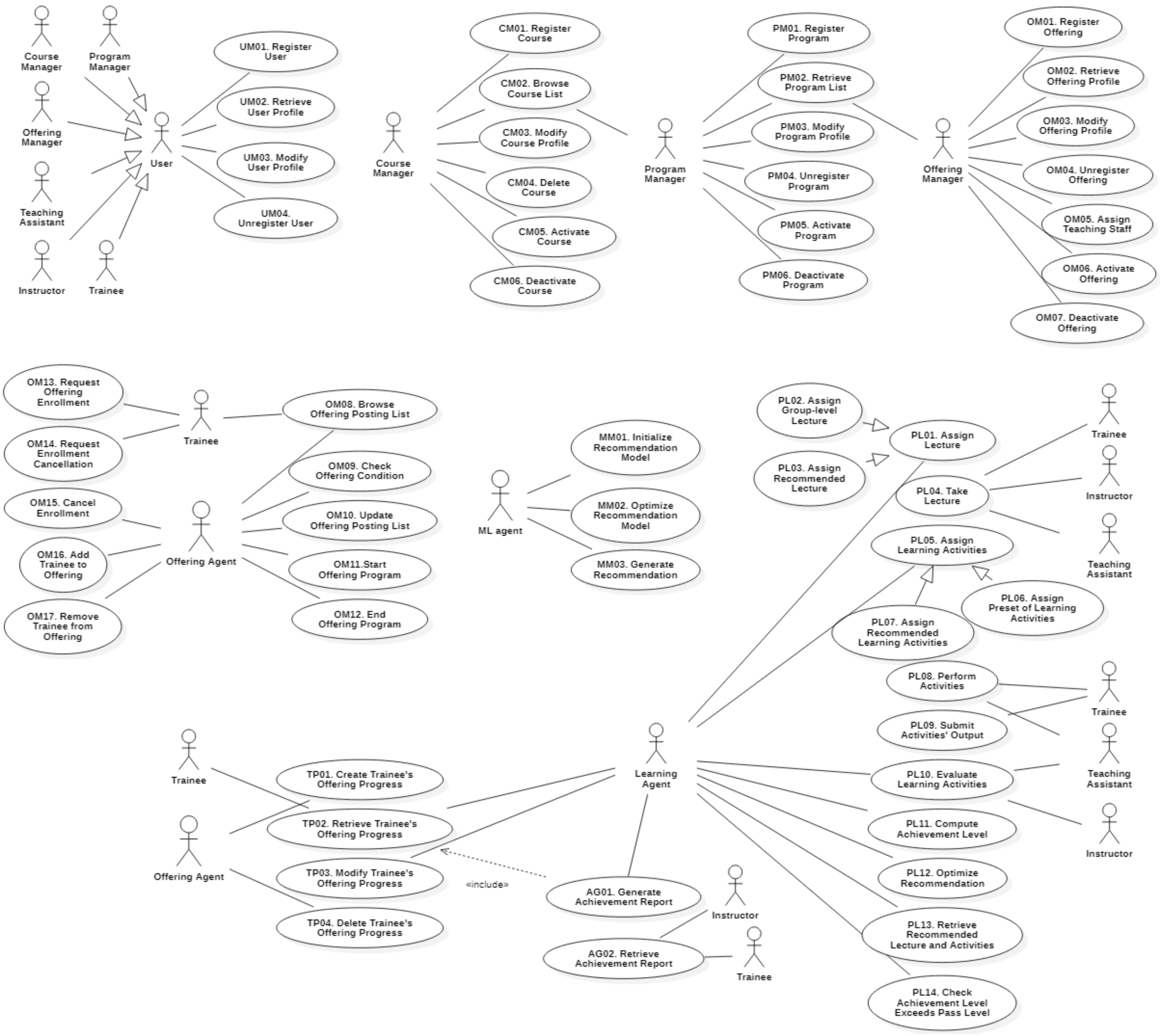
❑ Functional Groups

We use the same set of functional groups defined earlier.

❍ User Profile Management -> UM
❍ Program Profile Management -> PM
❍ Course Profile Management -> CM
❍ Offering Management -> OM
❍ Trainee Progress Management -> TP
❍ Personal Learning Management -> PL
❍ Machine Learning Model Optimization -> MM
❍ Achievement Report Generation -> AG

❑ Refined Use Case Diagram (Whole)

The following diagram shows the whole use case diagram. The refined use case diagram includes a number of enhancements and refinements over the context-level use case diagram.


Course Manager
Program Manager
Offering Manager
Teaching Assistant
Instructor
Trainee
User
UM01. Register User
UM02. Retrieve User Profile
UM03. Modify User Profile
UM04. Unregister User
Course Manager
CM01. Register Course
CM02. Browse Course List
CM03. Modify Course Profile
CM04. Delete Course
CM05. Activate Course
CM06. Deactivate Course
Program Manager
PM01. Register Program
PM02. Retrieve Program List
PM03. Modify Program Profile
PM04. Unregister Program
PM05. Activate Program
PM06. Deactivate Program
Offering Manager
OM01. Register Offering
OM02. Retrieve Offering Profile
OM03. Modify Offering Profile
OM04. Unregister Offering
OM05. Assign Teaching Staff
OM06. Activate Offering
OM07. Deactivate Offering
OM13. Request Offering Enrollment
OM14. Request Enrollment Cancellation
OM15. Cancel Enrollment
OM16. Add Trainee to Offering
OM17. Remove Trainee from Offering
Trainee
Offering Agent
OM08. Browse Offering Posting List
OM09. Check Offering Condition
OM10. Update Offering Posting List
OM11.Start Offering Program
OM12. End Offering Program
ML agent
MM01. Initialize Recommendation Model
MM02. Optimize Recommendation Model
MM03. Generate Recommendation
PL02. Assign Group-level Lecture
PL03. Assign Recommended Lecture
PL01. Assign Lecture
PL04. Take Lecture
PL05. Assign Learning Activities
PL06. Assign Preset of Learning Activities
PL07. Assign Recommended Learning Activities
PL08. Perform Activities
PL09. Submit Activities' Output
PL10. Evaluate Learning Activities
PL11. Compute Achievement Level
PL12. Optimize Recommendation
PL13. Retrieve Recommended Lecture and Activities
PL14. Check Achievement Level Exceeds Pass Level
Trainee
Instructor
Teaching Assistant
Trainee
Teaching Assistant
Instructor
Learning Agent
Trainee
Offering Agent
TP01. Create Trainee's Offering Progress
TP02. Retrieve Trainee's Offering Progress
TP03. Modify Trainee's Offering Progress
TP04. Delete Trainee's Offering Progress
«include»
AG01. Generate Achievement Report
AG02. Retrieve Achievement Report
Instructor
Trainee

5.1.3. [Step 3]. Derive Functional Components

There are three categories of functional components to consider; SRS-derived components, skeleton architecture-derived components, and interface-centric functional components.

- ❑ Category 1. Functional Components derived from the SRS

  The functional components are mainly derived from the system-intrinsic functionality, which is well modeled in its use case diagram. That is, the functional components can be systematically derived by clustering relevant use cases.

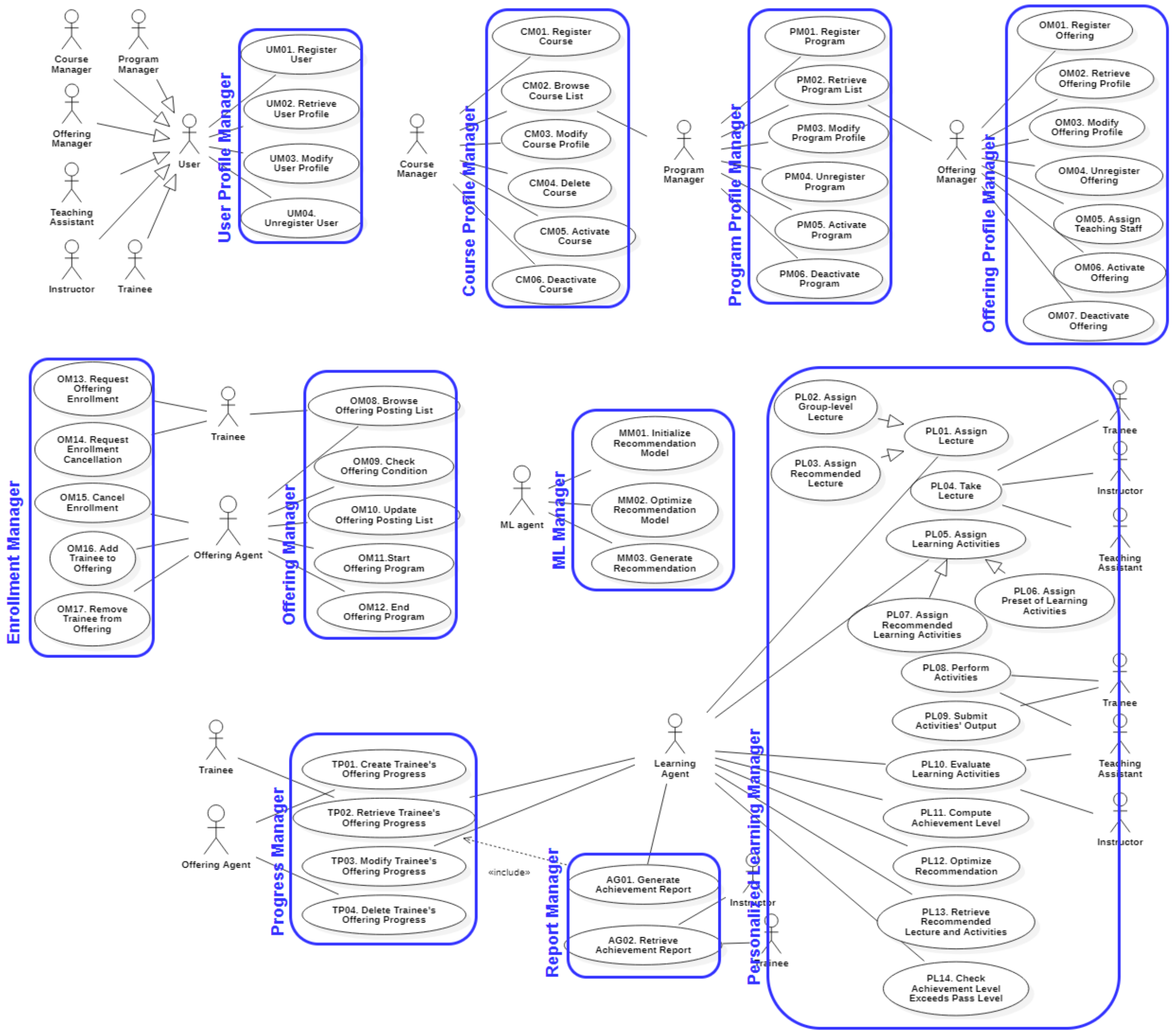
- ❑ Category 2. Functional Components derived from Skeleton Architecture

  A skeleton architecture is typically designed by applying architecture styles. An architecture style consists of components and connectors. Some of the components and connects may need to be modeled as functional components. That is, functional components could be derived from architecture styles in the skeleton architecture. Many of these functional components are pre-defined with a specific role in the architecture.

- Category 3. Interface-centric Functional Components

  An interface-centric functional component specifies a stable and public interface, which will be realized/implemented. For example, *Hardware Abstraction Layer (HAL)* is often adopted to designing architecture of embedded systems which interact with hardware devices such as sensors and actuators. Here, the *HAL* layer only specifies the interfaces of hardware devices in the system, and the layer itself does not include any implementation. Rather, its lower layer, *Hardware Adapter Layer*, implements the HAL for the given specific hardware devices.

- Functional Components derived from SRS


Course Manager
Program Manager
Offering Manager
Teaching Assistant
Instructor
Trainee
User
User Profile Manager
UM01. Register User
UM02. Retrieve User Profile
UM03. Modify User Profile
UM04. Unregister User
Course Manager
Course Profile Manager
CM01. Register Course
CM02. Browse Course List
CM03. Modify Course Profile
CM04. Delete Course
CM05. Activate Course
CM06. Deactivate Course
Program Manager
Program Profile Manager
PM01. Register Program
PM02. Retrieve Program List
PM03. Modify Program Profile
PM04. Unregister Program
PM05. Activate Program
PM06. Deactivate Program
Offering Manager
Offering Profile Manager
OM01. Register Offering
OM02. Retrieve Offering Profile
OM03. Modify Offering Profile
OM04. Unregister Offering
OM05. Assign Teaching Staff
OM06. Activate Offering
OM07. Deactivate Offering
Enrollment Manager
OM13. Request Offering Enrollment
OM14. Request Enrollment Cancellation
OM15. Cancel Enrollment
OM16. Add Trainee to Offering
OM17. Remove Trainee from Offering
Trainee
Offering Agent
Offering Manager
OM08. Browse Offering Posting List
OM09. Check Offering Condition
OM10. Update Offering Posting List
OM11.Start Offering Program
OM12. End Offering Program
ML agent
ML Manager
MM01. Initialize Recommendation Model
MM02. Optimize Recommendation Model
MM03. Generate Recommendation
Personalized Learning Manager
PL02. Assign Group-level Lecture
PL03. Assign Recommended Lecture
PL01. Assign Lecture
PL04. Take Lecture
PL05. Assign Learning Activities
PL06. Assign Preset of Learning Activities
PL07. Assign Recommended Learning Activities
PL08. Perform Activities
PL09. Submit Activities' Output
PL10. Evaluate Learning Activities
PL11. Compute Achievement Level
PL12. Optimize Recommendation
PL13. Retrieve Recommended Lecture and Activities
PL14. Check Achievement Level Exceeds Pass Level
Trainee
Instructor
Teaching Assistant
Trainee
Teaching Assistant
Instructor
Learning Agent
Instructor
Trainee
Trainee
Offering Agent
Progress Manager
TP01. Create Trainee's Offering Progress
TP02. Retrieve Trainee's Offering Progress
TP03. Modify Trainee's Offering Progress
TP04. Delete Trainee's Offering Progress
«include»
Report Manager
AG01. Generate Achievement Report
AG02. Retrieve Achievement Report

- Table of Functional Components derived from SRS

  This table shows the functional components of the system and their relevant use cases. It basically is the tabular representation of the use case clustering.

