Architecture Document

SWEN90007

LMS

Team: SDA Wombat

In charge of: Haobei Ma, Yu Guo, Yifan He

THE UNIVERSITY OF MELBOURNE

SCHOOL OF COMPUTING & INFORMATION SYSTEMS

Revision History

Date	Version	Description	Author
10/08/2020	01.00-D1	Initial draft	All team members
11/08/2020	01.00-D2	Added use cases 1-4	Haobei Ma
13/08/2020	01.00-D3	Added use cases 5-9	Yifan He
13/08/2020	01.00-D4	Added use cases 10-11	Yu Guo
14/08/2020	01.00-D5	Added user case diagram	Yu Guo
14/08/2020	01.00-D6	Updated user case diagram	Yifan He
15/08/2020	01.00-D7	Rechecked user cases and added use cases 12-13	Haobei Ma
22/08/2020	01.00-D8	Review the document	All team members
22/08/2020	01.00	First version of the document	All team members
11/09/2020	0.2.00-D1	Updated document based on feedback	Haobei Ma
14/09/2020	02.00-D2	Added Logical View	Haobei Ma
17/09/2020	02.00-D3	Added Process View	Haobei Ma
19/09/2020	02.00-D4	Added Development View	Yifan He
25/09/2020	02.00-D5	Added Physical View	Yu Guo
01/10/2020	02.00-D6	Added Scenarios	All team members
03/10/2020	02.00	Second version of the document	All team members

Contents

- ■ INTRODUCTION 3
- ■ ARCHITECTURAL REPRESENTATION 4
- ■ ARCHITECTURAL OBJECTIVES AND RESTRICTIONS 5
- ■ LOGICAL VIEW 6
- ■ PROCESS VIEW 8
- ■ DEVELOPMENT VIEW 12
- ■ PHYSICAL VIEW 16
- ■ SCENARIOS 19

■ Introduction

This document specifies the system's architecture LMS, describing its main standards, module, components, *frameworks* and integrations.

1.1 Proposal

The purpose of this document is to give, in high level overview, a technical solution to be followed, emphasizing the components and *frameworks* that will be reused and researched, as well as the interfaces and integration of them.

1.2 Target Users

This document is aimed at the project team, with a consolidated reference to the research and evolution of the system with the main focus on technical solutions to be followed.

■.1 Conventions, terms and abbreviations

This section explains the concept of some important terms that will be used throughout this document. These terms are detailed alphabetically in the following table.

Term	Description
Component	Reusable and independent software element with well defined public interface, which encapsulates numerous functionalities and which can be easily integrated with other components.
Module	Logical grouping of functionalities to facilitate the division and understanding of software.
MCQ	Multiple Choice Questions. Questions which have a correct answer from choices offered as a list.

■ Architectural representation

The specification of the system's architecture <System's name> follows the *framework* "4+1" [1], which defines a set of views, as shown in Figure 1. Each of these views approaches aspects of architectural relevance under different perspectives:

- The **logical view** shows the significant elements of the project for the adopted architecture and the relationship between them. Between the main elements are modules, components, packages and the application main classes;
- The **process view** shows the concurrency and synchronization aspects of the system, mapping the elements of the logical view to processes, *threads* and execution tasks;
- The **development view** focuses on aspects relating to the organization of the system's source code, architectural patterns used and orientations and the norms for the system's development;
- The **physical view** shows the hardware involved and the mapping of the software elements to the hardware elements in the system's environment.
- The **scenarios** show a subset of the architecturally significant use cases of the system.

