WITS
UNIVERSITY

Short course system

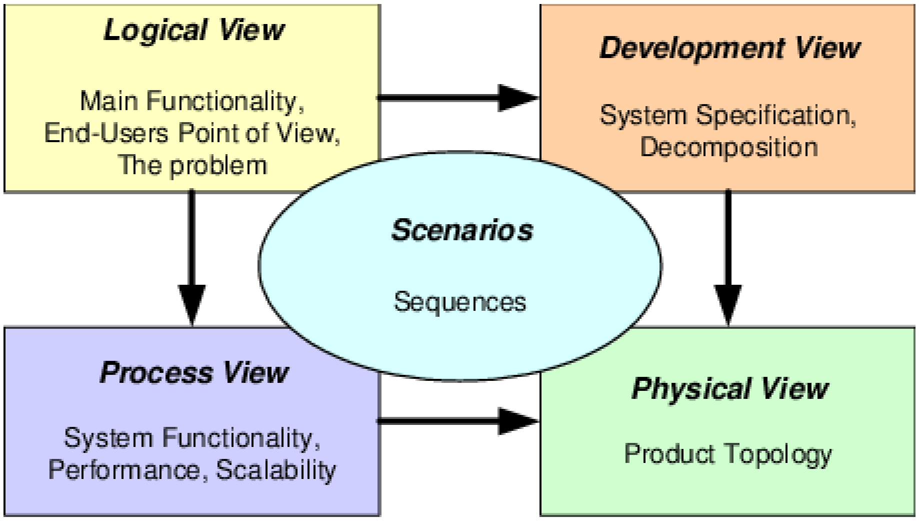
.Given Mathebula
.Khotsofalang Tsunyane
.Lindokuhle Mabena
.Mosis Mdlalose
.Bonginkosi Mnguni
.Nkosinathi Chuma
.Karabo Maila
.Mandilakhe Deyi
.Faith Moraka Moruti

09 may 2022

Contents

2 Sprint 3 Diagrams

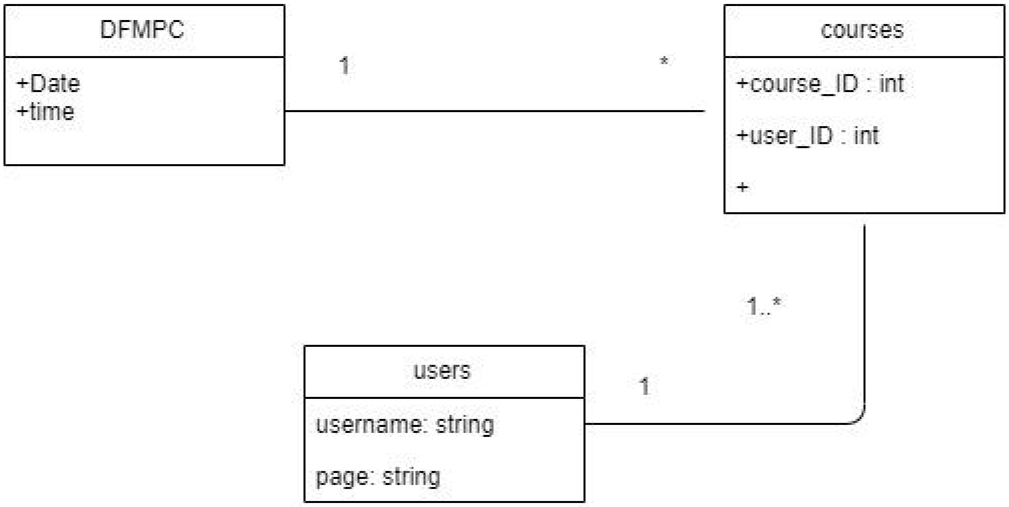
4 + 1 architecture



Logical View

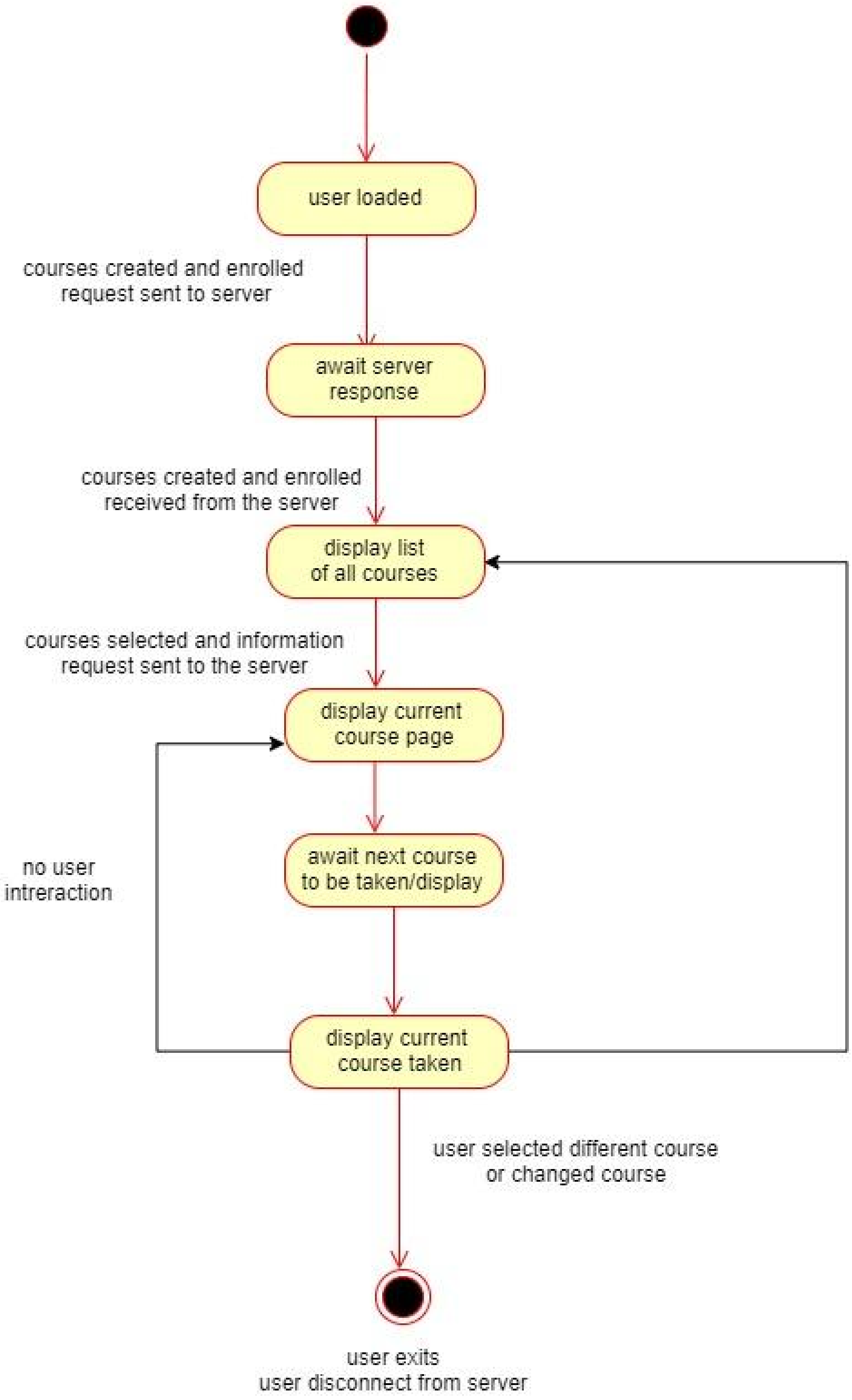
The logical view is concerned with the functionality provided to users and focuses on the decomposition of the system into a set of key abstractions, namely objects and classes.
 We represent it in UML using class diagrams and state diagrams