Functional Components	Use Cases

User Profile Manager	UM01. Register User UM02. Retrieve User Profile UM03. Modify User Profile UM04. Unregister User
Course Profile Manager	CM01. Register Course CM02. Browse Course List CM03. Modify Course Profile CM04. Delete Course CM05. Activate Course CM06. Deactivate Course
Program Profile Manager	PM01. Register Program PM02. Retrieve Program List PM03. Modify Program Profile PM04. Unregister Program PM05. Activate Program PM06. Deactivate Program
Offering Profile Manager	OM01. Register Offering OM02. Retrieve Offering Profile OM03. Modify Offering Profile OM04. Unregister Offering OM05. Assign Teaching Staff OM06. Activate Offering OM07. Deactivate Offering
Enrollment Manager	OM13. Request Offering Enrollment OM14. Request Enrollment Cancellation OM15. Cancel Enrollment OM16. Add Trainee to Offering OM17. Remove Trainee from Offering
Offering Manager	OM08. Browse Offering Posting List OM09. Check Offering Condition OM10. Update Offering Posting List OM11. Start Offering Program OM12. End Offering Program
ML Manager	MM01. Initialize Recommendation Model MM02. Optimize Recommendation Model MM03. Generate Recommendation
Trainee Progress Manager	TP01. Create Trainee's Offering Progress TP02. Retrieve Trainee's Offering Progress TP03. Modify Trainee's Offering Progress TP04. Delete Trainee's Offering Progress
Report Manager	AG01. Generate Achievement Report

	AG02. Retrieve Achievement Report
Learning Manager	PL01. Assign Lecture PL02. Assign Group-level Lecture PL02. Conduct Lecture PL03. Assign Recommended Lecture PL04. Conduct Learning Activities PL04. Take Lecture PL05. Assign Learning Activities PL06. Assign Preset of Learning Activities PL06. Request ML Model Personalization PL07. Activate Personalized Recommendation PL07. Assign Recommended Learning Activities PL08. Deactivate Personalized Recommendation PL08. Perform Activities PL09. Generate Achievement Report PL09. Submit Activities' Output PL10. Evaluate Learning Activities PL11. Compute Achievement Level PL11. Retrieve Trainee's Offering Progress PL12. Optimize Recommendation PL13. Retrieve Recommended Lecture and Activities PL14. Check Achievement Level Exceeds Pass Level

❍ Functional Components derived from Skeleton Architecture

Skeleton Architecture includes Dispatcher for QoS. Dispatcher consist of QoS Evaluator and Server Allocator.

- QoS Evaluator evaluates each PLA Server's QoS
- Server Allocator allocates optimal QoS PLA Server to PLA Client.

5.1.4. [Step 4] Refine Functional Components for Tiers

The skeleton architecture of the target system has the following tiers, and hence we refine functional components for the tiers.

Functional Components	PLA Client	PLA Server
User Profile Manager	cUser Profile Manager	sUser Profile Manager
Program Profile Manager		Program Profile Manager
Course Profile Manager		Course Profile Manager
Offering Profile Manager		Offering Profile Manager
Offering Manager	cOffering Manager	sOffering Manager
Enrollment Manager	cEnrollment Manager	sEnrollment Manager
Trainee Progress Manager	cTrainee Progress Manager	sTrainee Progress Manager

Learning Manager	cLearning Manager	sLearning Manager
ML Manager		ML Manager
Report Manager	cReport Manager	sReport Manager

❑ Description of Functional Components (for Client and Server Tiers)

Each functional component is described for its key functionality. In CEP, we provide a brief textual description for each component.

Some components are allocated to both tiers, and their functionalities would partially be different.

❍ sOffering Manager

This component provides the functionality of posting offering and starting offering's program on time.

❍ sTrainee Progress Manager

This component provides the functionality of managing trainee's progress on lectures and activities.

❍ …

❑ Interface-centric Functional Components

Not applicable to PLA system.

❑ Description for Components (for Dispatcher)

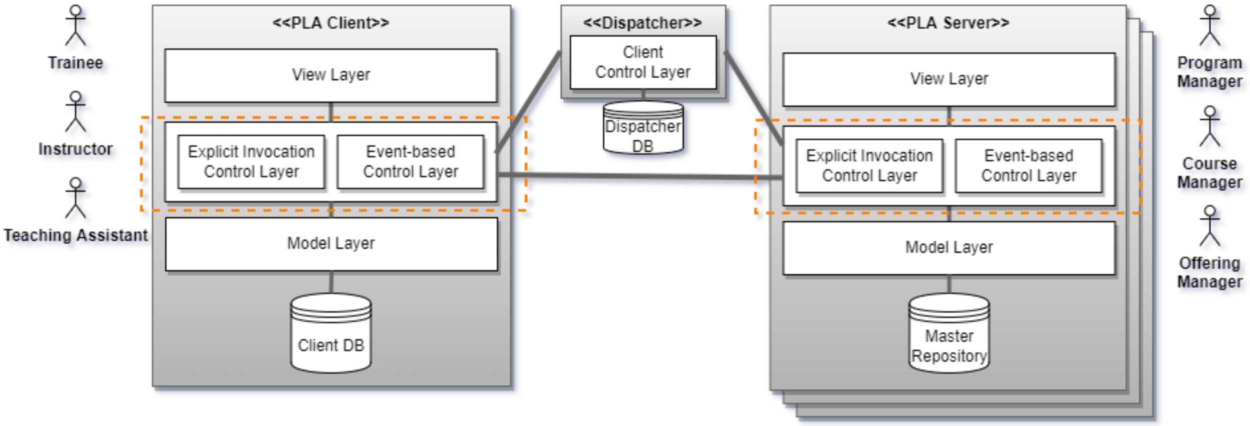
The components for Dispatcher are not mainly described in PLA System.

Not shown in the table.

5.1.5. [Step 5] Allocate Functional Components

Allocate functional components onto 'functionality place holders' of the skeleton architecture. A functional place holder is a layer, a partition, or any place which is defined to host some functionality.

❑ Place Holders for Functional Components

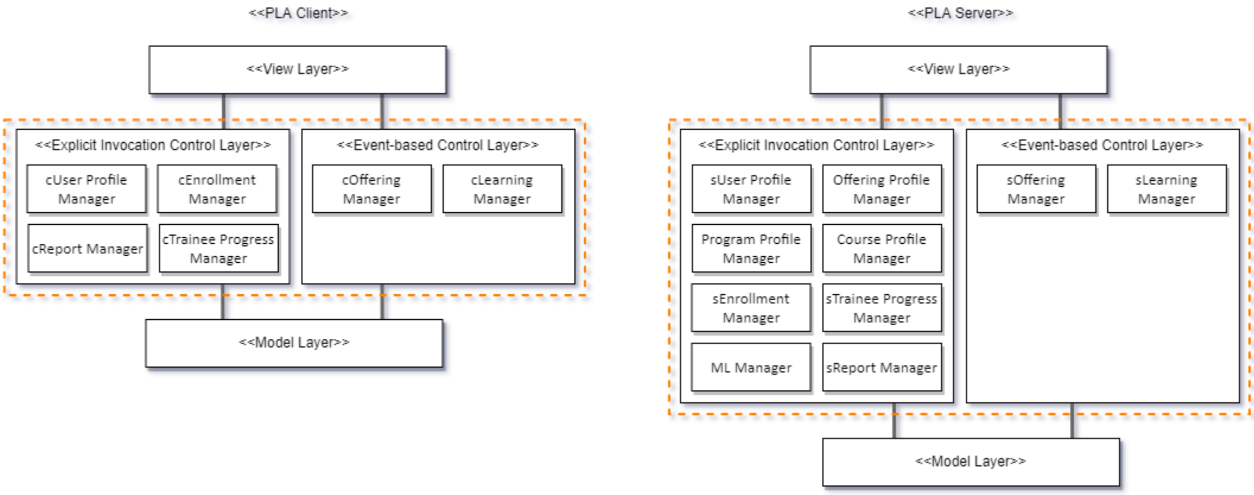


The control layer of each tier becomes the functionality place holders, as shown in the figure.

We now allocate the refined functional components on the functionality holders according to the table of refined functional components.

❑ Functional Components allocated

The functional components are allocated on the functionality place holders, as in the following figure.



Each place holder is assigned with several functional components.

The functional components for Dispatcher are mainly for finding optimal server for connection, they are not shown in here.

The skeleton architecture is now more complete with functional components.

5.1.6. [Step 6] Design Functional Components

Each functional component must be designed in greater details. There are different options for designing functional components in details. We use the following criteria for determining the type of each functional component.

❑ Criterion 1. Visibility of Functional Component

Whitebox Component or Blackbox Component

❑ Criterion 2. Type of Interface (for Blackbox Components)

Façade-type or Mediator-type

❑ Criterion 3. Variability of Functional Component

Closed Component or Open Component

❑ Criterion 4. Variation Points (for Open Component)

Variation point where the variability occurs.

We use the table of Functional Component Design Decisions as the following.

Refined Functional Components	Visibility	Interface Type	Open/Closed	Variant Points
sUser Profile Manager	Blackbox	Façade	Closed	N/A
Program Profile Manager	Blackbox	Façade	Closed	N/A
Course Profile Manager	Blackbox	Façade	Closed	N/A
Offering Profile Manager	Blackbox	Façade	Closed	N/A
sOffering Manager	Blackbox	Façade	Closed	N/A
sEnrollment Manager	Blackbox	Façade	Closed	N/A
sTrainee Progress Manager	Blackbox	Façade	Closed	N/A
sLearning Manager	Blackbox	Mediator	Closed	N/A
ML Manager	Blackbox	Façade	Open	Algorithm
sReport Manager	Blackbox	Façade	Closed	N/A
cUser Profile Manager	Blackbox	Façade	Closed	N/A
cOffering Manager	Blackbox	Façade	Closed	N/A
cEnrollment Manager	Blackbox	Façade	Closed	N/A
cTrainee Progress Manager	Blackbox	Façade	Closed	N/A
cLearning Manager	Blackbox	Façade	Closed	N/A
cReport Manager	Blackbox	Façade	Closed	N/A

- ❑ Whitebox Components

  There is no Whitebox component in PLA system.

- ❑ Designing Component with Openness

  There are some functional components that are specified with *Open* design such as ML Manager. This is to allow customizing and extending the variable parts of each component.

5.1.7. [Step 7] Define Interfaces of Functional Components

A functional component provides its functionality through a *provided* interface. A functional component with 'open' design may also need a *required* interface if it accepts a pluggable object as a variant. Note that the 'Required Interface' is only one of various ways of designing components with openness.

We first define the names of *provided* Interfaces for functional components. We use a prefix of 'i' to indicate an interface name.

Functional Components	PLA Client	PLA Server
User Profile Manager	icUser Profile Manager	isUser Profile Manager
Program Profile Manager		iProgram Profile Manager
Course Profile Manager		iCourse Profile Manager

Offering Profile Manager		iOffering Profile Manager
Offering Manager	icOffering Manager	isOffering Manager
Enrollment Manager	icEnrollment Manager	isEnrollment Manager
Trainee Progress Manager	icTrainee Progress Manager	isTrainee Progress Manager
Learning Manager	icLearning Manager	isLearning Manager
ML Manager		iML Manager
Report Manager	icReport Manager	isReport Manager

❑ Interface of Components

In this CEP, we define the interface of one component, *icEnrollment Manager*. The use cases included in the functional component are here;

❍ OM13. Request Offering Enrollment

❍ OM14. Request Enrollment Cancellation

❑ Provided Interface of *icEnrollment Manager*

Referring the use cases in the component, we define the following method signatures;

❍ getOfferingPostingList(void): offering_posting_list;
This method is to get offering list in posting for trainee to enroll.

❍ requestOfferingEnrollment(offering, trainee): Boolean;
This method is to request enrollment to designated offering. It returns true on success to request.

❍ getEnrolledOfferingList(trainee): enrolled_offering_list;
This method is to get currently enrolled offering list.

❍ requestEnrollmentCancellation(offering, trainee): Boolean;
This method is to request the cancellation of enrollment. It returns true on success to request in most case except for system failure.

5.2. Information View

Architecture design for Information View is to make decisions about persistent datasets, properties and their management. This activity includes a number of tasks including identifying data components, allocating data components, defining their data contents, ownership, data distribution, replication, migration, data security, and data timeliness.

5.2.1. [Step 1] Observe Informational Characteristics

We first observe the informational characteristics of the target system.

❑ Profile-related Classes

There are several classes for capturing various profiles in PLA system.

❑ Volume of Dataset

The volume of the dataset can be excessively high as the essential images are stored and achieved. The archived images are utilized in generating and enhancing machine learning models.

- Meta-information of Machine Learning Models

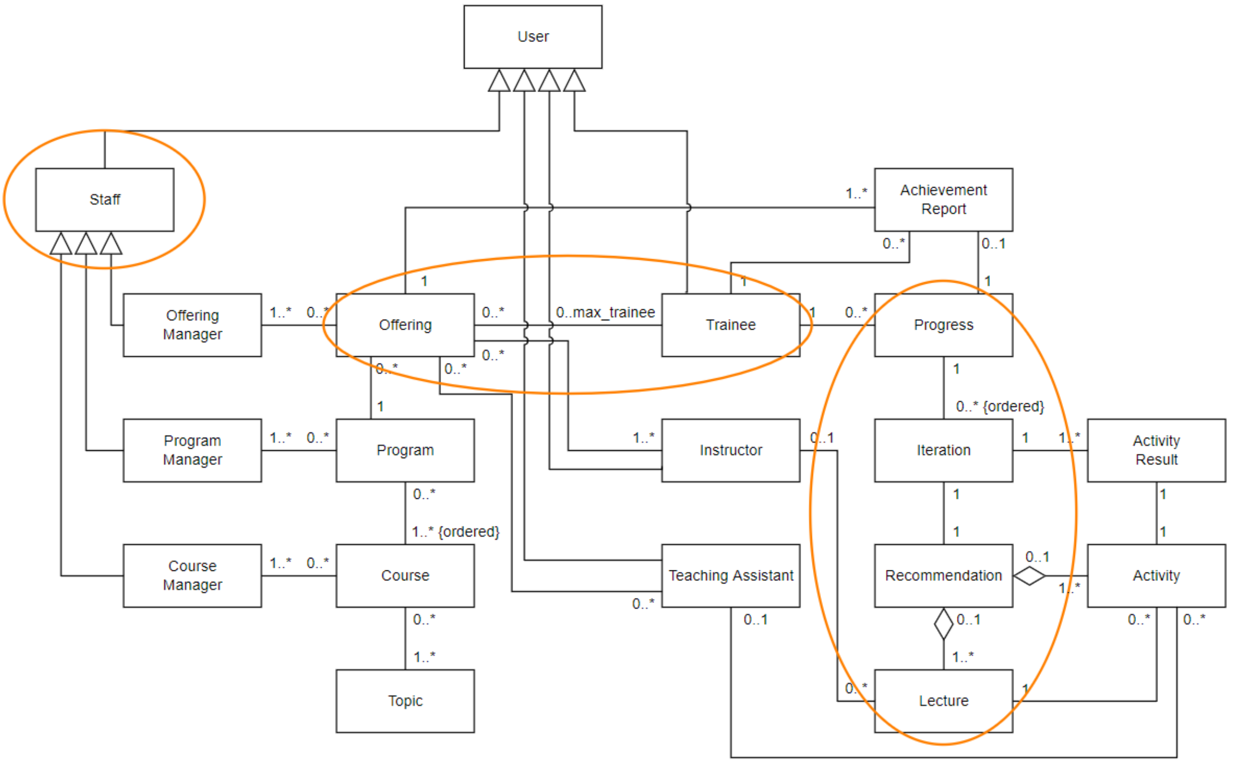
  Since PLA System provides analytics using a machine learning models, the meta-information of the models should be captured in some persistent classes.

5.2.2. [Step 2] Refine Persistent Object Model

In object-oriented paradigm, persistent datasets are modeled as entity-type classes. Hence, we refine the context-level class diagram with greater details, which becomes the basis for deriving data components.

- Refined Class Diagram

  The following diagram shows additional classes and refined relationships among classes.



- Refinements made on the Class Diagram

  The following refinements are made;

  - Adding *Staff* class as superclass of *Offering Manager, Program Manager* and *Course Manager*
  - Adding *Iteration* between *Progress* and *Recommendation* and define relationships
  - Defining Relationships *Offering* and *Trainee*.