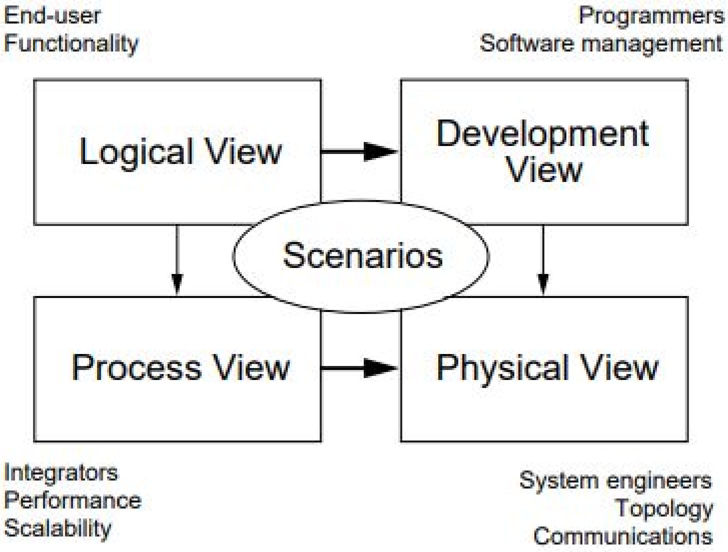


Figure 1. Views of *framework* "4+1"

source: Kruchten, P. B. (1995). The 4+ 1 view model of architecture. *IEEE software*, *12*(6), 42-50..

■ Architectural Objectives and Restrictions

The defined architecture's main objective is to make the system scalable, reliable, have high data integrity and security. As a system that is designed to be an online examination application, data integrity is of utmost importance. Privacy would be another big concern, and students will be more likely to use a system that is secure. The system has to be reliable to ensure that the students will not receive an error during the middle of their exams. The system, being in its initial stages, will have to be scalable to ensure longevity of the system and allow improvements to be made to include more features.

■ Logical View

This section shows the system's organization from a functional point of view. The main elements, like modules and main components are specified. The interface between these elements is also specified. The figures below illustrate the logical architecture of the system.