Class Diagram: Static view of the classes/objects in a system



State Diagram/statechart diagrams: give a dynamic view of objects Define states for each object and how objects transition between states Models and object’s lifetime from creation to termination

1.1 STATE DIAGRAM

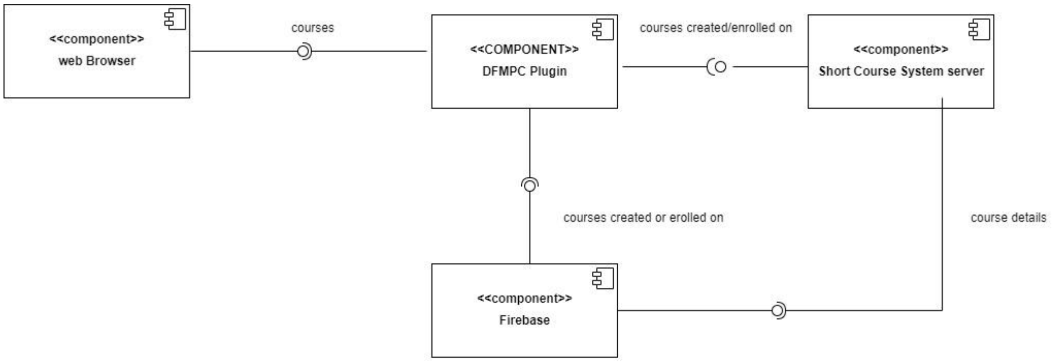


1.2 Development View: The development view is concerned with software administration from the point of view of the programmer

Focuses on components: modules of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.

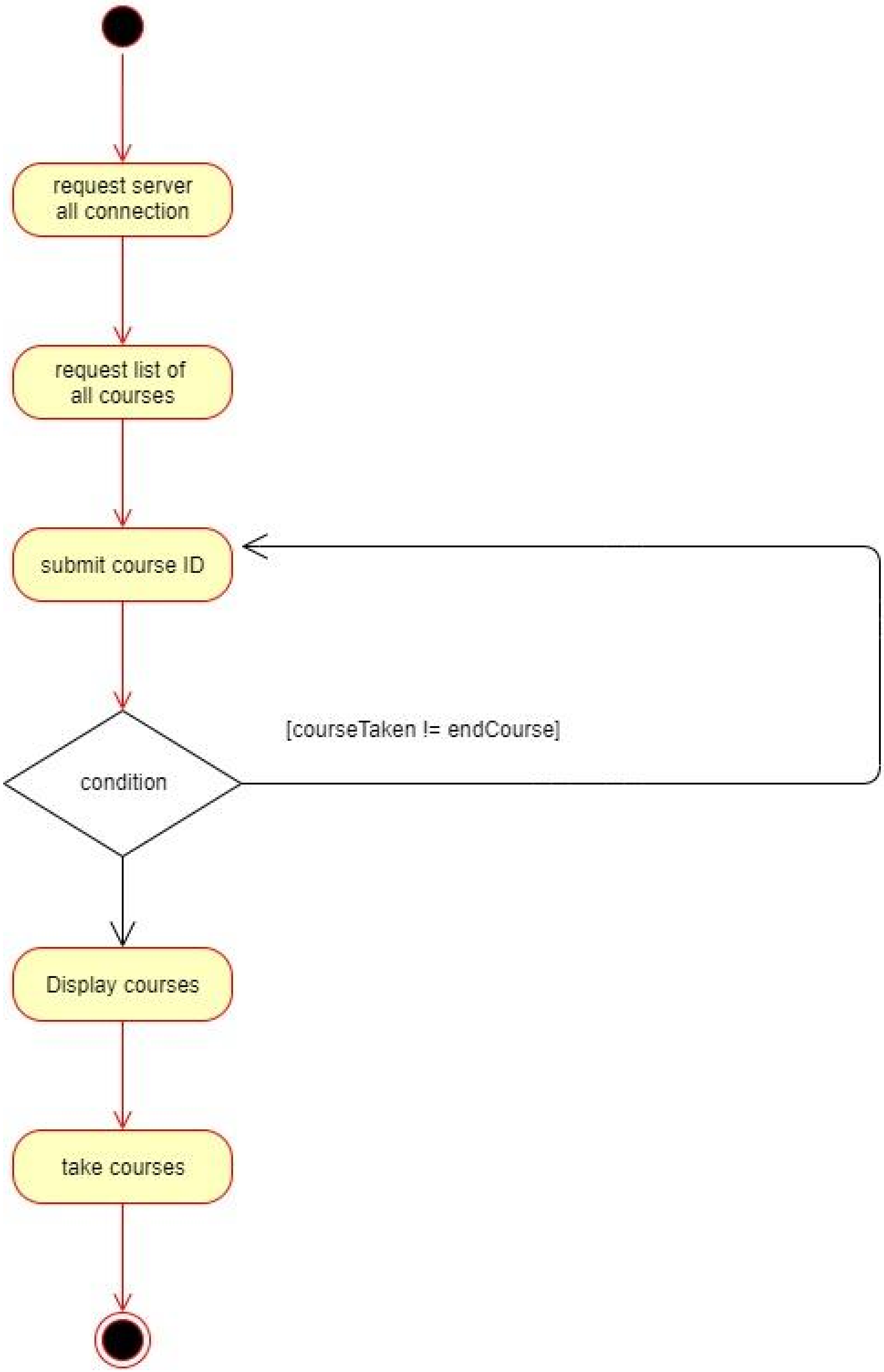
Represented in UML using component diagrams