- Adding key Attributes and Methods to Classes

  Each class is refined with key attributes and methods.

❍ Omitted in Sample Solution

Each class is specified with a textual description.

❑ User Profile-related Classes

User is the superclass which captures the common property of its subclasses.

❍ *Each subclass* captures the information of the its own role of a PLA system.

❑ Progress

Progress captures the information of trainee registered in an offering. *Progress* contains whole iterations of conduction offering, but also evaluated result.

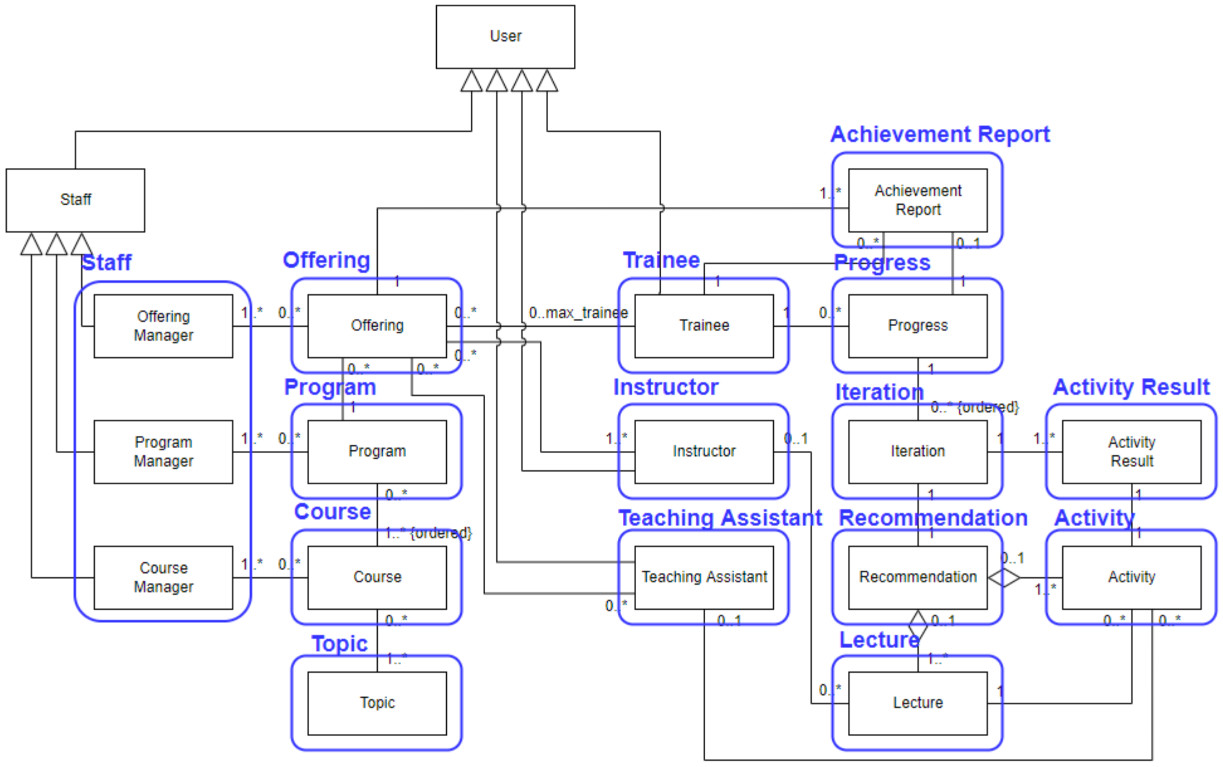
❑ Iteration

Iteration captures the information of conducting offering. *Iteration is consisted of* recommended lectures and activities, evaluated result, achievement level, and time elapsed of the iteration itself.

❑ …

5.2.3. [Step 3] Derive Data Components

By considering the strengths of inter-class relationships, we group a set of related classes into a data component.



As shown in the figure, classes with strong dependency such as inheritance and compositions are grouped into a same data component.

- ❑ Data Components derived from Class Diagram/SRS

  There are several data components that are derived by considering the strengths of relationships among classes.

- ❑ Data Components derived from Skeleton Architecture

  None

5.2.4. [Step 4] Refine Data Components for Tiers

The skeleton architecture of the target system has multiple tiers, and hence we need to refine them for the tiers.

Data Components	PLA Client	PLA Server
Staff		Staff
Trainee	C.Trainee	S.Trainee
Instructor	C.Instructor	S.Instructor
Teaching Assistant	C.TeachingAssistant	S.TeachingAssistant
Offering	C.Offering	S.Offering
Program		Program
Course		Course
Topic		Topic
Lecture	C.Lecture	S.Lecture
Progress	C.Progress	S.Progress
Iteration		Iteration
Recommendation		Recommendation
Achievement Report	C.AchievementReport	S.AchievementReport
Activity	C.Activity	S.Activity
Activity Result	C.ActivityResult	S.ActivityResult

- ❑ Data Components on *PLA Server*

  Most of the data components are allocated to *PLA Server*.

- ❑ Data Components allocated on multiple tiers

  When a data component is allocated on both tiers, the contents of the data components on the tiers are not same.

  - ❍ C.lecture, S.lecture
  - ❍ C.Progress, S.Progress

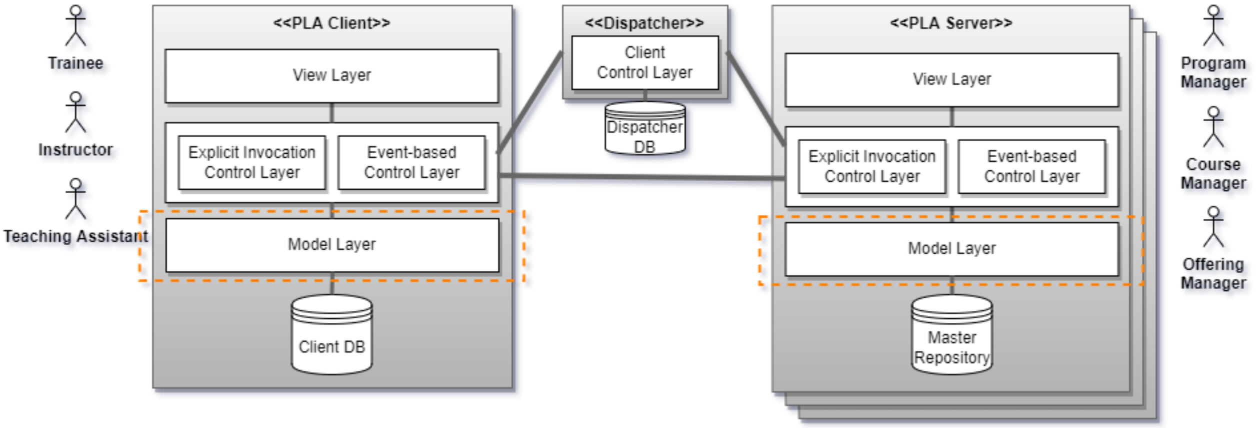
- ❑ Considering the Consistency with Allocation of Functional Components

  The allocation of data components is well aligned with the allocation of functional components. Hence, the inter-tier access between functional components and data components is not presented.

5.2.5. [Step 5] Allocate Data Components

Allocate the data components to the appropriate place holders of the architecture

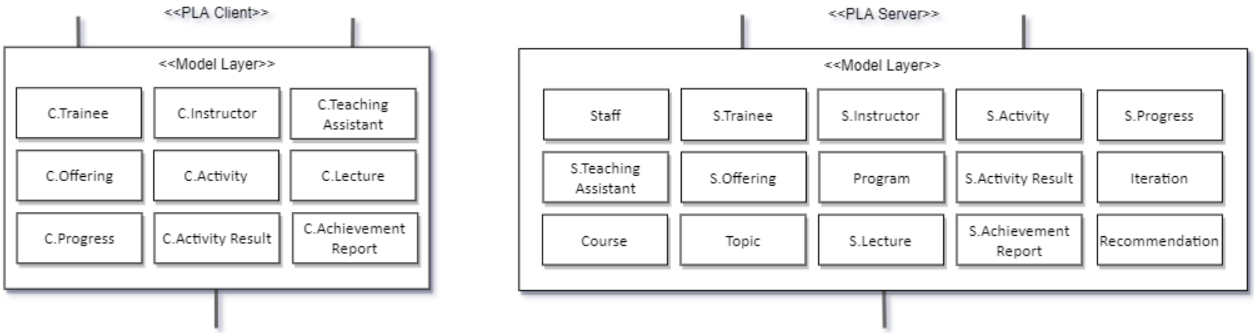
❑ Place Holders for Data Components



Each tier includes a model layer which hosts the data components.

❑ Data Components allocated

Based on the table of refined data components for tiers, we allocate the data components on the client tier.



The allocation of data components is made according to the table for ‘Data Components Refinement’. Each model layer is allocated with an appropriate set of data components.

5.2.6. [Step 6] Design Data Components

This step is to design the internal details of data components. This is trivial since each data component consists of classes and each class is defined with persistent attributes.

❑ Omitted in Sample Solution

5.2.7. [Step 7] Define Interfaces of Data Components

This step is to define the interface of each data component. The interfaces for data components are mostly for CRUD-type data manipulation.

❑ Omitted in Sample Solution

5.3. Behavioral View

The behavioral view of the architecture describes the dynamic aspect of the system, focusing on the runtime behavior of the system.

5.3.1. [Step 1] Observe Behavioral Characteristics

We observe the following behavior of the target system.

- ❑ Need for Explicit Invocation-based Control Flow
  Including profile-management functionality
- ❑ Event-driven Control Flow
- ❑ Parallel Processing to support Realtime Processing
- ❑ Closed-loop Control Flow for Treads
- ❑ Timer-based Invocation for Offering Management

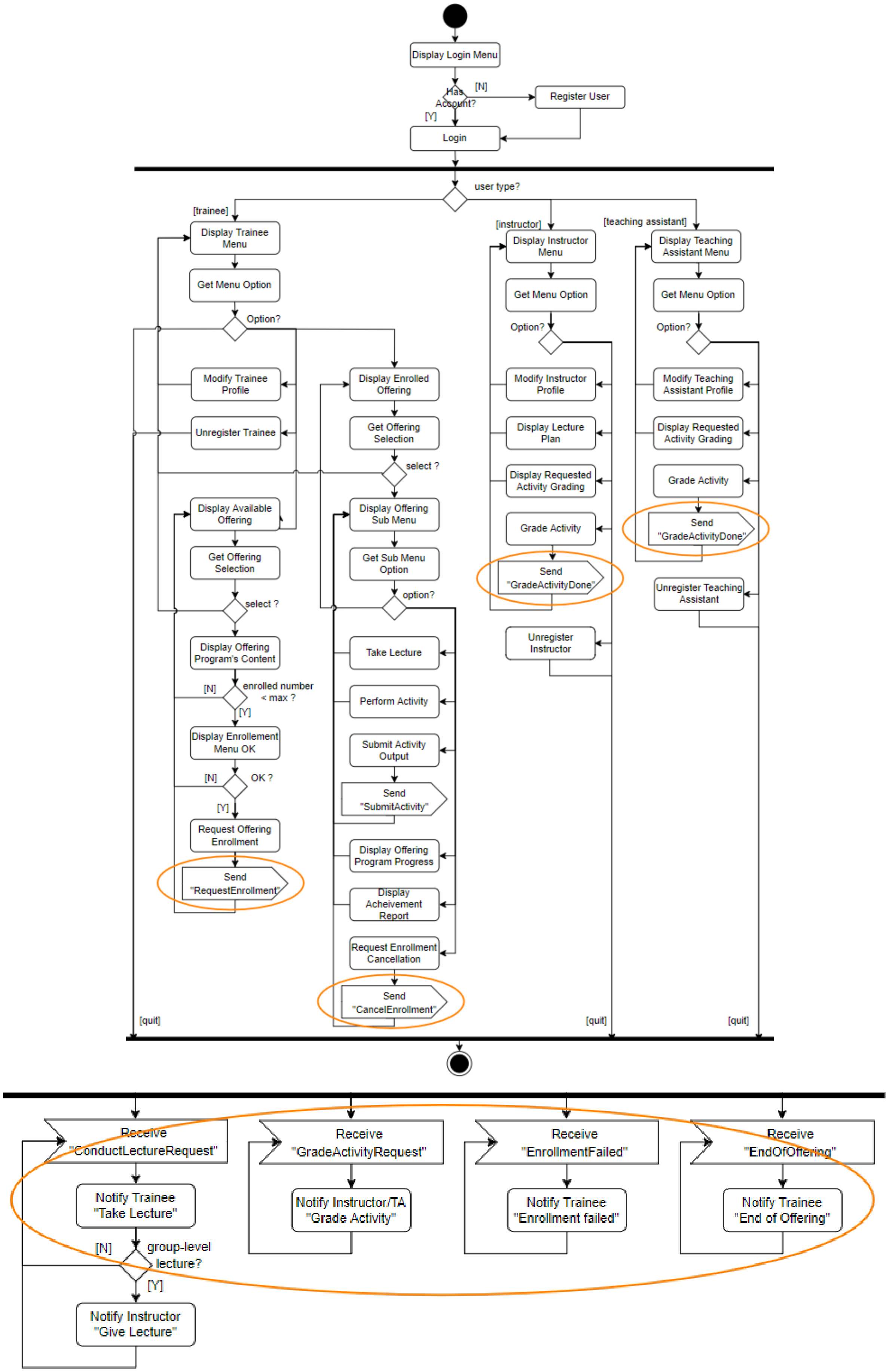
5.3.2. [Step 2] Refining Control Flow for whole System

We design the overall control flow of the target system. This is done by refining the context-level activity diagram. If the target system has multiple tiers, each tier has its own control flow and the interaction between the tiers should also be designed.