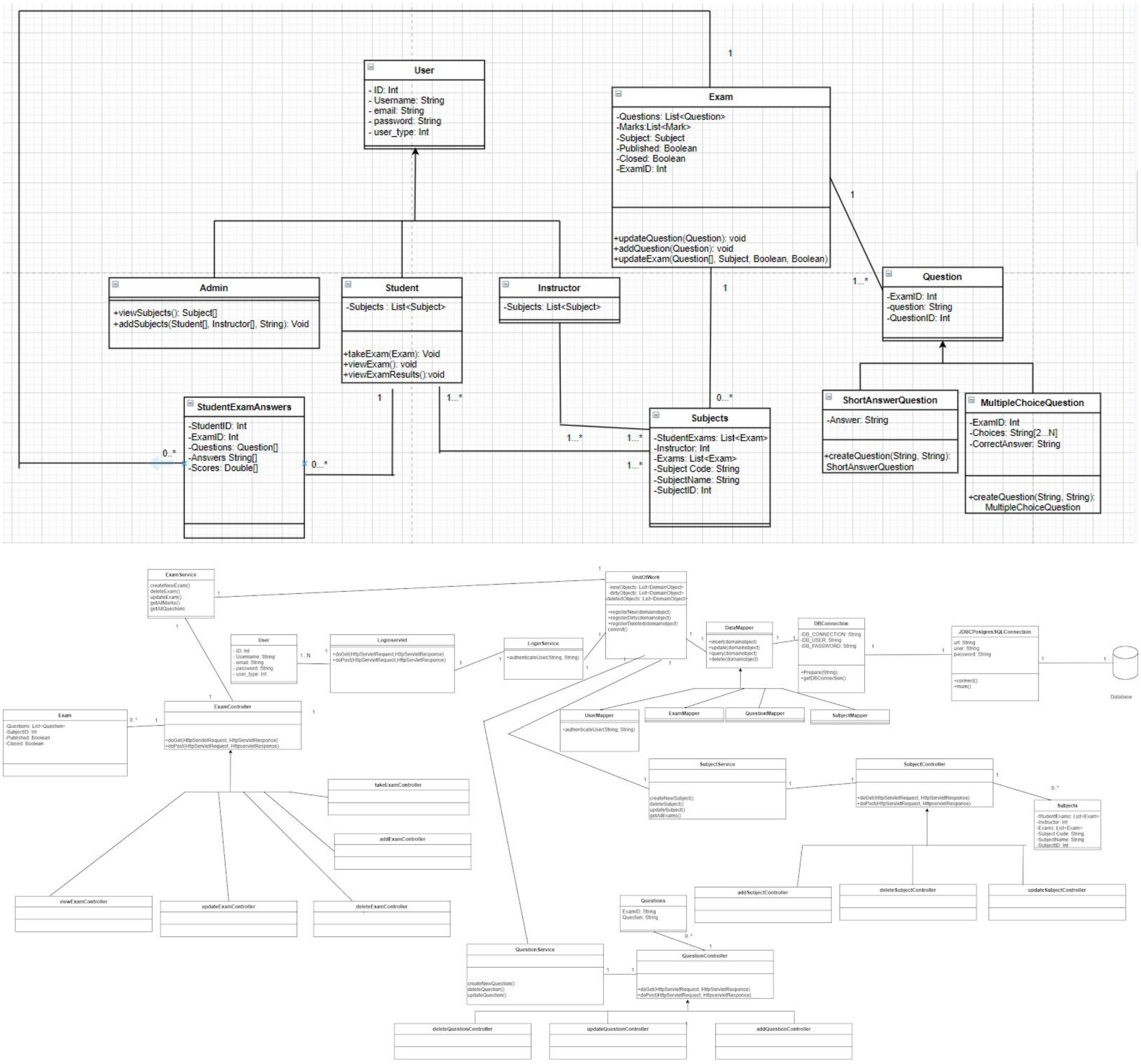


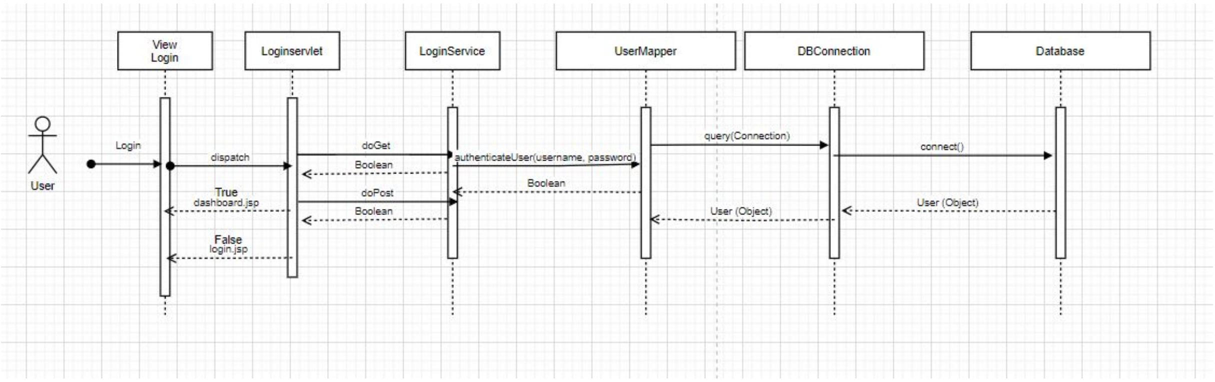
Figure. General view of the architecture

Note: The two pictures combine together to form a single class diagram.

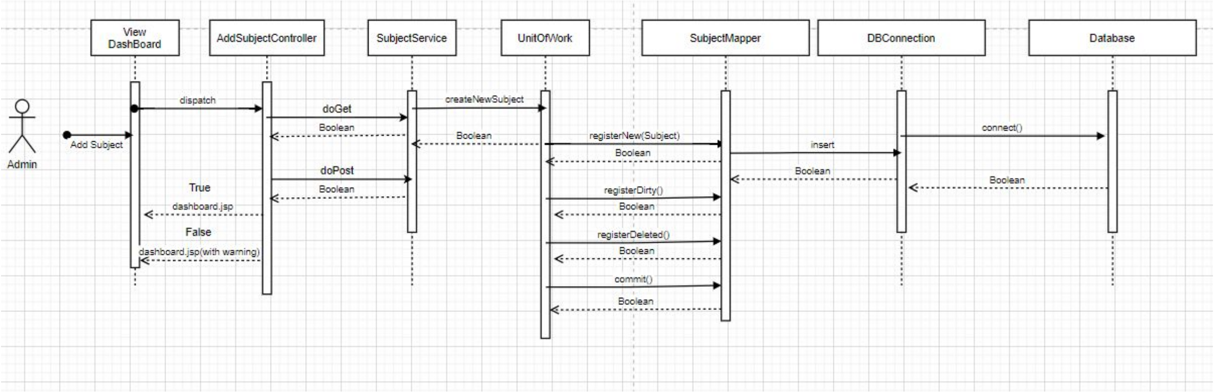
https://drive.google.com/file/d/1Ujrq8OLcP9jqEt7V415nlCcF9A4AiVnY/view?fbclid=IwAR0u-FFYdza6QN2v7JQ807c5a-I0KzpPSO4ajQfscVszyYSQ6Eqn7vlLhYM