1.2.1 component Diagram

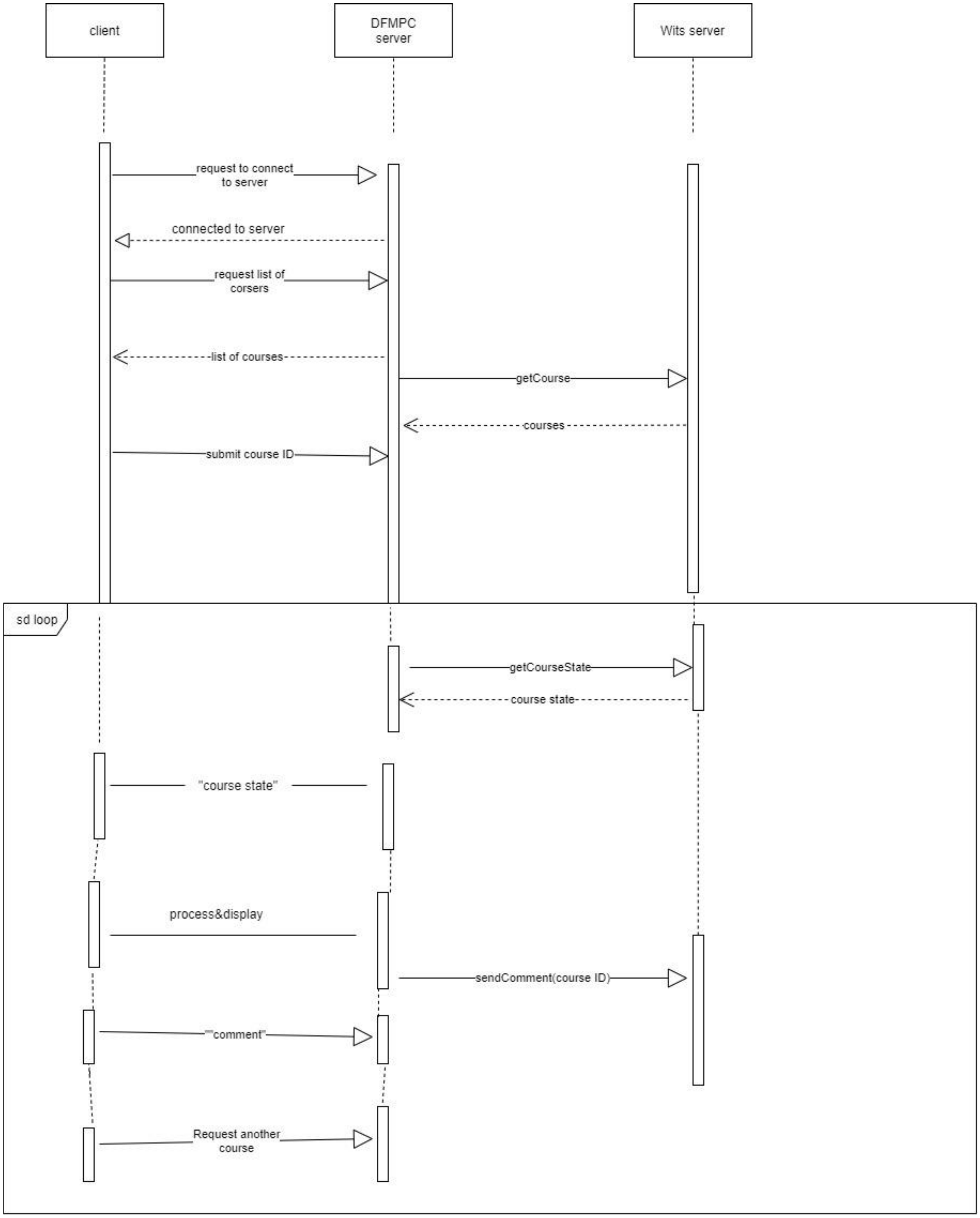


1.3 Process View: The process view is concerned with the dynamic aspects of the system, explaining the system processes and how they communicate
Focuses on the runtime behaviour of the system
Represented in UML using activity diagrams and sequence diagrams