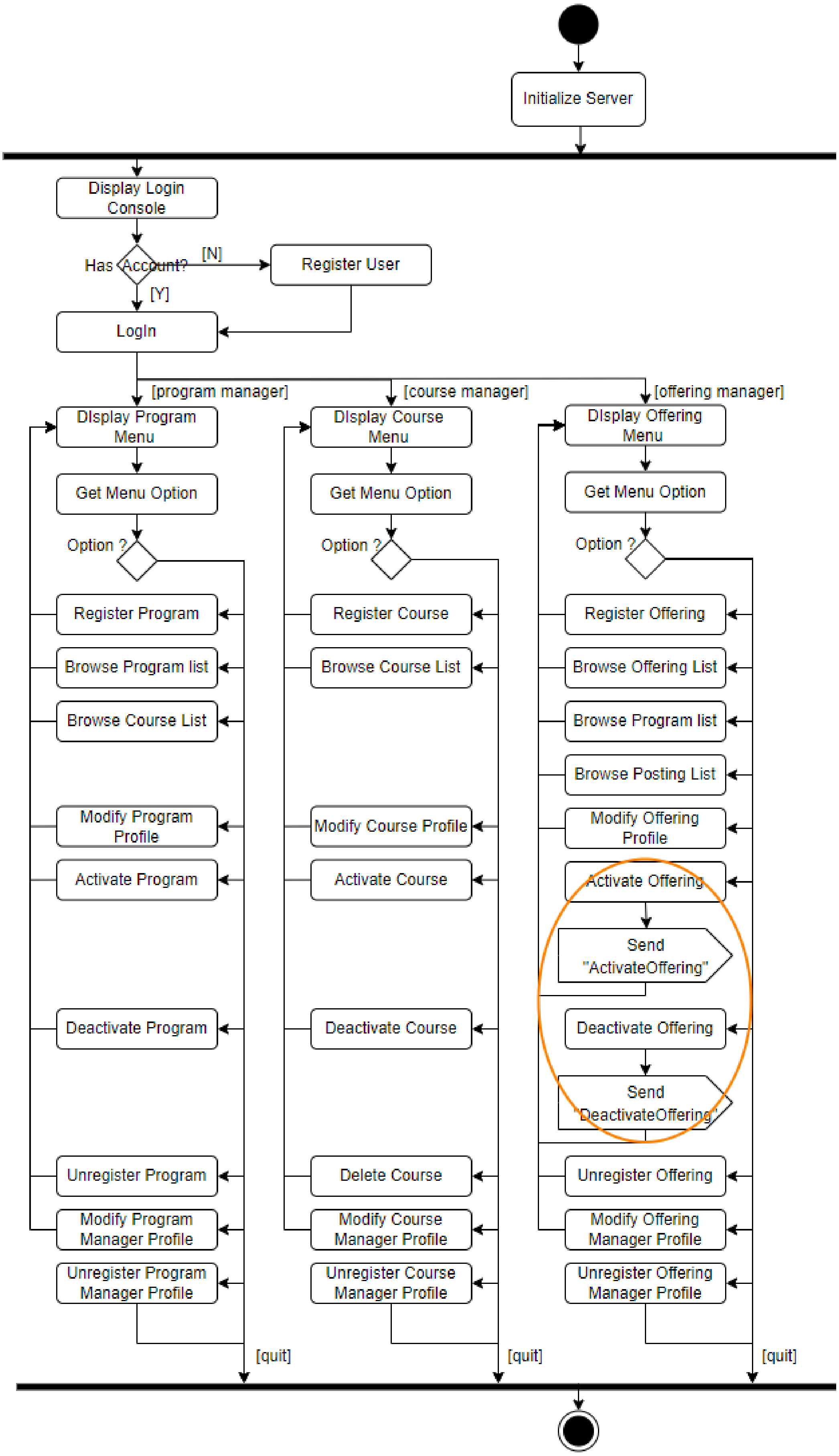
All the use cases in the functional view are reflected in the activity diagram, and all the actions and activities have their corresponding use cases. Hence, the consistency between the use case diagram and the activity diagram is well-maintained.

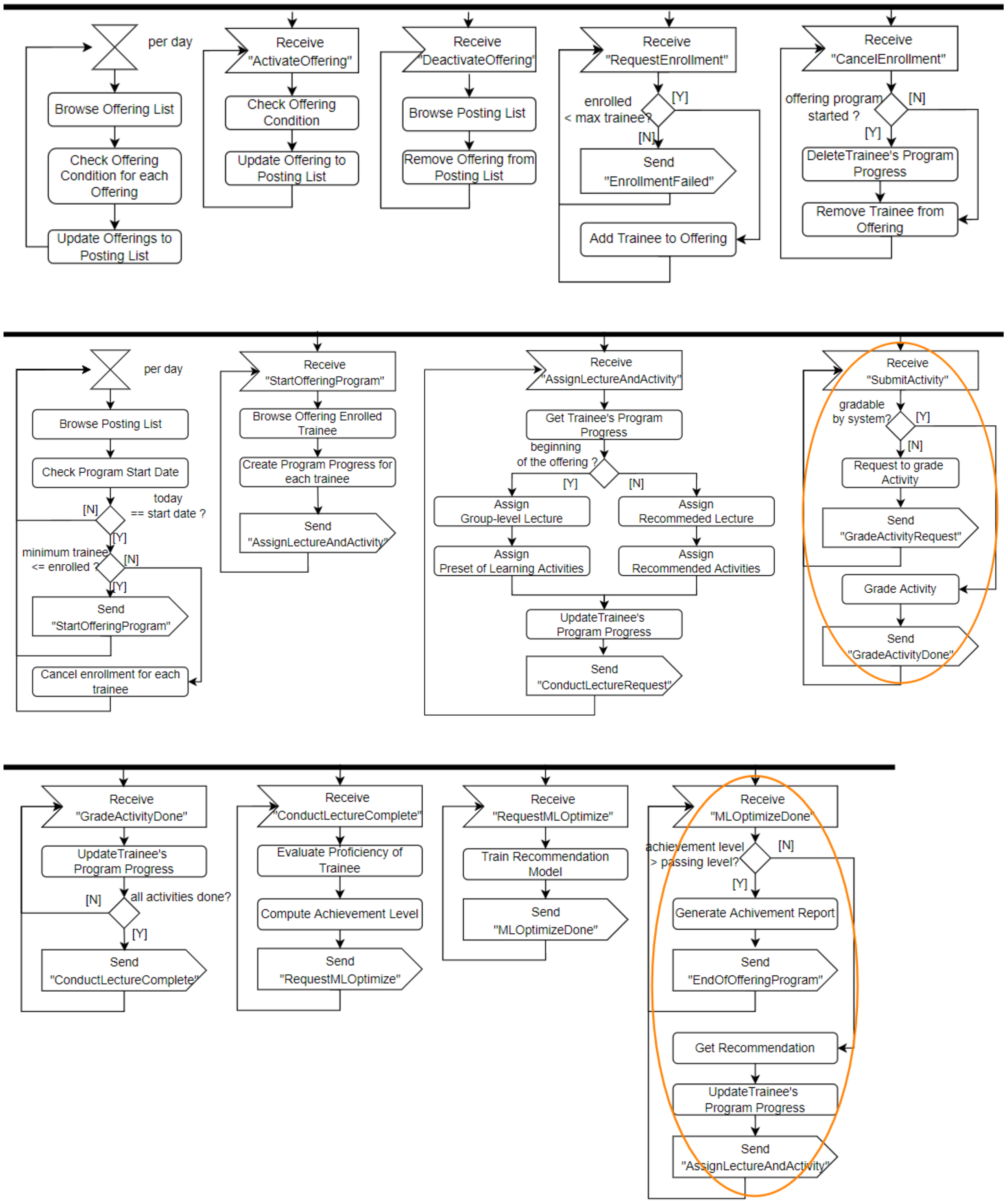
We specify the control flows of *PLA Client* and *PLA Server* in this CEP solution.

- Refined Control Flows of PLA Client


Display Login Menu
Has Account?
[N]
Register User
[Y]
Login
user type?
[trainee]
[instructor]
[teaching assistant]
Display Trainee Menu
Get Menu Option
Option?
Modify Trainee Profile
Unregister Trainee
Display Enrolled Offering
Get Offering Selection
select ?
Display Available Offering
Get Offering Selection
select ?
Display Offering Program's Content
enrolled number < max ?
[N]
[Y]
Display Enrollement Menu OK
OK ?
[N]
[Y]
Request Offering Enrollment
Send "RequestEnrollment"
Display Offering Sub Menu
Get Sub Menu Option
option?
Take Lecture
Perform Activity
Submit Activity Output
Send "SubmitActivity"
Display Offering Program Progress
Display Acheivement Report
Request Enrollment Cancellation
Send "CancelEnrollment"
Display Instructor Menu
Get Menu Option
Option?
Modify Instructor Profile
Display Lecture Plan
Display Requested Activity Grading
Grade Activity
Send "GradeActivityDone"
Unregister Instructor
Display Teaching Assistant Menu
Get Menu Option
Option?
Modify Teaching Assistant Profile
Display Requested Activity Grading
Grade Activity
Send "GradeActivityDone"
Unregister Teaching Assistant
[quit]
[quit]
[quit]
Receive "ConductLectureRequest"
Notify Trainee "Take Lecture"
[N]
group-level lecture?
[Y]
Notify Instructor "Give Lecture"
Receive "GradeActivityRequest"
Notify Instructor/TA "Grade Activity"
Receive "EnrollmentFailed"
Notify Trainee "Enrollment failed"
Receive "EndOfOffering"
Notify Trainee "End of Offering"

- Refined Control Flows of PLA Server

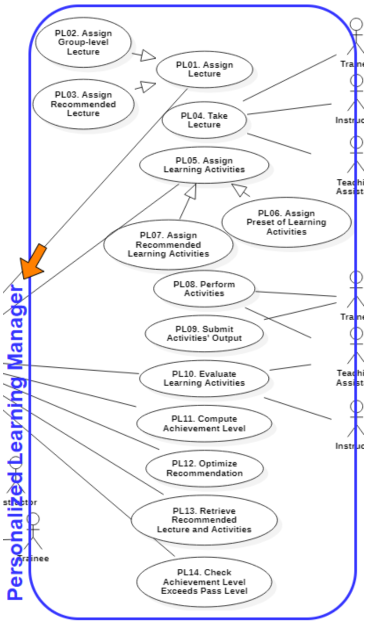



per day
Browse Offering List
Check Offering Condition for each Offering
Update Offerings to Posting List
Receive "ActivateOffering"
Check Offering Condition
Update Offering to Posting List
Receive "DeactivateOffering"
Browse Posting List
Remove Offering from Posting List
Receive "RequestEnrollment"
enrolled < max trainee?
[Y]
[N]
Send "EnrollmentFailed"
Add Trainee to Offering
Receive "CancelEnrollment"
offering program started ?
[N]
[Y]
DeleteTrainee's Program Progress
Remove Trainee from Offering
per day
Browse Posting List
Check Program Start Date
today == start date ?
[N]
[Y]
minimum trainee <= enrolled ?
[N]
[Y]
Send "StartOfferingProgram"
Cancel enrollment for each trainee
Receive "StartOfferingProgram"
Browse Offering Enrolled Trainee
Create Program Progress for each trainee
Send "AssignLectureAndActivity"
Receive "AssignLectureAndActivity"
Get Trainee's Program Progress
beginning of the offering ?
[Y]
[N]
Assign Group-level Lecture
Assign Recommeded Lecture
Assign Preset of Learning Activities
Assign Recommended Activities
UpdateTrainee's Program Progress
Send "ConductLectureRequest"
Receive "SubmitActivity"
gradable by system?
[Y]
[N]
Request to grade Activity
Send "GradeActivityRequest"
Grade Activity
Send "GradeActivityDone"
Receive "GradeActivityDone"
UpdateTrainee's Program Progress
all activities done?
[N]
[Y]
Send "ConductLectureComplete"
Receive "ConductLectureComplete"
Evaluate Proficiency of Trainee
Compute Achievement Level
Send "RequestMLOptimize"
Receive "RequestMLOptimize"
Train Recommendation Model
Send "MLOptimizeDone"
Receive "MLOptimizeDone"
achievement level > passing level?
[N]
[Y]
Generate Achivement Report
Send "EndOfOfferingProgram"
Get Recommendation
UpdateTrainee's Program Progress
Send "AssignLectureAndActivity"

5.3.3. [Step 3] Choosing Elements for Detailed Control Flow

From the overall control flow, we choose some parts of the overall control for designing the detailed control flow. For CEP, we choose only one element for modeling the detailed control flow.

❑ "Conducting Offering" use case of Personalized Learning Manager Component


PL02. Assign Group-level Lecture
PL01. Assign Lecture
PL03. Assign Recommended Lecture
PL04. Take Lecture
PL05. Assign Learning Activities
PL06. Assign Preset of Learning Activities
PL07. Assign Recommended Learning Activities
PL08. Perform Activities
PL09. Submit Activities' Output
PL10. Evaluate Learning Activities
PL11. Compute Achievement Level
PL12. Optimize Recommendation
PL13. Retrieve Recommended Lecture and Activities
PL14. Check Achievement Level Exceeds Pass Level
Trai
Instruc
Teachi Assist
Trai
Teachi Assist
Instruc
structor
Trainee
Personalized Learning Manager

5.3.4. [Step 4] Defining Detailed Control Flow

We should design the detailed control flows for all the selected elements.

❑ Functionality of "Taking Personalized Learning (Conducting Offering)"

This use case allows trainee has personalized lecture and activities to maximize learning efficiency. This use case starts with assigning one or more group-level lectures to trainee for beginning. There are multiple iterations of lectures and activities for finding optimized recommendation and maximizing learning efficiency. There are a number of factors to consider in designing its control flow.

❍ Course Pass Conditions

An Offering is mapped to only one Program. Hence the Offering follows the sequence of Course in the Program. To move on next course, trainee needs to meet pass conditions. The pass conditions are consisted of course point of given activities, minimum number of lecture conducted in the course, maximum iteration allowed and maximum duration allowed.

❍ Offering End Condition

An Offering has pass level. If and only if all the courses are taken then trainee's achievement level is greater than pass level, the offering is finished. An Offering also has Fail condition for training duration. When total training time elapsed exceeding Offering's training duration, the Offering is ended. An Achievement report for the Offering is generated by following the end of Offering.

- Design Scheme to Apply

  The functionality of this use case involves a complicated workflow with sequences, branches, and loops. Hence, we choose to represent the detailed design using pseudocode form of the *algorithm*.

- Representation Meta-Elements of Personalized Learning
  - Offerings
    - PLA system has a number of Offerings which contains only One Program.
    - $OFFR_1$, $OFFR_2$, …, $OFFR_{numOfferings}$
  - Courses
    - An Offering is a sequence of Courses.
    - CRS_1, CRS_2, …, $CRS_{numCourses}$; CRS in Offering is represented as CRS^{OFFR}_{number}.
    - CRS^{6}_{8} indicates that the Course #8 of Offering #6.
  - Topics
    - A Course is a set of Topics. Topic is consisted of tuple of (Lecture list, Exercise list, Lab list, Assignment list, Project list, Presentation list, Test list).
  - Trainees
    - An Offering has a number of enrolled trainees.
    - TRN_1, TRN_2, …, $TRN_{numTrainees}$; Trainee in Offering is represented as TRN^{OFFR}_{number}.
    - TRN^{3}_{5}. Indicates that the Trainee #5 of Offering #3.
  - Progress
    - A Progress represents each Trainee's progress on Offering. It defines all history of learning process including activities' result.
    - PGRS, Trainee's progress on Offering is represented as $PGRS^{OFFR, TRN}$.
    - $PGRS^{2, 6}$ Indicates that the Progress of Trainee #6 in Offering #2.
  - Iterations
    - A Progress includes a sequence of running Iterations. An Iteration is consisted of lecture, activities and the evaluation of activities.
    - $ITER^{3, 4, 2}_{7}$ indicates that the Iteration #7 of Course #2 for Trainee #4 in Offering #3
  - Recommendations
    - A Recommendation includes set of lecture list and set of activity list of Course. It is generated by ML model and gives a ways of personalized learning.
    - Recommendation in Course is represented as $RCMD^{TRN, CRS}$.
    - $RCMD^{3, 4}$ indicates that the recommendation of Course #4 for Trainee #3.
  - Achievement Reports
    - An achievement report is output of Offering. It is generated at the end of the Offering.
    - Achievement Report of Offering represented as $RPT^{OFFR, TRN}$.
    - $RPT^{5, 6}$ indicates that the achievement report of Trainee #6 for Offering #5.

❑ Defining Primitive Methods

- check_beginning(PRGS) → Boolean

  This method returns TRUE when there is no history of the trainee's progress. If recommended lecture and activities are given, it returns FALSE.

- get_duration(PRGS) → running time of Offering

  This method is to get elapsed time of trainee on an Offering.

