AUTOMATED ATTENDANCE MANAGEMENT SYSTEM

Exercise No:6 Package and component Diagram

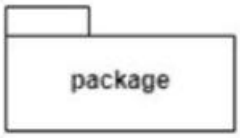
Team Members:
Sujay Sathya (174)
Srinath S (205)
Utkarsh Uniyal (185)

Aim:

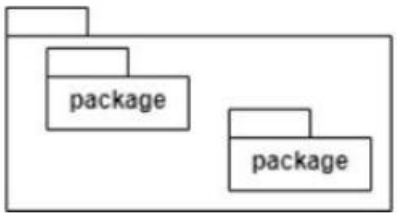
To create a UML package diagram and Uml component diagram for the automated attendance management system.

UML Notations for package diagram:

Packages : a package is used to group elements and provides a namespace for the grouped elements.

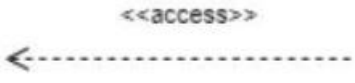


Sub-system : a collection of packages and their relations.

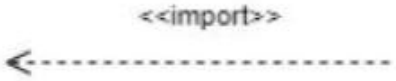


Dependency : signifies that a model requires another model to be implemented.

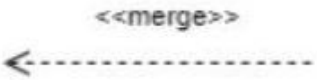
Access: Importing by a packageable element