1.3.1 Activity Diagram: show the workflow of a system

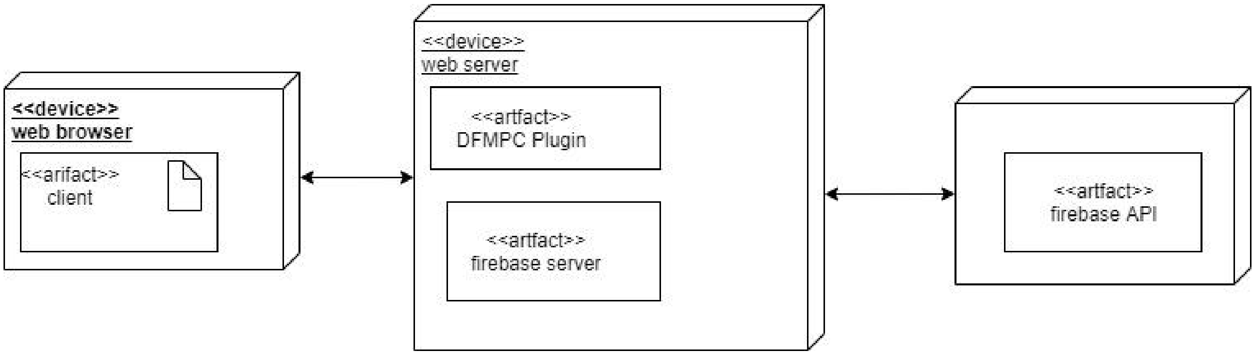

request server all connection
request list of all courses
submit course ID
[courseTaken != endCourse]
condition
Display courses
take courses

1.3.2 Sequence Diagram: show how components/objects in a system interact over time

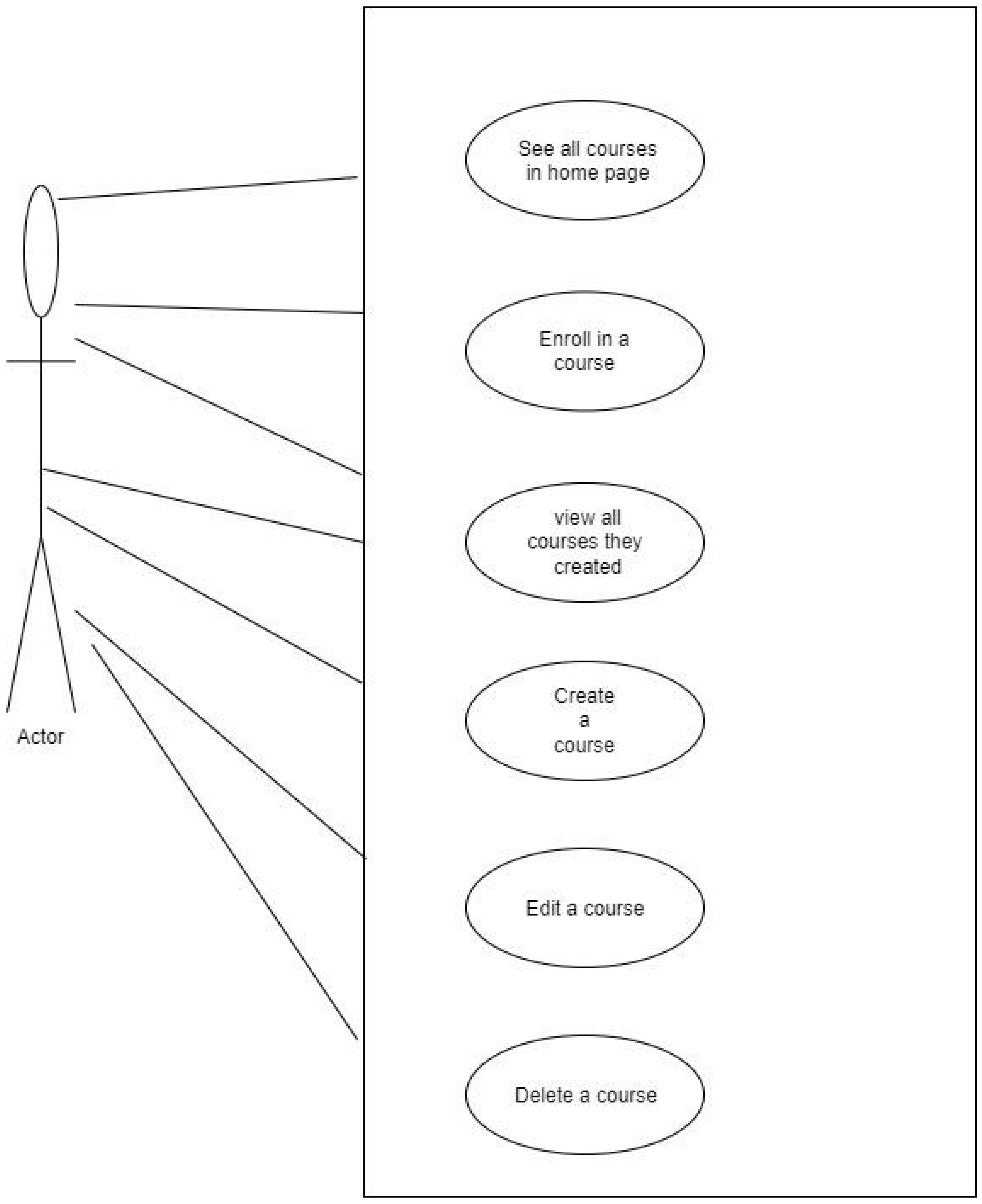


1.4 Physical View: is concerned with the physical distribution of the system, as well as the physical connections between distributed parts
Depicts the system from a system engineer's point of view
Here we represent it in UML using deployment diagrams