- check_course_pass_condition(CRS, PRGS) → next action

  This method is to check that the trainee passes the course or not. To pass the course, trainee should get minimum number of lectures defined. Under this condition, when the course point is greater than pass level, this method returns 'Move on Next Course'. get elapsed time of trainee on an Offering. Progress has all history of

  When trainee gets smaller number of lectures than minimum lecture required.

  - When trainee gets smaller number of lectures than minimum lecture required, it returns : Need Lectures in Course, to get more iteration

  When trainee gets more lectures than minimum lectures required.

  - When the course point is greater than pass level, it returns : Move on next Course
  - When the iteration reaches max limit number, it returns : Move on next Course.
  - When the duration reaches max limit, it returns : Move on next Course.
  - Otherwise, it returns : Need Lectures in Course, to get more iteration.

- compute_achievement_level(PRGS)

  This This method is to calculate trainee's achievement level for given Offering. Progress has all Course's score. The achievement level can get by next formula.

  achievement level = $\frac{\sum_{q=0}^{\text{numCourses}} \text{course\_point}_q \times \text{course\_weight}_q}{\sum_{q=0}^{\text{numCourses}} \text{perfect\_course\_score}_q \times \text{course\_weight}_q}$

- optimize_recommendation(CRS, TRN, RCMD, efficiency)

  This method is to optimize recommendation model with the result of recommended action. This method can be designed by utilizing Reinforcement Learning. Considering this RL method, the following parts can be defined. In CEP, it is not handled in detail.

  - Agent : Trainee
  - Action : Take Recommended Lectures and Activities
  - State : Current Lectures and Activities
  - Environment : Course of Offering
  - Value Function : Sum of expected reward when following recommendation, the core of the algorithm is a Bellman equation.
  - Reward : activities' efficiency = $\frac{\sum_{q=0}^{\text{numActivities}} \text{evaluation\_result}_q \times \text{activity\_weight}_q}{\sum_{q=0}^{\text{numActivities}} \text{perfect\_score}_q \times \text{activity\_weight}_q}$
  - Goal : Optimal recommendation that Maximizes Reward

- Generate_achievement_report(PRGS) → RPT

  This method is to generate an achievement report of a trainee for an offering.

- Main Control Flow of ‘Conducting Offering’

  We define the main control flow for the use case using the primitive methods.

  - Input: $OFFR_i$
  - Output: $RPT^{i,k}$ of TRN_k for $OFFR_i$
  - Begin

```
Start of Conducting Offering
For each $TRN_k$ in $OFFR_i$ parallel do
    initialize_achievement_level($PRGS^{i,k}$)
    n ← 0            // iteration counter of a course
    p ← 0            // course counter
    $RCMD^{k,p}$ ← (empty set of lectures, empty set of activities)
    While (p < number of courses in $OFFR_i$)
        If (get_achievement_level($PRGS^{i,k}$) >= pass_level of $OFFR_i$)
            Break
        If (get_duration($PRGS^{i,k}$) < training_duration of $OFFR_i$)
            Break
        If (check_beginning($PRGS^{i,k}$) == TRUE || $RCMD^{k,p}$ is empty)
            $RCMD^{k,p}$ ← get_lectures_and_activities_for_beginning($CRS^{i}_{p}$),
        lectures, activities ← $RCMD^{k,p}$
        update_progress_lecture($ITER^{i,k,p}_{n}$ , lectures)
        take_lecture($TRN_k$, $ITER^{i,k,p}_{n}$)
        update_progress_activity($ITER^{i,k,p}_{n}$, activities)
        output_of_activities ← empty set
        evaluation_result ← empty set
        For each $activity_m$ in activities
            $output\_of\_activities_m$ ← perform_activity($TRN_k$, $activity_m$)
            $evaluation\_result_m$ ← evaluate_activity($activity_m$, $output\_of\_activities_m$)
        $efficiency_n$ ← $\frac{\sum_{q=0}^{numActivities} evaluation\_result_q \times activity\_weight_q}{\sum_{q=0}^{numActivities} perfect\_score_q \times activity\_weight_q}$
        $course\_point_p$ ← average of { $efficiency_0$ , … , $efficiency_n$ }
        update_progress_result($ITER^{i,k,p}_{n}$ , evaluation_result, $efficiency_n$, $course\_point_p$)
        compute_achievement_level($PRGS^{i,k}$)
        optimize_recommendation($CRS^{i}_{p}$, $TRN_k$, $RCMD^{k,p}$, $efficiency_n$)
        If (check_course_pass_condition($CRS^{i}_{p}$, $PRGS^{i,k}$) == Need Lectures in Course)
            n ← n + 1;
        Else
            n ← 0, p ← p + 1;          // Both Fail and Pass case, move on to next course
        $RCMD^{k,p}$ ← get_recommendation($CRS^{i}_{p}$, $TRN_k$)
    $RPT^{i,k}$ ← generate_achievement_report($PRGS^{i,k}$)
    update_progress_report($PRGS^{i,k}$, $RPT^{i,k}$ )
End of Conducting Offering
```

End

5.4. Deployment View

Deployment view of the architecture is concerned with the topology of software components on the physical layer, as well as the physical connections between these components.

5.4.1. [Step 1] Observe Deployment Characteristics

❑ The PLA Client tier will be realized with various device, such as mobile, desktop, laptop.

❑ The PLA Server tier will be realized with a number of large-sized servers.

The number of server instances can initially be determined by a forecasted degree of clients' requests. However, the number may need to be dynamically adjusted by the actual degree of clients' requests.

❑ The Dispatcher tier will be realized with a medium-sized server.

5.4.2. [Step 2] Define Nodes

The skeleton architecture of the target system consists of 3 tiers.

❑ Node 1. PLA Client

This node denotes a client application running on Windows, Linux, macOS, iOS, Android and web Browser also. Hence, the computation power and resources are already specified by the OS and device vendors.

❑ Node 2. PLA Server

The PLA server requires a large-scale of parallel processing and large storage capacity. All persistent data will be managed with MySQL DBMS. It also generates each trainee's personal learning recommendation. This node is provided with Spring Framework which has a strength of building larger scale applications that use a cloud approach.

❑ Node 3. Dispatcher

This node is deployed with a python-base Django Framework web server

5.4.3. [Step 3] Define Network Connectivity

❑ Between Server and Client: HTTP protocol-based network configuration.

❑ Between Server and Dispatcher: HTTP protocol-based network configuration.

❑ Between Client and Dispatcher: HTTP protocol-based network configuration.

5.4.4. [Step 4] Define Artifacts to Deploy

❑ Artifacts for PLA Client

❍ Functional Components specified in Functional View-design
Data Components specified in Information View-design

❍ Implementation of Behavior View-design

❑ Artifacts for PLA Server

  - ❍ Functional Components specified in Functional View-design
  - ❍ Data Components specified in Information View-design
  - ❍ Implementation of Behavior View-design

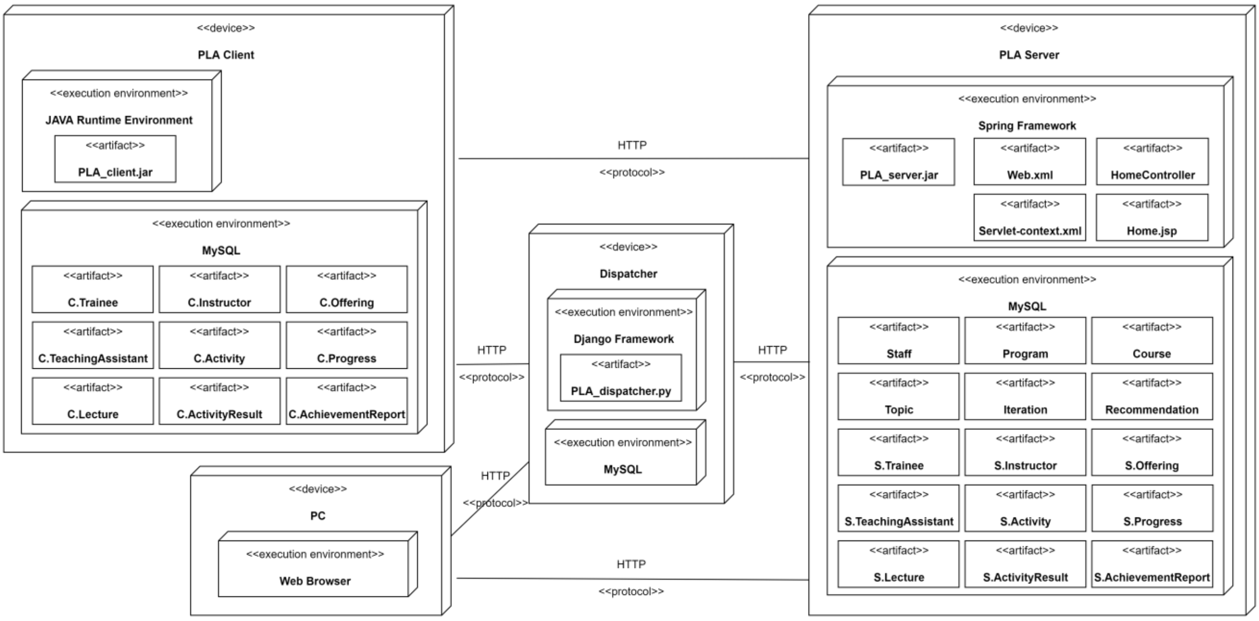
- ❑ Artifacts for Dispatcher
  - ❍ Functional Components described in Functional View-design

5.4.5. [Step 5] Allocate Artifacts on Nodes

Allocate all the deployable artifacts and show the network connections

- ❑ Deployment Diagram

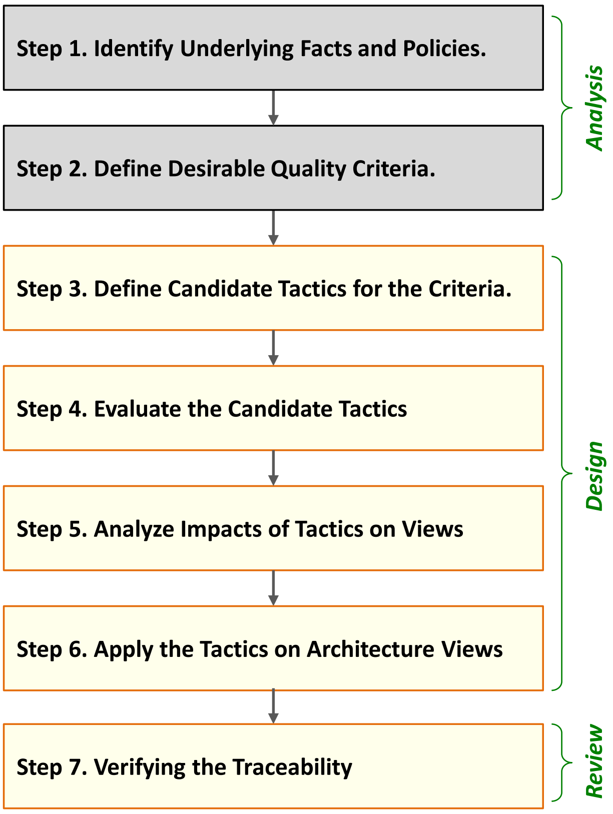
  Allocate components and sub-systems to the appropriate place holders of the diagram as shown in the following diagram.



We define network connection between the tiers.

6. Activity 5. NFR-specific Architecture Design

This chapter shows the architectural design for the given NFRs. The following process is used for applying NFR-based design.


Step 1. Identify Underlying Facts and Policies.
Step 2. Define Desirable Quality Criteria.
Step 3. Define Candidate Tactics for the Criteria.
Step 4. Evaluate the Candidate Tactics
Step 5. Analyze Impacts of Tactics on Views
Step 6. Apply the Tactics on Architecture Views
Step 7. Verifying the Traceability
Analysis
Design
Review

6.1. Design for NFR-1. High Effectiveness of Personalized Learning

The most unique and noble feature of the system is the ability to personalize the learning contents and activities for each trainee. Hence, the system should be designed to provide the high effectiveness of personalizing the learning contents and activities.

In contrast to conventional group-level lecture, the system should consider all the achievement-related measures of each trainee and reflect them on the personalization. Especially, the learning effectiveness of a given learning activity such as a laboratory should be measured for each trainee and reflected in further personalization.

6.1.1. [Step 1] Underlying Facts and Policies

We define the following facts and policies regarding the NFR.

- ❑ (F1) Evolving Recommendation Model

  As trainee continuing program, more effective contents and activities set would be recommended.

- ❑ (F2) Incorrectness from Cold-start

  In the beginning of Offering, not effective recommendation would be provided.

❑ (F3) Time on Performing

The time each trainee performs on the same activity can be a factor in efficiency.

❑ (F4) Different Time Pattern of Learning

Each trainee has different time pattern of learning.

❑ (F5) Long-term Memory Aid

Learning that leads to long-term memory is more effective.

❑ (F6) Variety of Contents

Providing content in a variety of ways enhances the effectiveness of learning.

6.1.2. [Step 2] Criteria for Satisfying NFR

❑ (C1) Select proper Recommendation Model (Relevant to F1)

The model for recommendation should make efficient set of contents and activities.

❑ (C2) More Personalized Initial Recommendation (Relevant to F2)

There should be a way of recommendation that reflecting personal preferences in the beginning.

❑ (C3) Time on Performing (Relevant to F3)

Effectiveness formulation should consider the time on performing activity.

❑ (C4) Pattern Monitoring (Relevant to F4)

Different learning pattern in time manner should be get from Trainee's activity. Learning suggestion and notification should be provided.

❑ (C5) Memorizing Method (Relevant to F5)

The system should be designed in a way that long-term memory is maintained.

❑ (C6) Multiple type of Contents (Relevant to F6)

The system should provide the various type of contents including audio, text, and video.

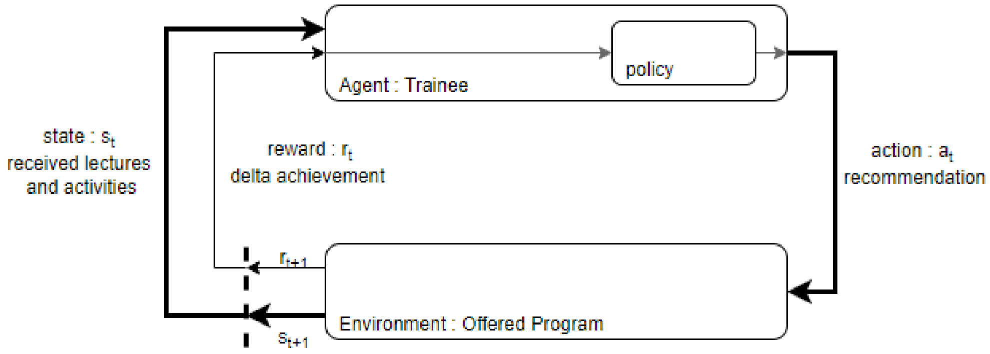
6.1.3. [Step 3] Candidate Tactics for the Criteria

The following tactics are proposed for the identified criteria.

❑ (T1) Select Reinforcement Learning for Recommendation Model (Relevant to C1)

Finding personalized recommendation is like real-world problem. For real-world problem, we can find the proper solution with try and error from repetition. There are several solutions with stochastic approach and ML approach. This system found RL (Reinforcement Learning) is optimal application which is biased to each trainee.

Components of RL are related with PLA system in below figure.