Import: Importing by packages

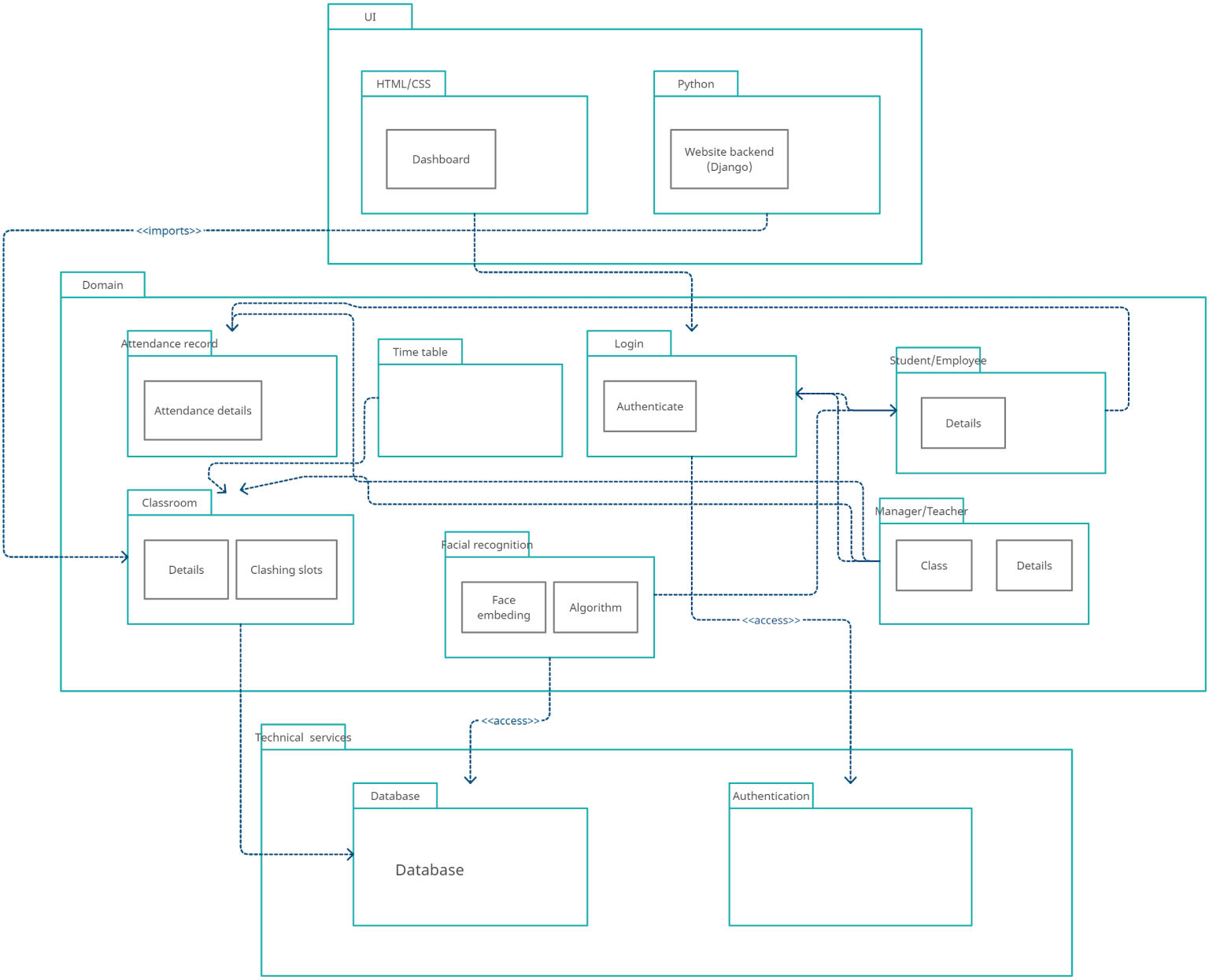


Merge: Contents from two packages are combined.



Generalization: the specific classifier inherits the features of the more general classifier.

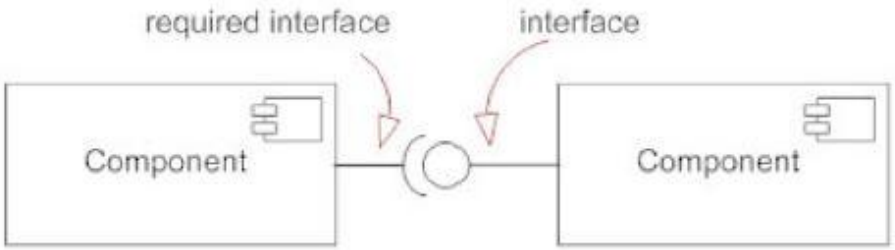
Package Diagram:


UI
HTML/CSS
Dashboard
Python
Website backend (Django)
<<imports>>
Domain
Attendance record
Attendance details
Time table
Login
Authenticate
Student/Employee
Details
Classroom
Details
Clashing slots
Facial recognition
Face embeding
Algorithm
Manager/Teacher
Class
Details
<<access>>
<<access>>
Technical services
Database
Database
Authentication

UML Notations for component diagram:

Component: A component is a physical, replaceable part of a system that conforms to, and provides the realization of, a set of interfaces.

Interface: An interface (small circle or semicircle on a stick) describes a group of operations used (required) or created (provided) by components.
A full circle represents an interface created or provided by the component. A semi-circle represents a required interface, like a person's input.



Dependency Symbol/Association:

Shows that one part of your system depends on another. Dependencies are represented by dashed lines linking one component (or element) to another.

An association specifies a semantic relationship that can occur between typed instances.It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.