1.4.1 Development Diagram: model the physical deployment of artifacts on nodes



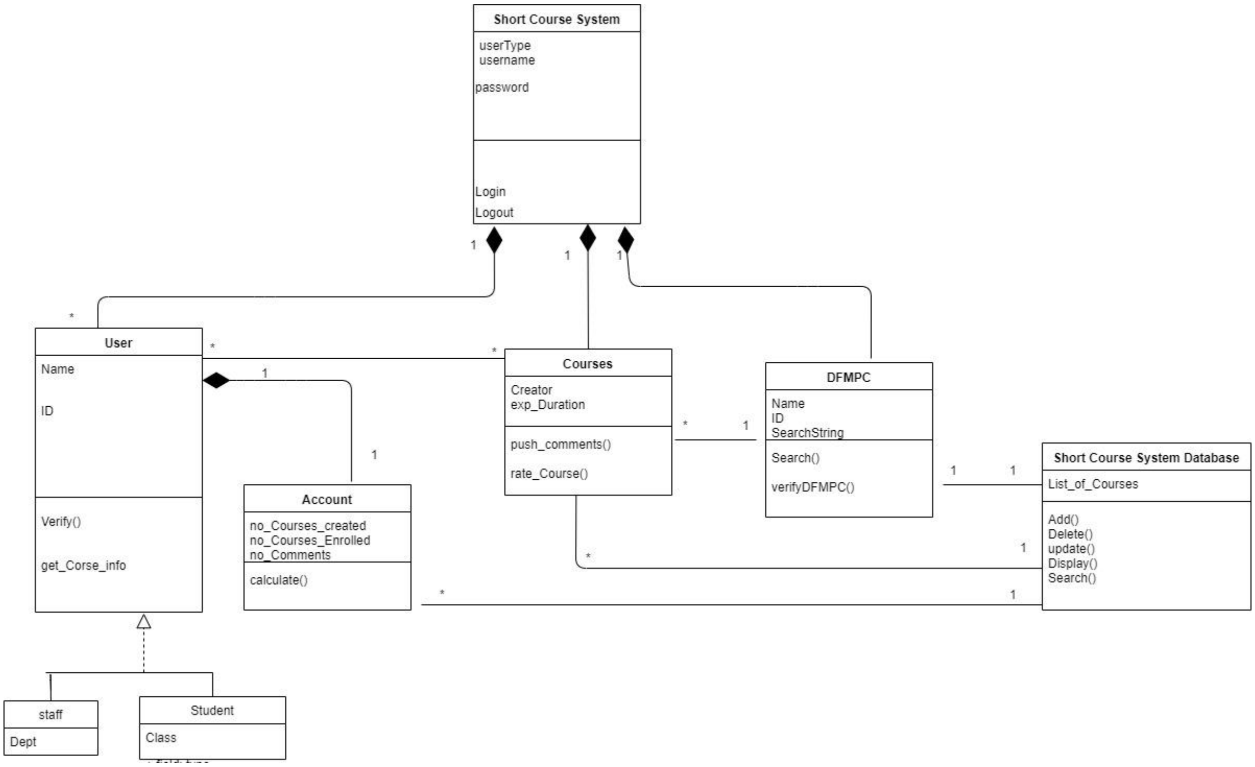
1.5 Use case Diagram: A use case diagram is **used to represent the dynamic behaviour of a system**. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application.


See all courses in home page
Enroll in a course
view all courses they created
Create a course
Edit a course
Delete a course
Actor

Sprint 3

2.1 Logical Diagram: is concerned with the functionality provided to users and focuses on the decomposition of the system into a set of key abstractions, namely objects and classes.
We represent it in UML using class diagrams and state diagrams