Reward from recommended action is calculated with next formulation

- Reward is represented improved amount of achievement level

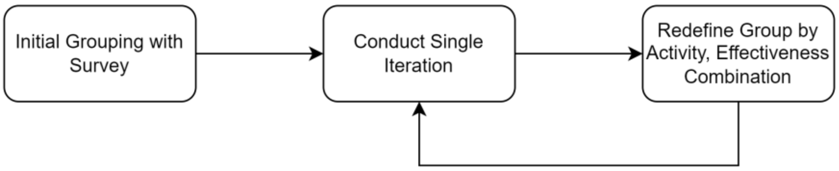
$$r_t = \; AchievementLevel_t - AchievementLevel_{t-1}$$

(Where, $AchievementLevel = \frac{\sum_{q=0}^{numCourses} \mathrm{course\_point_q} \times \mathrm{course\_weight_q}}{\sum_{q=0}^{numCourses} \mathrm{perfect\_course\_score_q} \times \mathrm{course\_weight_q}}$)

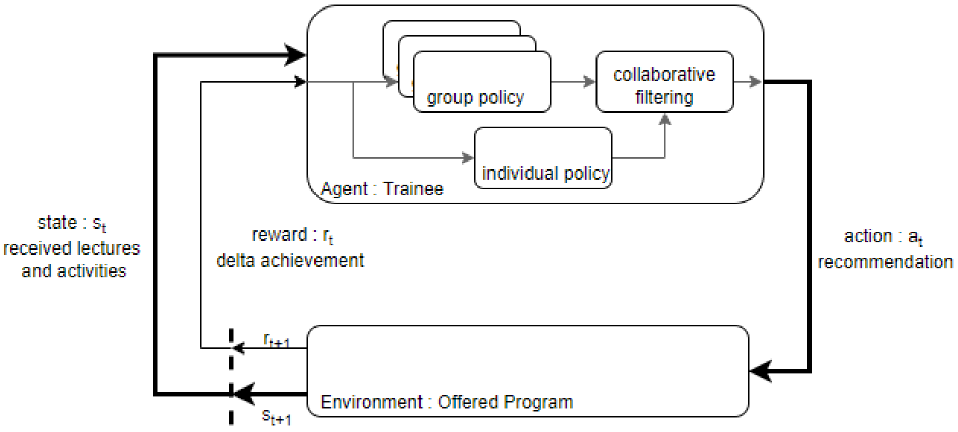
❑ (T2) Increase the accuracy of initial recommendations (Relevant to C1, C2)

To deal with cold-start problem, this tactic applies 2 kinds of strategies.

- Before starting program, Trainee will get initial survey about activity preference and self-rating of the subject understanding. With survey, initial grouping is done. The trainee's group is repeatedly adjusted at the end of each iteration.


Initial Grouping with Survey
Conduct Single Iteration
Redefine Group by Activity, Effectiveness Combination

- Another approach combined with is collaborative filtering. This algorithm corelates likely-effective recommendation from group policy and recommendation from individual policy.



At the end of each iteration, group policy which is trainee involved and individual policy updated. The weight of group policy is gradually decrease and finally individual policy is only selected for recommendation.

- (T3) Considering time consumed at each activity (Relevant to C3)

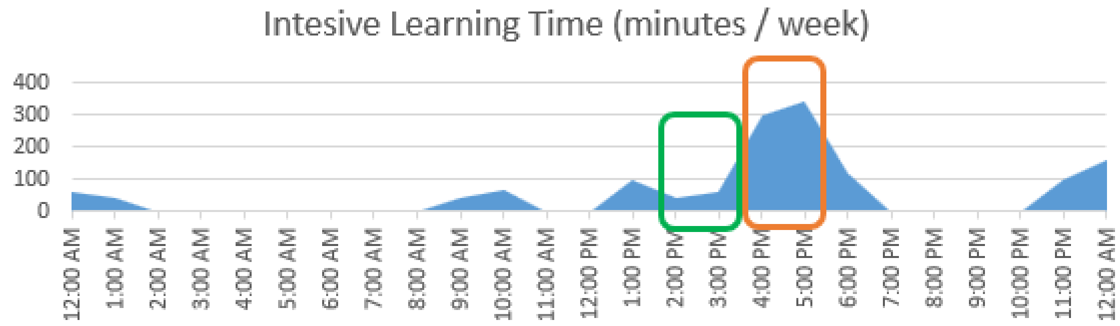
  This tactic is to give weight to time spent on activity, when effectiveness of activities is calculated.

  $$\text{activities' efficiency} = \frac{\sum_{q=0}^{numActivities}(evaluation\_result_q \times \alpha_q) \times activity\_weight_q}{\sum_{q=0}^{numActivities} perfect\_score_q \times activity\_weight_q}$$

  (where, $0.9 \leq \alpha_q \leq 1$) α_q is time factor. When activity is done before predefined time, it is 1. Every 10 percent of predefined time, it is decreased by 0.01. When it reaches 0.9 there is no more decrement.

- (T4) Use pattern analysis for learning effect (Relevant to C4)

  This tactic encourages Trainee at a time for intensive learning and continuously informs him or her the activities to be done. It helps Trainee to perform at that time.



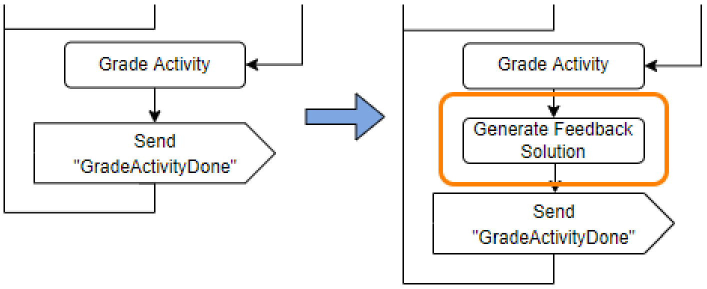
  It is implemented with timer on client-side application. Above figure is example of trainee's activity pattern. Let the orange rectangle as intensive time and the green as boosting time. The system continuously notifies trainee during boosting time, expecting the trainee to have given lecture or activities.

- (T5) Help trainee to get long-term memory (Relevant to C5)

  The iteration with lecture – activity can be more productive. This tactic describes 2 ways of improving subject memorization

  - Immediate Feedback with activity result

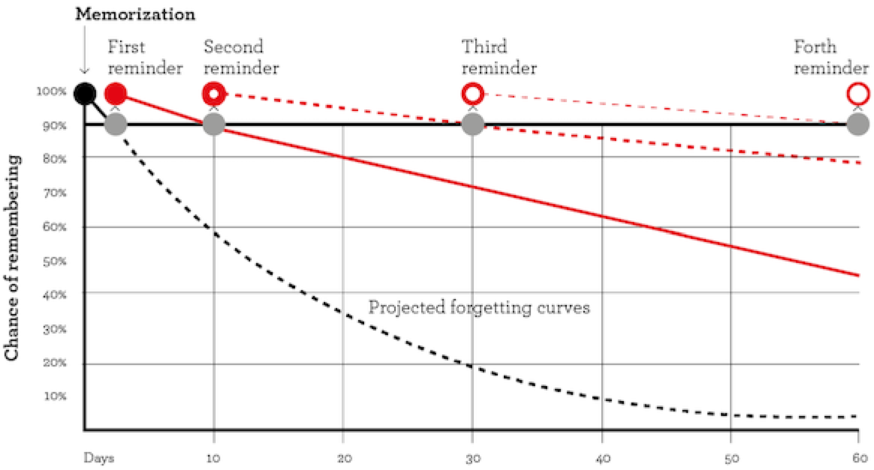
    One of the long-term memory method is testing effect. Every activity of system could be one of this effect. But the activity is finished at returning score only, the activity has less effective on memorizing. Giving immediate feedback solution for activity result helps carrying the memory for a long time.



Behavioral view control flow of grading activity

- Review contents after a period of time

  It is known as the Spacing effect of distributed practice.



A comparison of retention rates the use of distributed practice. Source: Farnam Street

It is implemented by expanding iterations of Offering. Between iteration, summarized content of lecture is randomly chosen and provided trainee to recall the content. It is enough to carrying 70% of memorization after a month, with first reminder within a week.

- (T6) Various learning experience (Relevant to C6)

  This tactic is to give different kinds of learning experience about a subject.

  - Form of contents

    The content can have form or textual, auditory and visual.

  - Additional materials

    A relational learning recommendation is also great way to become more effective. It can be any of above form of content.

6.1.4. [Step 4] Evaluation of the Candidate Tactics

We evaluate the proposed candidate tactics in terms of their benefit and cost.

ID	Tactics	Y/N	Justification
T1	Select Reinforcement Learning for Recommendation Model	Y	Benefit) The personalized recommendation provided with each trainee's policy from Reinforcement Learning. Cost) The computational and storage overhead for this tactic is minimal. Decision) The tactic can be implemented with a minimal cost, but the benefit is high.
T2	Increase the accuracy of initial recommendations	Y	Benefit) It can solve cold-start problems; thus, reliability of PLA service is ensured. Cost) Initial state, it consumes more resources to figure out trainee's group and update group policy. Decision) This tactic can be implemented with a small cost, and it is concentrated on initial stage.

T3	Considering time consumed at each activity	Y	Benefit) Performing time not only reflect efficiency, but also trainee's concentration. Cost) The computation overhead for this tactic is minimal. Decision) The tactic can be implemented with a minimal cost, but the benefit is high
T4	Use pattern analysis for learning effect	Y	Benefit) This tactic motivates learners to lead to effective learning. Cost) Some trainees may be reluctant to receive notifications. Decision) The tactic can be implemented as an optional feature.
T5	Help trainee to get long-term memory	Y	Benefit) This tactic improves trainee's satisfaction even after offering. Cost) More time and more human resource is needed. Decision) The tactic is important feature, need to be implemented even the cost is expected, but the benefit is high
T6	Various learning experience	N	Benefit) It stimulates trainee's interest and increases the learning effect. Cost) It requires more storage both client and server. Some trainees may not need various types of content. Decision) The tactic can be implemented as an expandable component of lecture.

6.1.5. [Step 5] Impact Analysis of Tactics on Views

We analyze the impacts of each selected tactic.

ID	Tactics	Functional View	Information View	Behavior View	Deployment View
T1	Select Reinforcement Learning for Recommendation Model		Define policy as a component	Update Control flow of conduct offering algorithm with RL	
T2	Increase the accuracy of initial recommendations	Define an attribute for taking survey and grouping trainees.	Define policy for groups and individual variation.	Update the control flow of enrollment.	
T3	Considering time consumed at each activity			Update effectiveness calculation algorithm	
T4	Use pattern analysis for learning effect	Define an attribute for pattern monitoring and notifier	Define cumulative pattern data	Add control flow for analysis and utilization	
T5	Help trainee to get long-term memory	Refine taking lecture attribute and method		Update control flow of performing activity.	

				Update conduct offering algorithm with reviewing lecture	

6.1.6. [Step 6] Architecture with Tactics Applied

We design the detailed control flow for selected tactics. In CEP, we choose only one tactic.

- **Target Tactic: T5. Help Trainee to Get Long-term Memory**
- **Apply random suggestion of lecture review on conduct offering algorithm**

  After a trainee completes set of lecture and activity and before next lecture is given, By following memorization retention graph on 6.1.4. (T5), reviewing within a week is enough to carrying 70% retention rate after a month. Lecture within a week is randomly selected and the summarized content of lecture is suggested to review.

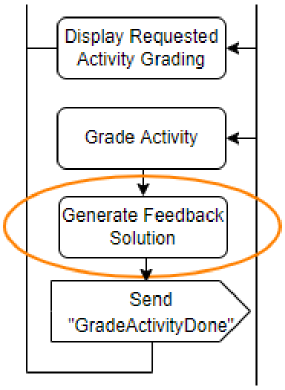
  - Algorithm changes for this tactic

```
For each $activity_m$ in activities
    $output\_of\_activities_m$ ← perform_activity($TRN_k$, $activity_m$)
    $evaluation\_result_m$ ← evaluate_activity($activity_m$, $output\_of\_activities_m$)
$efficiency_n$ ← $\frac{\sum_{q=0}^{numActivities} evaluation\_result_q \times activity\_weight_q}{\sum_{q=0}^{numActivities} perfect\_score_q \times activity\_weight_q}$
$course\_point_p$ ← average of { $efficiency_0$ , … , $efficiency_n$ }
update_progress_result($ITER^{i,k,p}_n$ , evaluation_result, $efficiency_n$, $course\_point_p$)
compute_achievement_level($PRGS^{i,k}$)
suggest_lecture_review($PRGS^{i,k}$)
optimize_recommendation($CRS^i_p$, $TRN_k$, $RCMD^{k,p}$, $efficiency_n$)
```

  - Method to generate feedback about activity added.
  - Progress contains all history of taken lecture and activities.
  - Lecture includes its summarization as a component and review history

- **Refine control flow of performing activity**

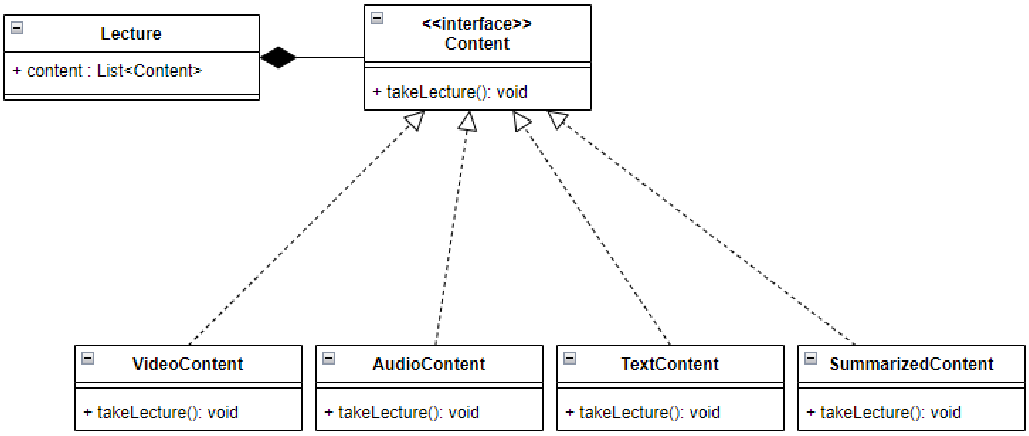
  Control flow of grading activity is in below, step for immediate generation of feedback solution added. The method for Feedback generation also defined. Above offering algorithm evaluate_activity() is high-lighted to show the modification.