■ Process View

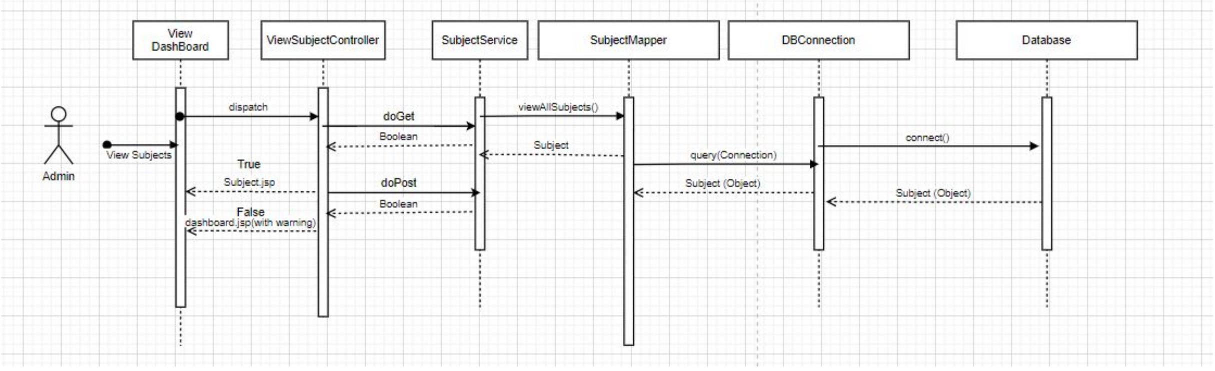
The section shows the mapping of the logical architecture elements to the processes and threads of the system execution. The figures below illustrate this mapping.