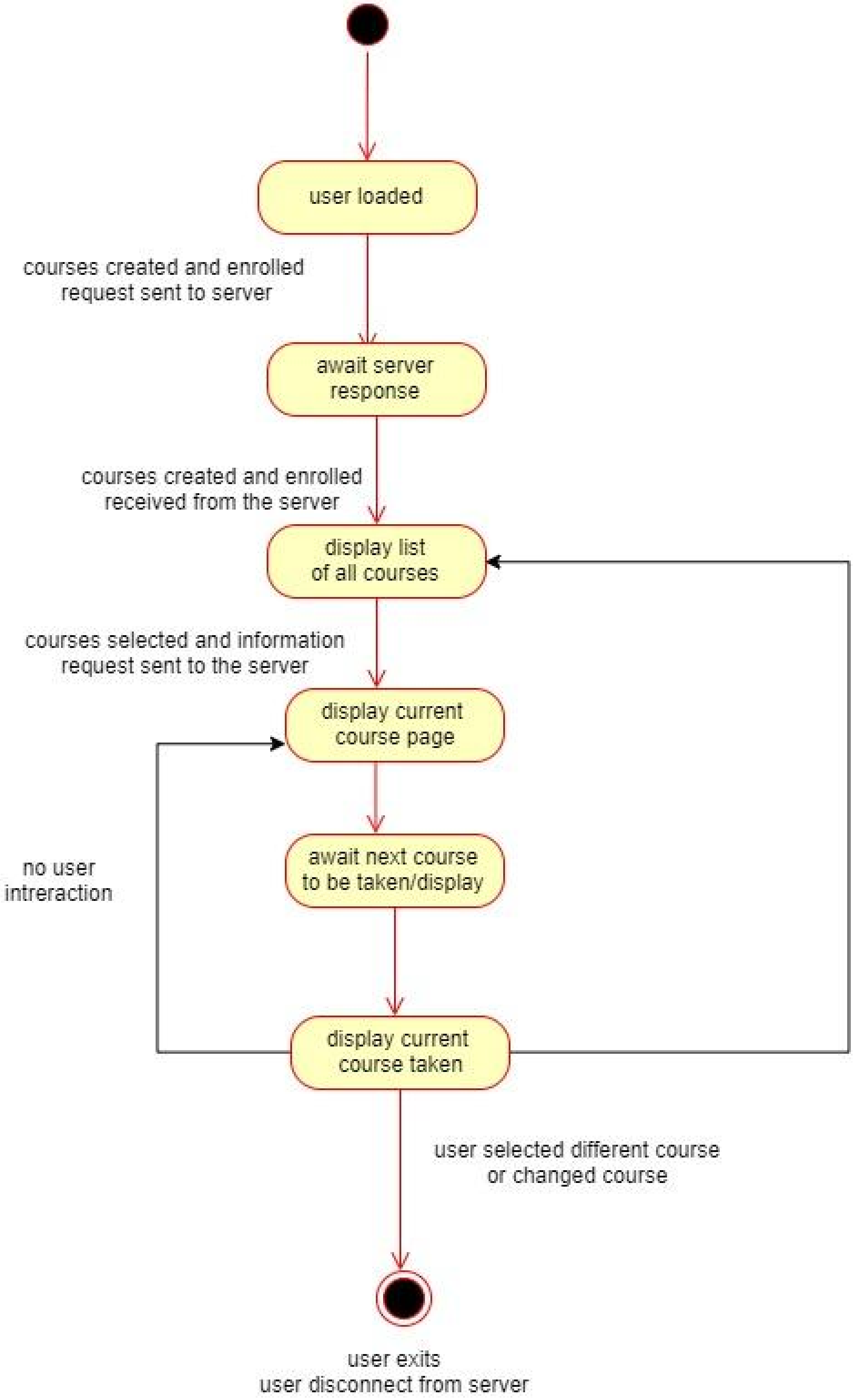
2.1.1 Class Diagram



The class diagram outline the relationship between classes and also allows for the analysis of the system as a whole

2.1.2 **State Diagram:** give a dynamic view of objects Define states for each object and how objects transition between states Models and object’s lifetime from creation to termination

1.1 STATE DIAGRAM

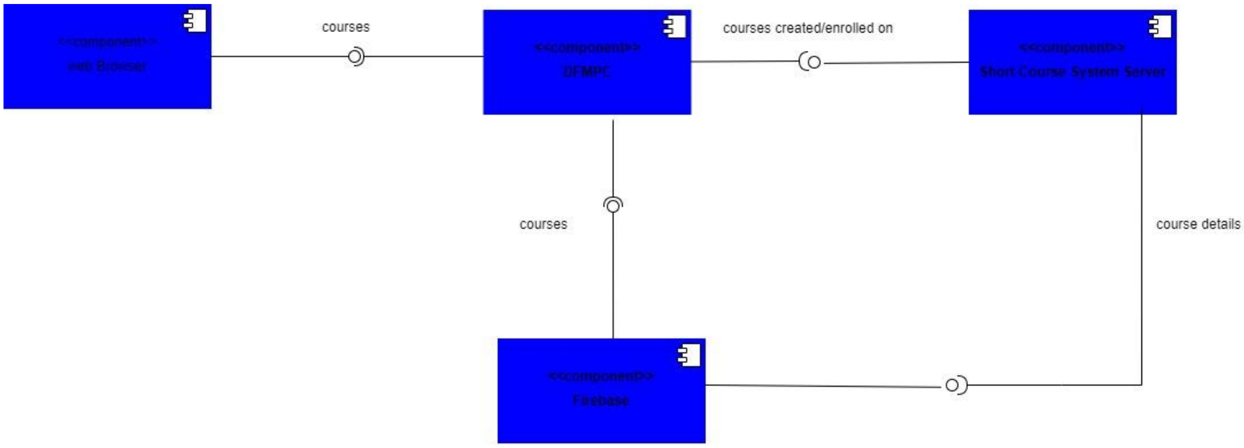

user loaded
courses created and enrolled request sent to server
await server response
courses created and enrolled received from the server
display list of all courses
courses selected and information request sent to the server
display current course page
await next course to be taken/display
no user intreraction
display current course taken
user selected different course or changed course
user exits
user disconnect from server

The above diagram shows how the user(Wits staff/student) interacts with the Short Course System. The user sees the list of all courses on our system and they can choose to enroll on the Course. They can also create their own slides. They are going to follow all our required steps to create their slides, if everything is verified their courses/slides will be added to their created courses successfully.

State Diagram 2.1.2b

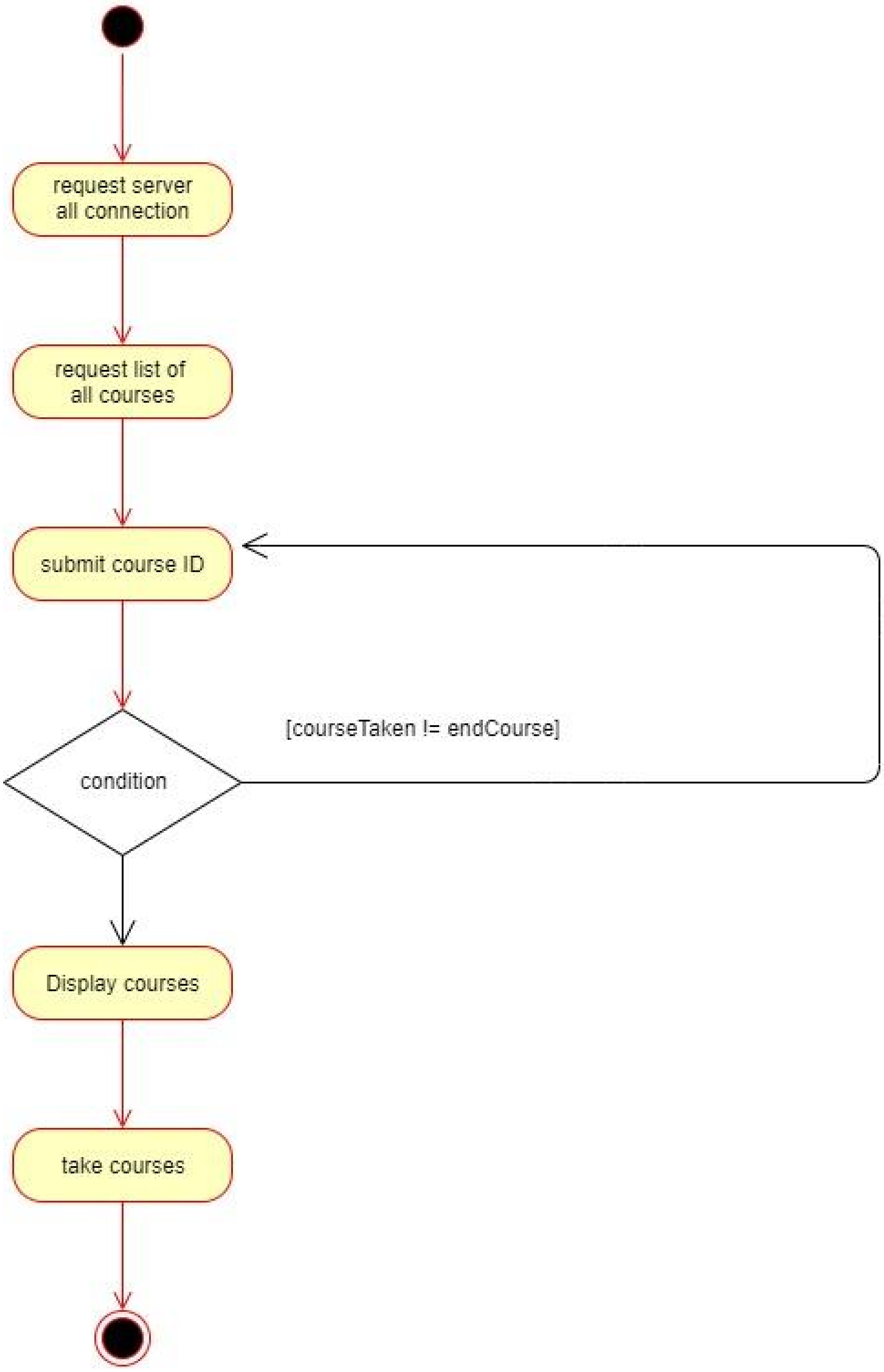
2.2Development View: The development view is concerned with software administration from the point of view of the programmer
Focuses on components: modules of classes that represent independent systems or subsystems with the ability to interface with the rest of the system.
Represented in UML using component diagrams