Display Requested Activity Grading
Grade Activity
Generate Feedback Solution
Send "GradeActivityDone"

❑ **Apply strategy pattern for taking lecture including summarized lecture**

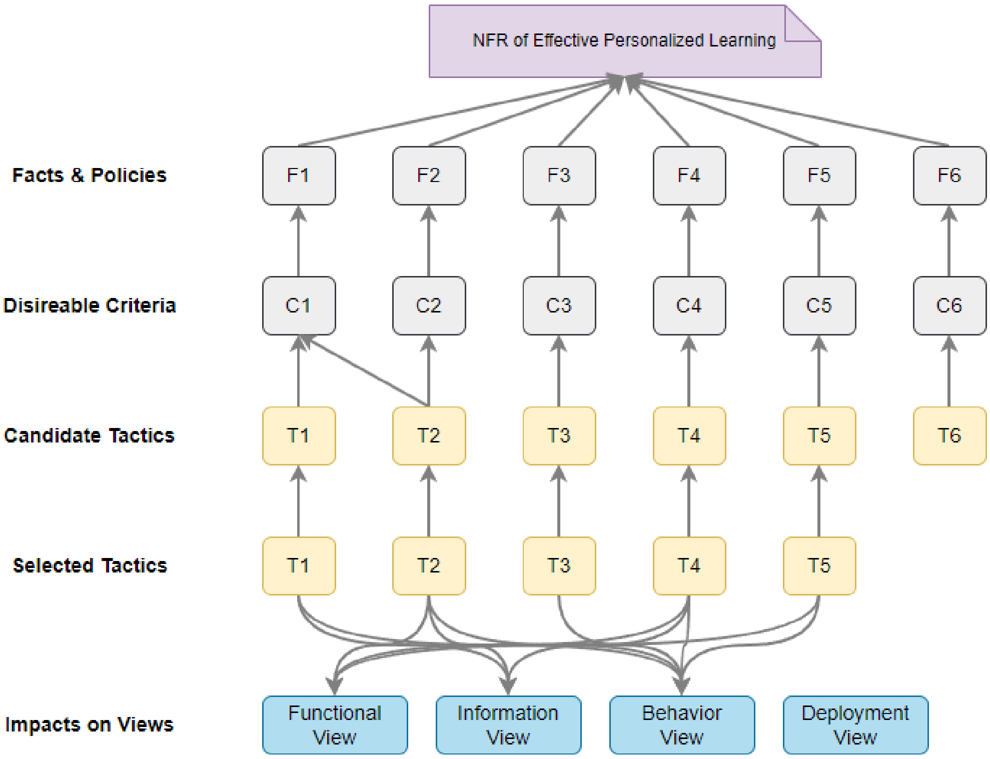
❍ suggest_lecture_review() algorithm follows below flow with refined Lecture object.

```
lecture = suggestLectureReview(progress);
reviewContent = getSummarizedLecture(lecture);
reviewContent.takeLecture();
```



6.1.7. [Step 7] Verifying the Traceability

It is important to enforce the traceability among the facts/policies, criteria, tactics and impacts on views. The following figure shows the trace links among facts, criteria, tactics and their impacts on views.



As shown in the figure, all the elements defined with ‘conforms-to’ relationships, which yields a high traceability and consistency.

6.2. Design for NFR-2. High QoS of the Cloud Service

The system should be designed to provide a high QoS as a cloud service on the following quality aspects.

❑ Reliability

The system should always be reliable.

❑ Availability

The system should always be available, not only being alive but also performing well.

❑ Scalability

The system should be scalable for potentially volatile invocation loads by users.

❑ Performance

The system should provide a high performance.

In CEP, Scalability for QoS is selected.

6.2.1. [Step 1] Underlying Facts and Policies

❑ (F1) Processing Capacity

User invocation could be extremely concentrated at a specific time and during a specific period.

❑ (F2) Storage Capacity

With diverging business environment, profiles of user and contents of program would be increase.

❑ (F3) Cost effective

Stakeholder wants to minimize cost for expansion.

6.2.2. [Step 2] Criteria for Satisfying NFR

❑ (C1) Highly Reactive System (Relevant to F1)

The system should be reacted to every single user invocation. There should be several ways to handle the entire invocations before scaling and during scaling up or out.

❑ (C2) Physical System Upgrade (Relevant to F1)

The physical system should be open for expansion.

❑ (C3) Minimizing Data Size (Relevant to F2, F3)

User profile(including progress) and Contents size should be optimized.

❑ (C4) Dynamic Storage Expansion (Relevant to F2)

Data Storage should accommodate all persistent data on the system and it should be accessible at a specific time.

- ❑ (C5) Demand Prediction (Relevant to F3)

  The system should allocate heavy and light users uniformly throughout servers.

- ❑ (C6) Adaptive System Capacity (Relevant to F3)

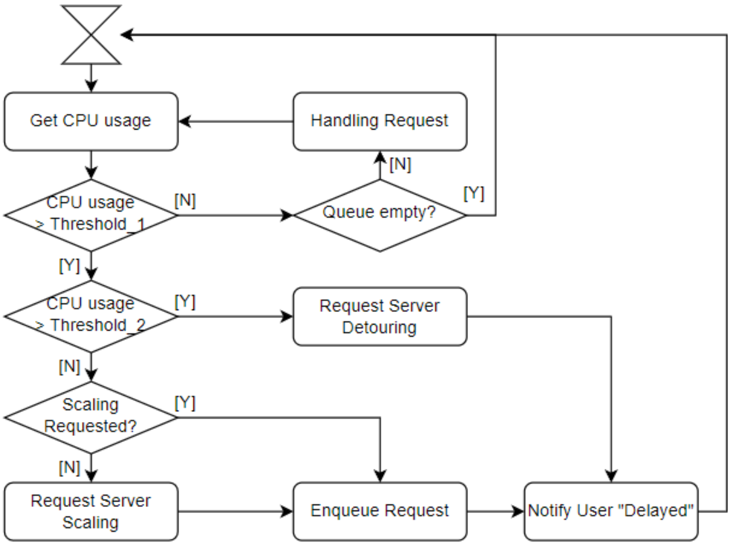
  The usage of system resources should be monitored. System resources under idle should be minimized.

6.2.3. [Step 3] Candidate Tactics for the Criteria

The following tactics are proposed for the identified criteria.

- ❑ (T1) Implementing delayed processing manager and notifier for service delays (Relevant to C1)

  Whether the capacity of system is expanding or not, every user invocation should be reacted by system immediately. The usage of CPUs on Server is monitored periodically and when it hits threshold delayed processing manger queuing and notify user about system delaying. The manager also requests more resources to main server which controls server scaling.



- ❑ (T2) Increasing server's capacity (Relevant to C2, C4)

  Increasing capacity can be realized in 2 ways. One is vertical scale-up. By adding more CPU and storage of a server, it brings less CPU usage under similar user invocation load. The another is horizontal scale-out. PLA system already reflect this by adding dispatcher and server replication, but scaling out more physical server may not cost-effective.

- ❑ (T3) Applying better video encoding for lecture contents (Relevant to C3)

  Lecture content takes majority share of data storage. The contents provided with various video format. It is essential applying effective video encoding that assure quality for user and size for store. With comparing most common encoding, MPEG-4 and H.264, PLA system can have more effective lecture contents with H.264.

- ❍ Advantages of H.264 over MPEG-4
  - ➢ About 1.5 to 2 times compression rate, small size for longer recording time
  - ➢ Fluent and better video quality for real time playback
  - ➢ Lower bit-rate required for network transmission
- ❍ Disadvantages of H.264 over MPEG-4
  - ➢ Longer encoding time
  - ➢ License cost

❑ (T4) Applying compression on completed offerings' data (Relevant to C3)

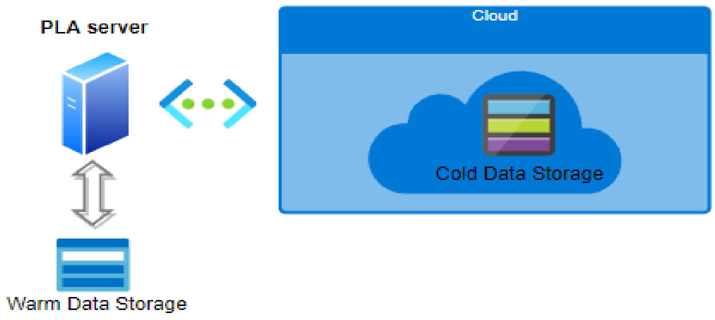
Progress Data of Offering should be flawless, thus PLA system applies 'Lossless compression algorithm' to satisfy both integrity and space saving.

Most popular lossless data compression algorithm are 7zip, bzip2 and gzip. With experiment with various file size with those algorithm, bzip2 shows best result. PLA system choose to apply bzip2 for data compression.

	Compression Ratio = Uncompressed Size / Compressed Size		
File Size	7z	Bzip2	gzip
4KB	1.8	1.9	1.8
238KB	5.7	5.5	4.9
2MB	11.9	19.2	9.3
5MB	9.8	10.5	8.0

❑ (T5) Handling grey data with backup storage (Relevant to C4)

Taking cold cloud storage to prevent scaling. Grey data; the data has not been accessed in a long period of time. This grey data is gradually accumulated and will occupy most of the server storage later. Physical server storage is limited and it is inevitable to scaling up the storage by growing business.



The data also valuable and should be preserved. By moving these data to another storage, server can be free from storage shortage and the occurrence of scaling can be depend on CPU's usage only. When it comes to cost, cloud storage provided with a very reasonable cost while physical storage costs as much as 100 times than cloud.

- ❑ (T6) Clustering Users by uptime (Relevant to C5)

  Implementing Uptime manager to calculated daily usage by profiling user's activity over system. PLA system divided users into 3-groups (heavy – medium – light user group). The classification marked on user profile and it is used by dispatcher allocating proper server for client connection.

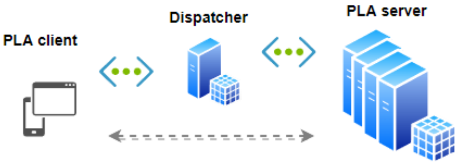
  From client, each user's uptime can be measured. Uptime consists of two distinct values, total connected time with server and Total active time with server.

  Daily collected values are sent to server's uptime manager and used as clustering 3-groups.

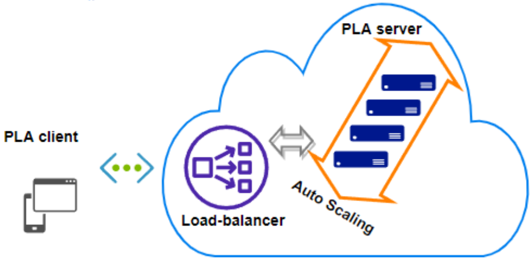
  The cluster divided in equal proportion. This group is used as a parameter that Dispatcher allocates server for load-balancing.

- ❑ (T7) Adopting IaaS for auto-scaling (Relevant to C4, C6)

  Auto-scaling a feature that increases or decreases resources based on workload. With IaaS, PLA system can have advantages of less effort for server maintenance and resilient scaling of server resources including scale–in.



PLA system without IaaS



PLA system with IaaS

  Major IaaS providers(AWS, Google Cloud and Azure) all support auto-scaling. When the auto-scaling is applied, the resources are exactly expanded by workload; it is cost-effective as there is no unnecessary resource expansion.

6.2.4. [Step 4] Evaluation of the Candidate Tactics

We evaluate the proposed candidate tactics in terms of their benefit and cost.

ID	Tactics	Y/N	Justification
T1	Implementing delayed processing manager and notifier for service delays	Y	Benefit) Users can notice current delay and have tolerance about their request. Server can handle every user invocation with delay. Cost) Repeated delaying on user invocation makes this system not attractive. Decision) The tactic can be implemented and the cost is inevitable under heavy-load, but the benefit is high.
T2	Increasing server's capacity (already applied)	Y	Benefit) The handling of variability increases the detection coverage of regular activities.

			Cost) The complexity of figuring and customizing the variability is minimal. Decision) This tactic can be implemented with a minimal cost, but the detection coverage is increased.
T3	Applying better video encoding for lecture contents	N	Benefit) Contents quality is better with less size. It reduces occurrence of system scaling. Cost) More encoding time for contents, license cost needed Decision) Stakeholder does not want increasing of running cost. Take MPEG-4 for video encoding
T4	Applying compression on completed offerings' data	Y	Benefit) End of offering, user's data can be managed with minimal size. It saves storage capacity and reduces occurrence of system scaling. Cost) Compression/ Decompression time consumed Decision) This tactic can be implemented with a minimal cost. the benefit is also high.
T5	Handling grey data with backup storage	Y	Benefit) After Offering finished, all the related data is grey data. This tactic reduces demand of server resource and prevent scaling-up of server. Cost) Cloud storage costs reasonable price. Decision) This tactic can be applied with a minimal cost. the benefit is also high.
T6	Clustering Users by uptime	Y	Benefit) It helps dispatcher to choose optimal server that makes well-estimation of load-balancing. Cost) Computational load on client. Decision) This tactic can be implemented, client-side cost is acceptable.
T7	Adopting IaaS for auto-scaling	Y	Benefit) The scaling out and in it automated by IaaS service provider. PLA system can be managed with matched amount resources to active load. Cost) Depend on IaaS service provider. Decision) The tactic can be adopted with a minimal cost, but the benefit is high.

6.2.5. [Step 5] Impact Analysis of Tactics on Views

We analyze the impacts of each selected tactic.

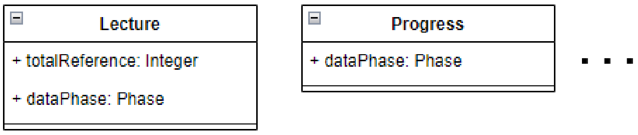
ID	Tactics	Functional View	Information View	Behavior View	Deployment View
T1	Implementing delayed processing manager and notifier for service delays	Define method for handling notified message.	Define 'request queue' component	Add control flow to handle delayed user request.	
T2	Increasing server's capacity (already applied)				Replication with Dispatcher (Already applied)
T4	Applying compression on completed	Define method to compress / decompress data		Refine control flow of finished offering	

	offerings' data				
T5	Handling grey data with backup storage	Define attribute and method to support grey data migration	Define persistent data on cold storage	Define control flow of grey data migration	Allocate cold storage server
T6	Clustering Users by uptime	Define an attribute and method for monitoring server-access and active load of trainee.		Refine control flow of server allocator for dispatcher.	
T7	Adopting IaaS for auto-scaling				Server implemented inside IaaS service provider

6.2.6. [Step 6] Architecture with Tactics Applied

We design the detailed control flow for selected tactics. In CEP, we choose only one tactic.

- **Target Tactic: T5. Handling grey data with backup storage**
- **Add attribute that represents Data Phase**



Data Phase has 3 step. Creation, in-Premise, and storage.

The most size of storage consumed by lectures and trainees' progress data.

When dataPhase attribute represents a storage phase, it means the data resides in cold storage.

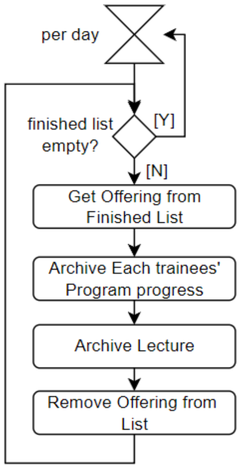
- **Define persistent data on cold storage.**

D.progress and D.lecture defined to represent migrated data on cold storage server.