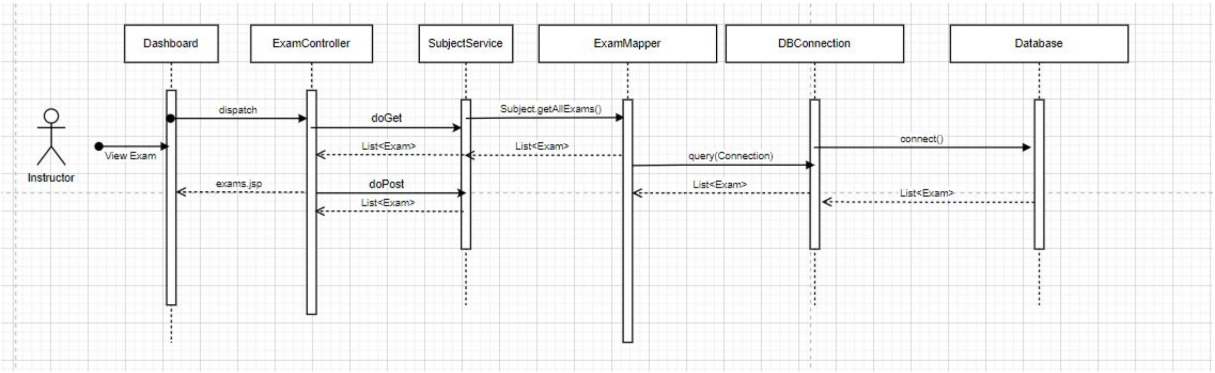
User Logins



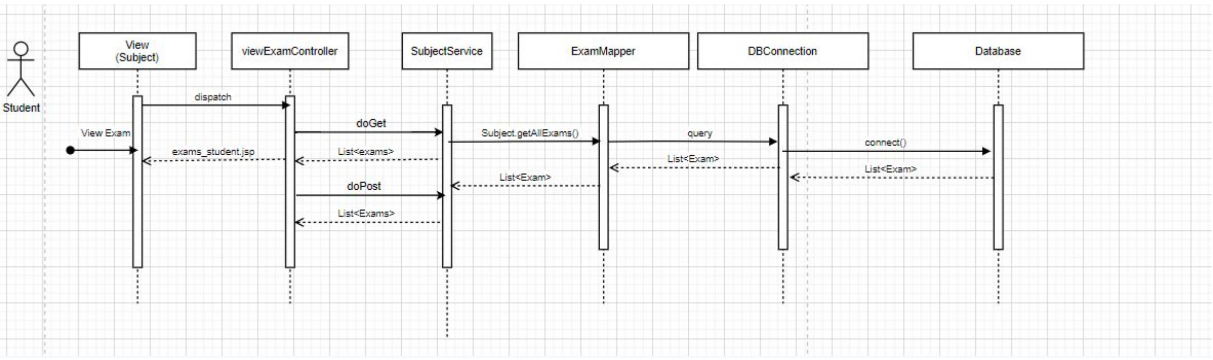
Admin adds subjects



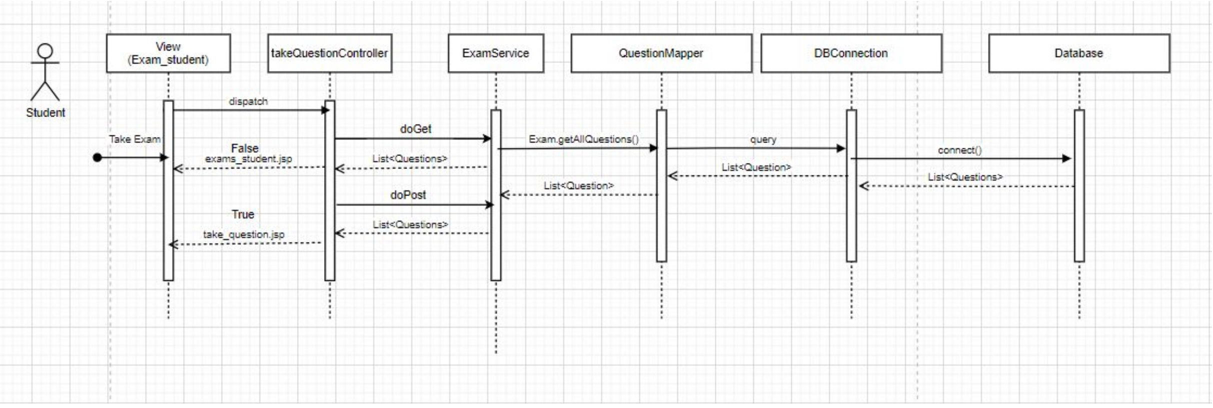
Admin views subjects



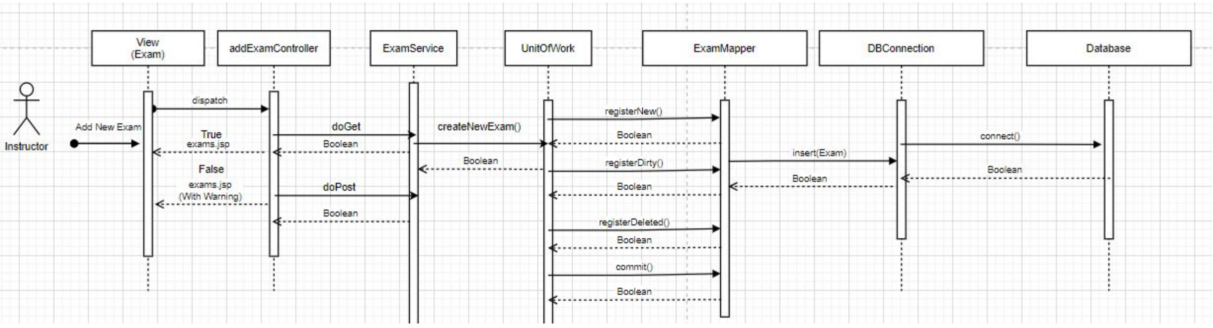