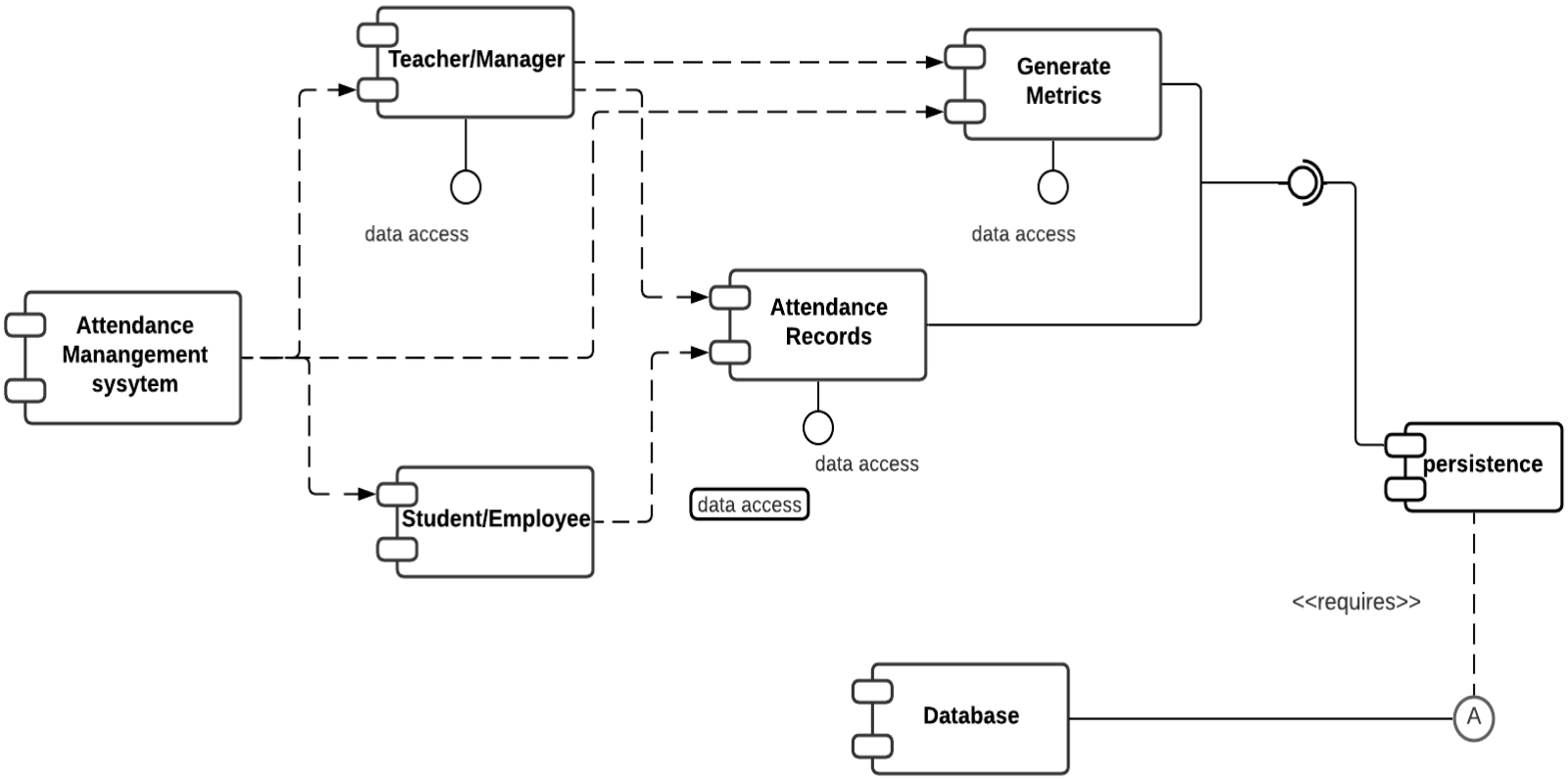
Components used in the class diagram:

1. Automated Attendance management system
2. Registered User
    a. Teacher/Manager
    b. Student/Employee
3. Attendance Records
4. Generate metrics
5. Persistence
6. Database

Artifacts Used:

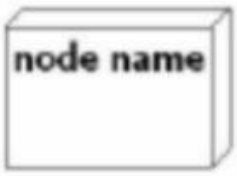
1. MongoDB
2. Data access

Component Diagram:



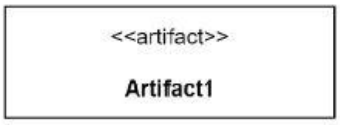
UML Notations for deployment diagram:

Node: Nodes represent either hardware devices or software execution environments. They could be connected through communication paths to create network systems of arbitrary complexity.

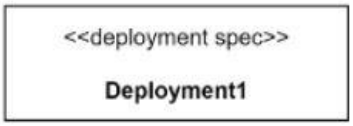


Communication Path: A communication path is association between two deployment targets, through which they are able to exchange signals and messages.Communication path is notated as association, and it has no additional notation compared to association.