Data Components	PLA Client	PLA Server	Cold Storage Server
Lecture	C.Lecture	S.Lecture	D.Lecture
Progress	C.Progress	S.Progress	D.Progress

- **Define flow control for data migration**

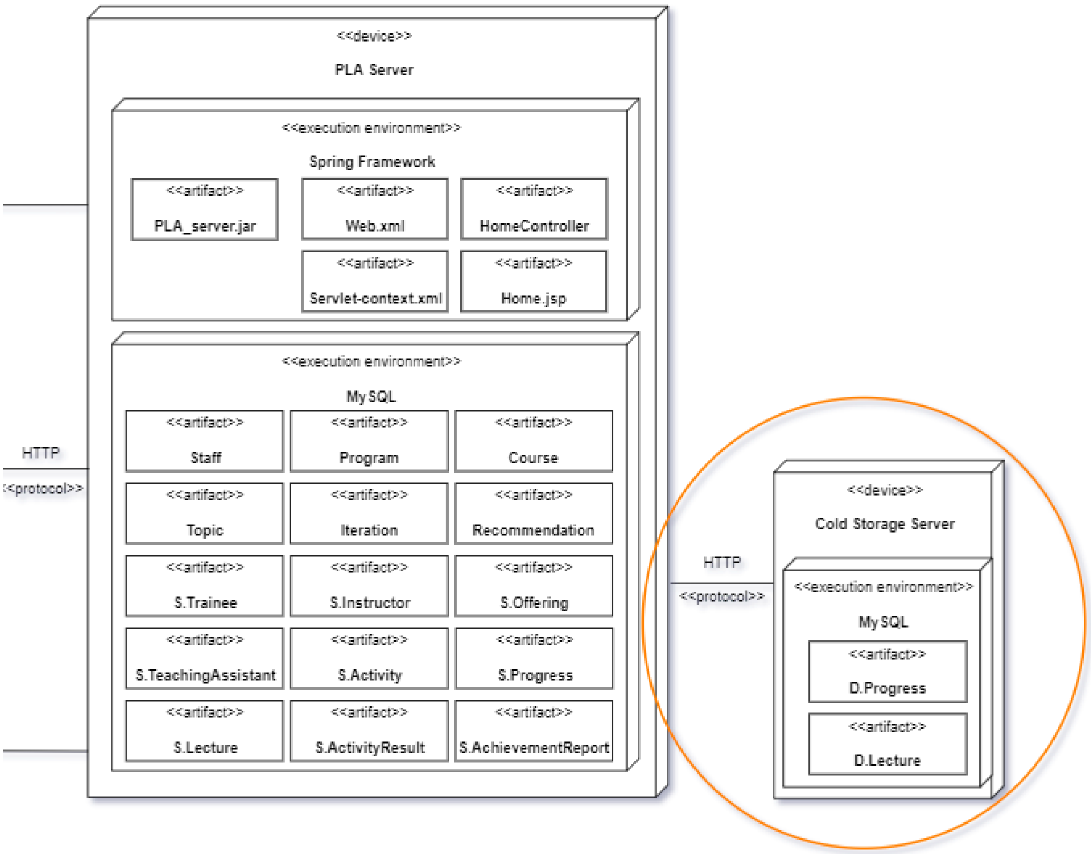
Archiving data has not been considers in PLA system. New functionality for data migration is taken by offering manger and flow control for migration described as below diagram.


per day
finished list empty?
[Y]
[N]
Get Offering from Finished List
Archive Each trainees' Program progress
Archive Lecture
Remove Offering from List

Candidate of data migration can be extended to every persist data on server Database.

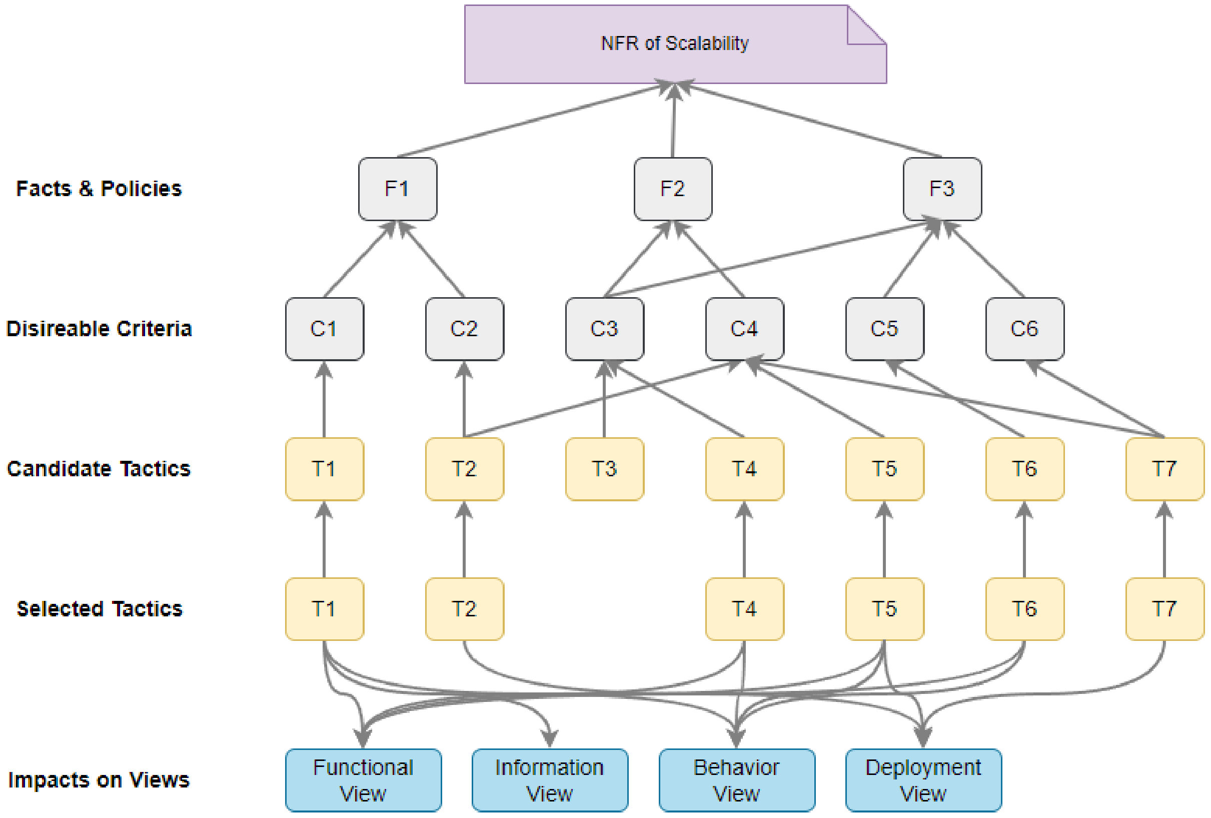
- **Allocate Cold Storage server**

Extend storage and define Cold Storage Server from PLA server to store inactive data.



6.2.7. [Step 7] Verifying the Traceability

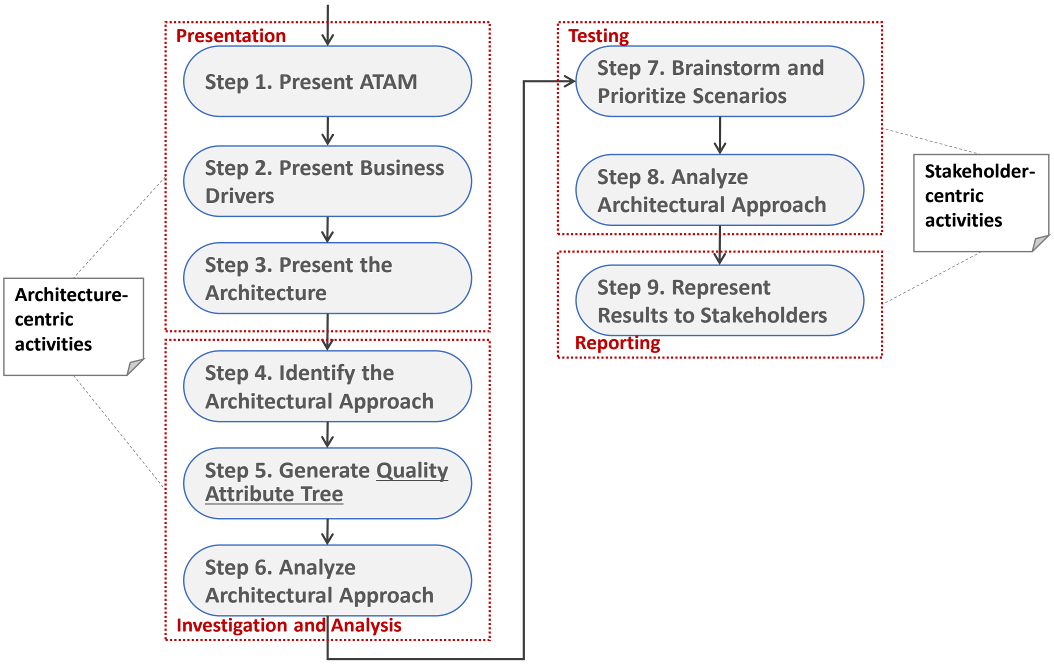
It is important to enforce the traceability among the facts/policies, criteria, tactics and impacts on views. The following figure shows the trace links among facts, criteria, tactics and their impacts on views.



As shown in the figure, all the elements defined with 'conforms-to' relationships, which yields a high traceability and consistency.

7. Activity 6. Architecture Validation

- ❑ Steps of ATAM


Presentation
Step 1. Present ATAM
Step 2. Present Business Drivers
Step 3. Present the Architecture
Step 4. Identify the Architectural Approach
Step 5. Generate Quality Attribute Tree
Step 6. Analyze Architectural Approach
Investigation and Analysis
Testing
Step 7. Brainstorm and Prioritize Scenarios
Step 8. Analyze Architectural Approach
Step 9. Represent Results to Stakeholders
Reporting
Architecture-centric activities
Stakeholder-centric activities

7.1. [Step 1] Presenting ATAM

- ❑ Omitted

7.2. [Step 2] Presenting Business Drivers

- ❑ Omitted

7.3. [Step 3] Presenting the Architecture

- ❑ Omitted

7.4. [Step 4] Identifying the Architectural Approaches

The architectural approaches applied to *PLA System* are described in chapters 5 and 6. The architectural approaches are reflected in the following view;

- ❑ Functional View
- ❑ Information View
- ❑ Behavior View
- ❑ Deployment View

7.5. [Step 5] Generating Quality Attribute Tree

The following table describes the quality attribute utility tree used for evaluating the architecture of *PLA System*. We define 4 scenarios for evaluating functionality and for 5 scenarios for evaluating scalability of the *PLA System*. Each scenario is described with a different degree of importance and difficulty.

Quality	Refinements	Scenarios	Importance	Difficulty
Functionality	NFR-1. High Effectiveness of Personalized Learning	NFR-1-1. Effectiveness of recommendation measured and it should be higher than 50% after half of offering.	High	Medium
		NFR-1-2. All activities have deadline, and the activities should be recommended at a level of difficulty that meets the deadline.	Medium	Low
		NFR-1-3. After a month, proficiency level of trainees should be remained 70% to the end of offering.	Medium	High
		NFR-1-4. For effective of recommendation, similar preference group of trainee have similar recommendation. Evaluate effectiveness by group recommendation.	Medium	High
Scalability	NFR-2. High QoS of Cloud Service	NFR-2-1. Evaluate Workload by user invocation. It should be maintained under 80%.	Medium	Medium
		NFR-2-2. Every user invocation should be handled at high workload.	High	Medium
		NFR-2-3. User should not notice delay on invocation.	Medium	Medium
		NFR-2-4. Inactive data should be compressed and stored different storage. Total inactive data should be maintained under 30%.	High	High
		NFR-2-5. Evaluate trainees' active time on server. Grouping trainees to 3 groups by evaluated value. Heavy, normal, light users on server should be proportional.	Medium	Low

7.6. [Step 6] Analyzing Architectural Approaches

The following table describes an analysis result of architectural approaches addressing a scenario, NFR-2-4.

Analysis of Architectural Approach	
Scenario #	NFR-2-3. User should not notice delay on invocation.
Attribute	Functionality

Environment	Normal Operation, Operation over huge volatile invocation loads.			
Stimulus	CPU usage higher than 90%, scaling of system is requested.			
Architectural Decision	**Sensitivity**	**Trade-Off**	**Risk**	**Nonrisk**
D1-1. Handling invocation with deferred queue		T1-1-1	R1-1-1	
D1-2. Distribute trainees by their active time				N1-1-1
D1-3 Move out inactive data to cold storage	S1-1-1	T1-1-2	R1-1-2	
D1-4 Use IaaS for auto-scaling		T1-1-3	R1-1-3	
Reasoning	The decisions are made for meeting this scenario since the chosen decisions are commonly used for improving scalability.			
Architectural Diagram	Refer to refined component diagram in 6.2.			

- ❑ Sensitivity Points
  - ❍ S1-1-1. Computational load of data compression and migration
- ❑ Trade-off
  - ❍ T1-1-1. Responsiveness (+) vs. Processing time (-): The process of using deferred queue make immediate response to user and let user understanding current environment. Every invocation can be handled but overall processing time increases.
  - ❍ T1-1-2. Resource availability (+) vs. Cost effectiveness (-): Migration data improves data storage's availability. Managing additional storage needs physical and operational cost.
  - ❍ T1-1-3. Scalability (+) vs. Security, Dependency (-): IaaS is cost-effective at scaling server out, but also scaling in. All of server data is not in physical device. It gives lack of security and dependency on IaaS service provider.
- ❑ Risk
  - ❍ R1-1-1. Even user got the response from server, the invocation takes time to finish its operation. It makes inconvenience to user by giving waiting time.
  - ❍ R1-1-2. The risk is caused since expanding storage makes cost. Managing more storage also makes maintenance cost.
  - ❍ R1-1-3. This risk is caused since IaaS is virtualization service over cloud. System depends on IaaS provider. System does not have any control over cloud security.
- ❑ Nonrisk
  - ❍ N1-1-1. Distribute trainees to proper server does not have risk.

7.7. [Step 7] Brainstorming and Prioritizing Scenarios

- ❑ List of scenarios collected by all stakeholders (i.e. clients, staffs, trainees, and instructors)
  - ❍ About 4 scenarios are additionally acquired for evaluating functionality of PLA system.

Most scenarios are gathered from clients, staffs and instructors.

- ➢ Need more convincible recommendation and measurable harness for cold-start trainee.
- ➢ Provide different level, length of lecture by trainee's proficiency level.
- ➢ Detecting outlier from activity result.
- ➢ Measure offered program's effectiveness over the subject
- ➢ …

- ❍ About 2 scenarios are additionally acquired for evaluating scalability for QoS. Most scenarios are gathered from clients.
  - ➢ Measure the time of scaling on normal operation and on heavy work load.
  - ➢ Measure the restore data latency from cold-storage to warm-storage.
  - ➢ …

7.8. [Step 8] Analyzing Architectural Approaches

Since the result of this step is same as the one of step 6, we do not include the table.

7.9. [Step 9] Presenting the Results

- ❑ All evaluation team concludes the following results;
  - ❍ Concerns on the functionality, specifically on high effectiveness of personalized learning, need to be re-considered.
    - ➢ The recommendation model suggests not only activities but also lectures to learn. Reinforcement Learning could make effectiveness set of activities, but when it comes to lecture, there is low interconnection at finding next lecture to learn. When recommendation model for lecture is divided and takes proper method, personalized learning will have much higher effectiveness.
  - ❍ Concerns on the scalability also need to be re-considered.
    - ➢ Since the initial scenarios did not cover most of the load-balancing cases, it seems like that there might be some flaws in the design. A more design approach is needed.

8. Concluding Remarks

The architecture description in this document is to meet both the functional and non-functional requirements for the system. It is the result of applying the proposed core process of designing software architecture.

It is believed that this architecture description is practically implementable with current technologies and such implementation would yield a high level of quality-in-use.

➢ END OF ARCHITECTURE DESCRIPTION ≺