2.2.1 component Diagram: show how different components in a system interact

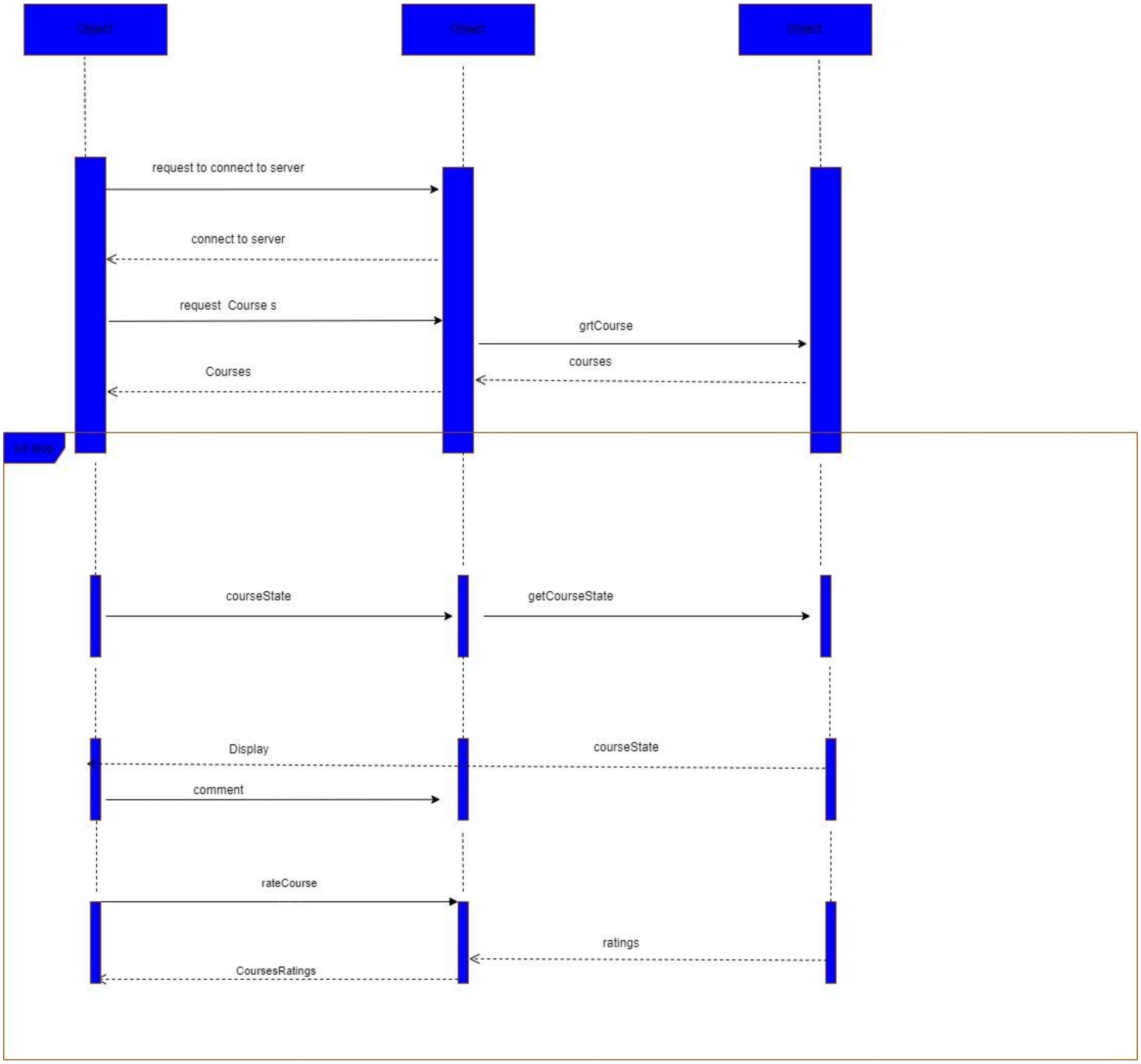


2.3 Process View: The process view is concerned with the dynamic aspects of the system, explaining the system processes and how they communicate
Focuses on the runtime behaviour of the system
Represented in UML using activity diagrams and sequence diagrams