Instructor views exams



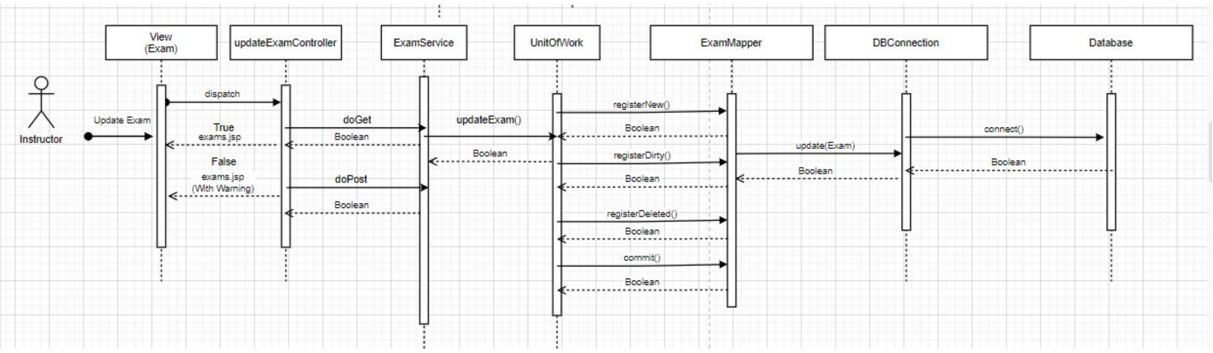
Student views exams



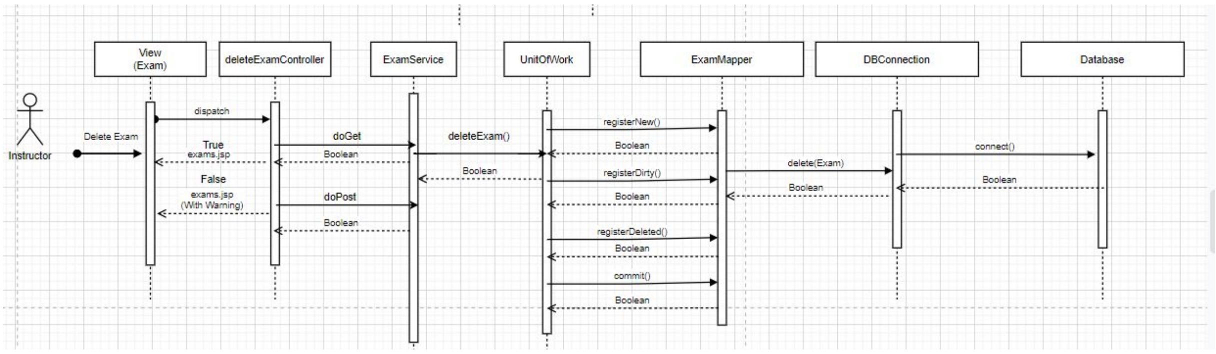
Student takes exams



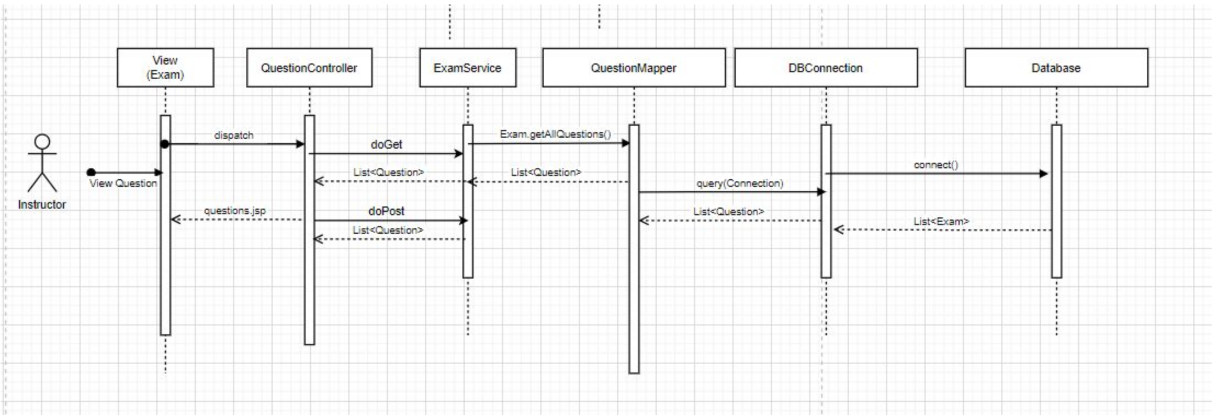
Instructor adds new exams