Artifacts: An artifact is a classifier that represents some physical entity, a piece of information that is used or is produced by a software development process, or by deployment and operation of a system. Artifact is a source of a deployment to a node.



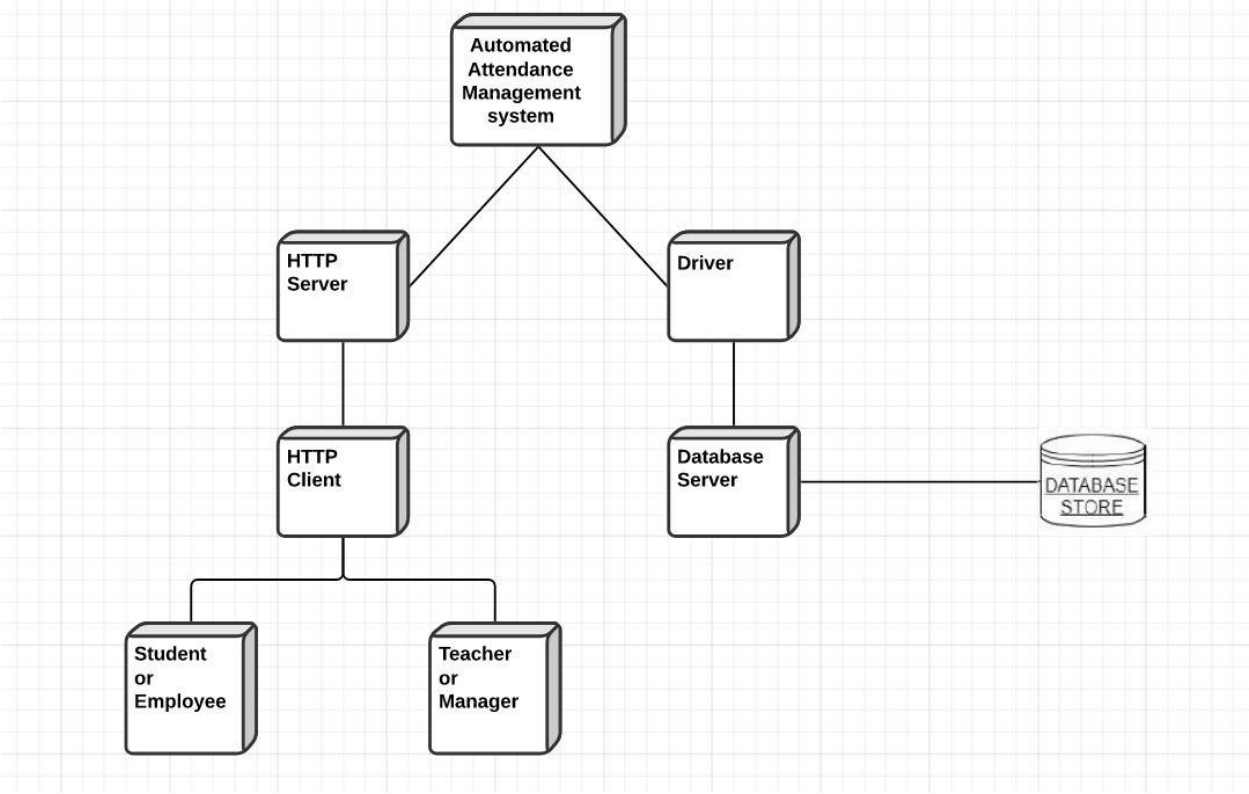
Deployment Specifications: A deployment specification is an artifact that specifies a set of deployment properties that determine execution parameters of a component artifact that is deployed on a node. A deployment specification can be aimed at a specific type of container for components.



List of nodes identified from component diagram:

1. Automated Attendance Management system
2. HTTP Server and HTTP Client
3. Client:
    a. Teacher/Manager
    b. Student/Employee
4. Driver
5. Database Server
6. Database store

Deployment Diagram:



Documentation:

From this exercise we learn how to efficiently categorize which component can access what data making it easier during implementation. The deployment diagram also provides a clear flow of the processes that take place within the **Automated Attendance Management System.**