2.3.1 Activity Diagram: show the workflow of a system

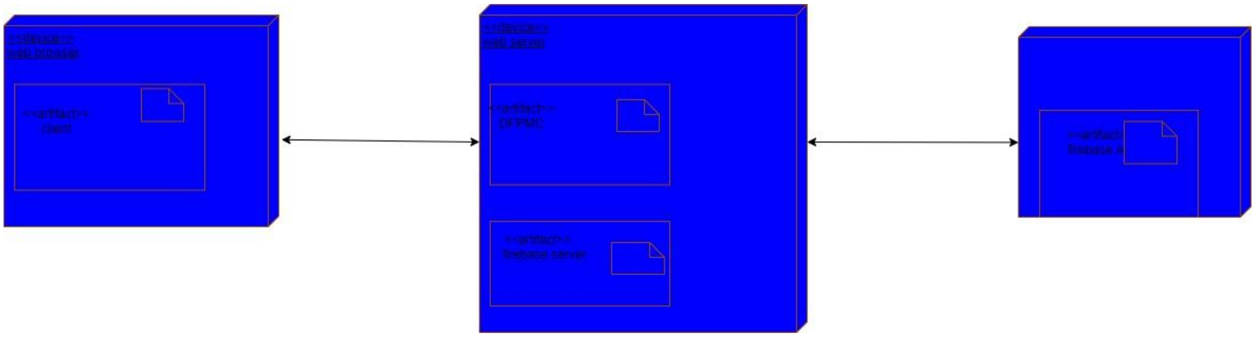

request server all connection
request list of all courses
submit course ID
condition
[courseTaken != endCourse]
Display courses
take courses

2.3.2 Sequence Diagram: show how components/objects in a system interact over time

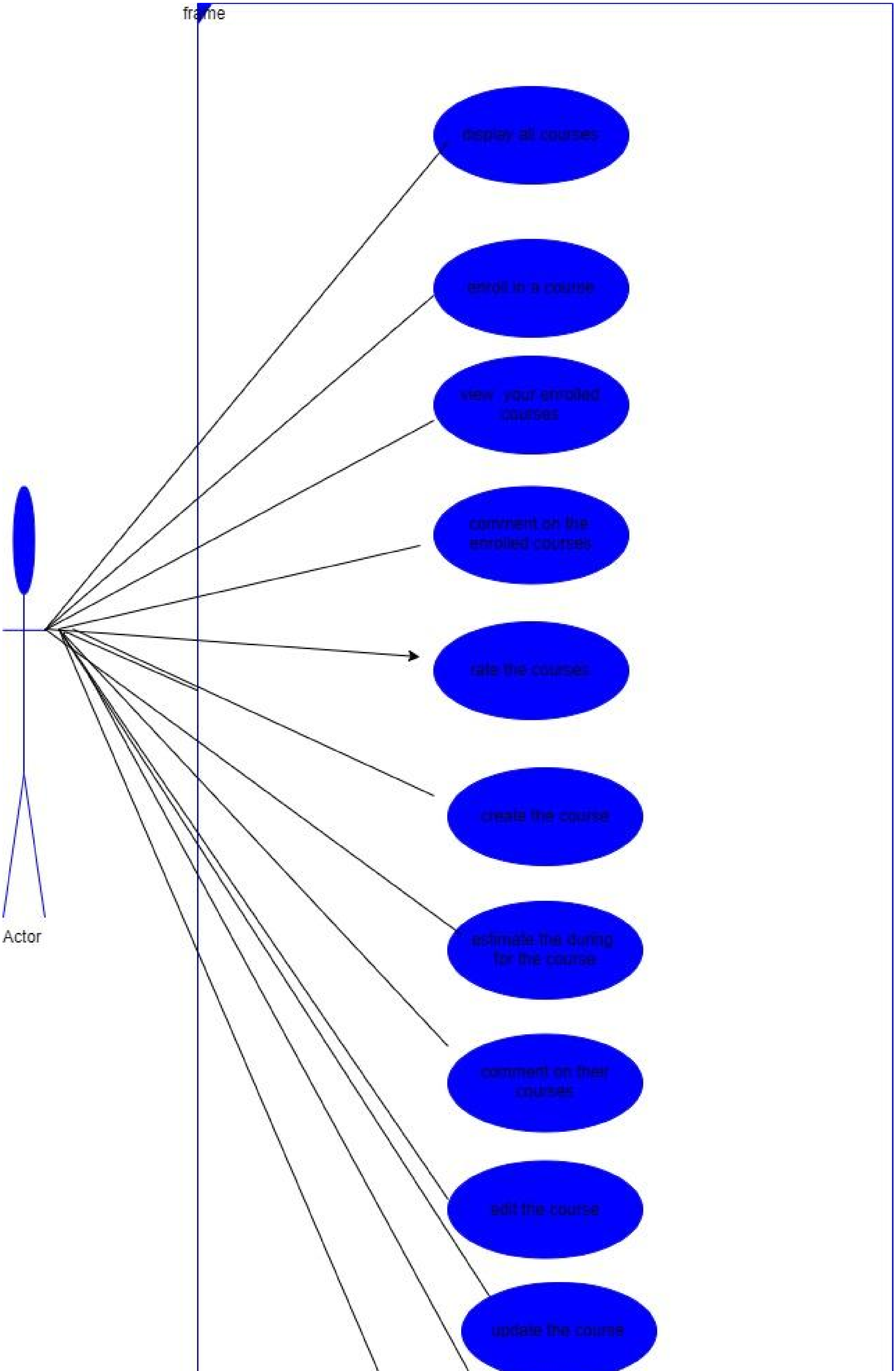


2.4 Physical View: is concerned with the physical distribution of the system, as well as the physical connections between distributed parts
Depicts the system from a system engineer’s point of view
Here we represent it in UML using deployment diagrams

2.4.1 Development Diagram: model the physical deployment of artifacts on nodes



Use Case Diagram: A use case diagram is **used to represent the dynamic behaviour of a system**. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application.


frame
display all courses
enroll in a course
view your enrolled courses
comment on the enrolled courses
rate the courses
create the course
estimate the during for the course
comment on their courses
edit the course
update the course
Actor