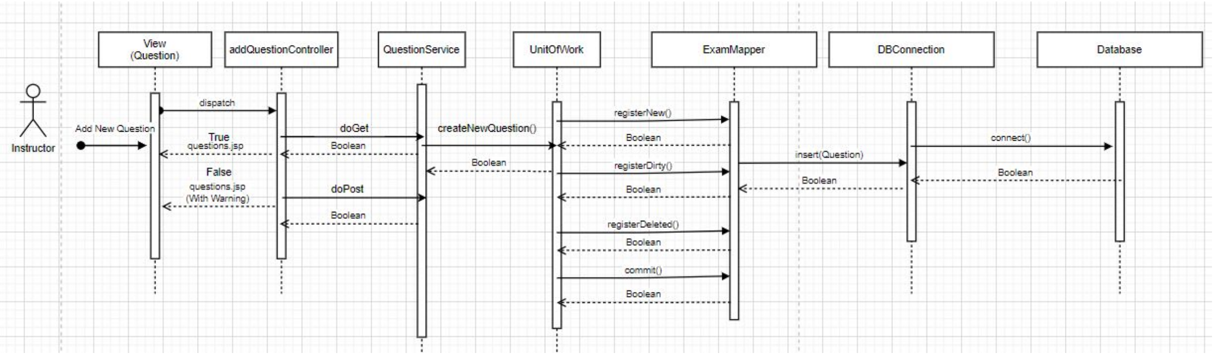
Instructor updates exam



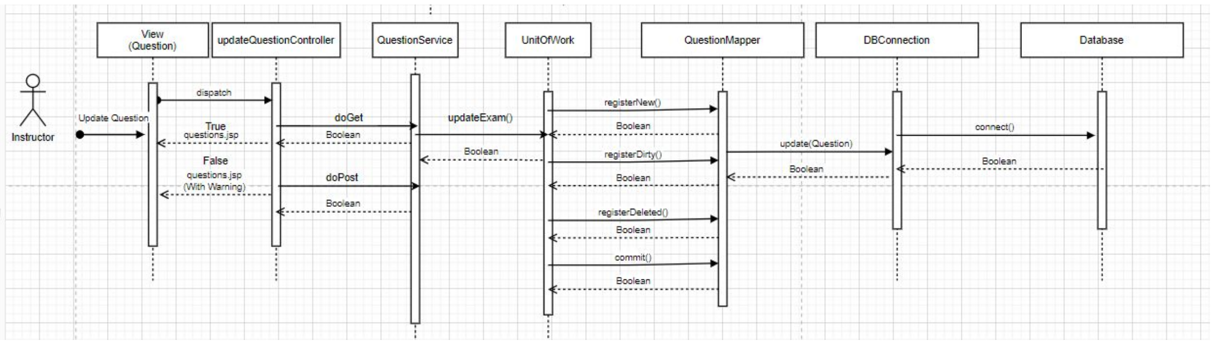
Instructor deletes exam



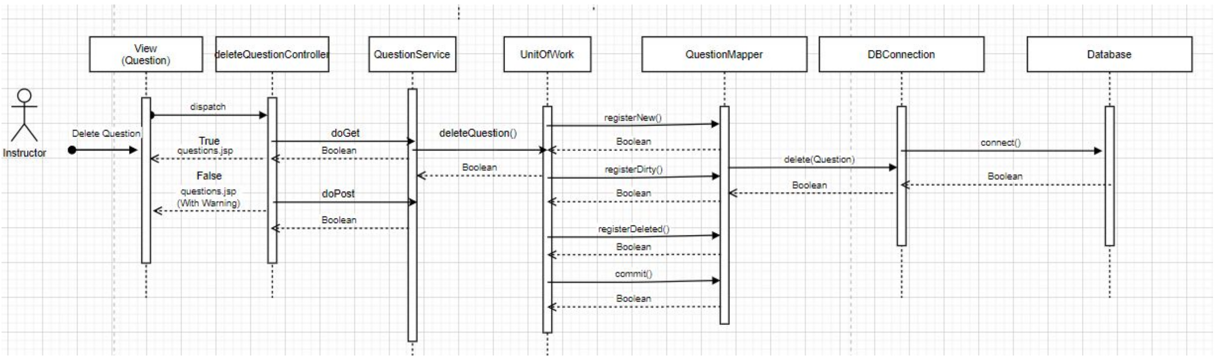
Instructor views questions



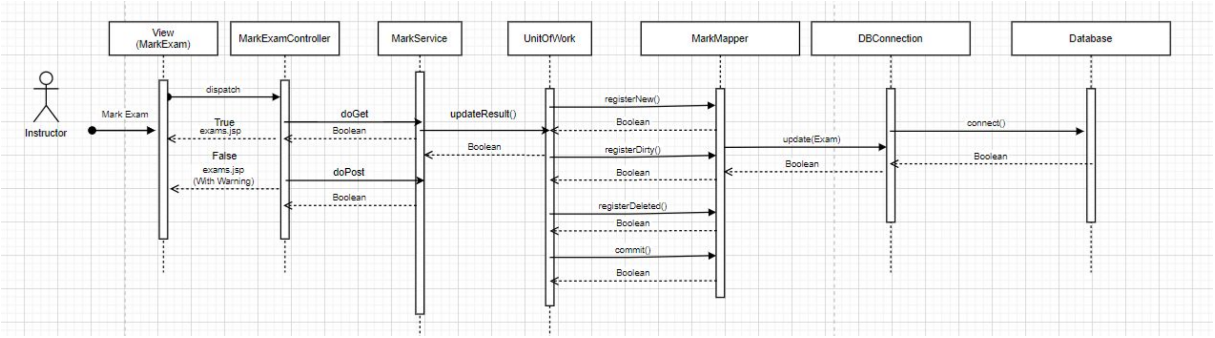
Instructor adds new questions



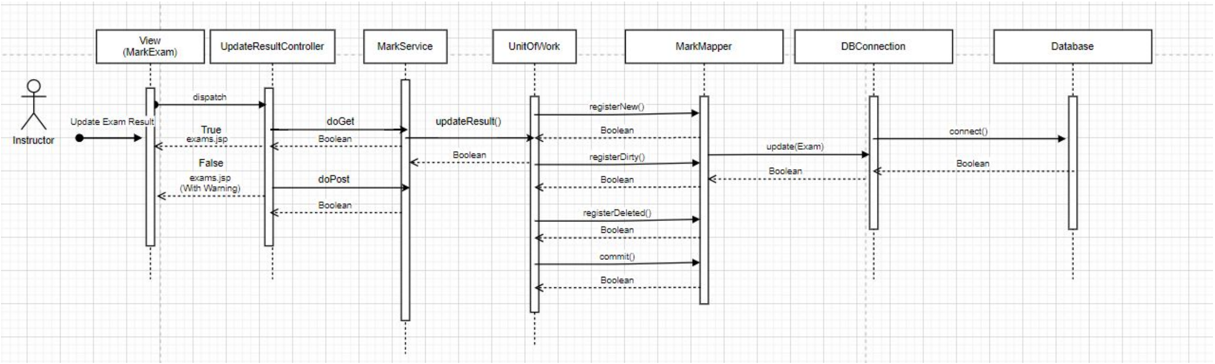
Instructor updates questions



Instructor deletes questions



Instructor marks exam



Instructor updates exam marks

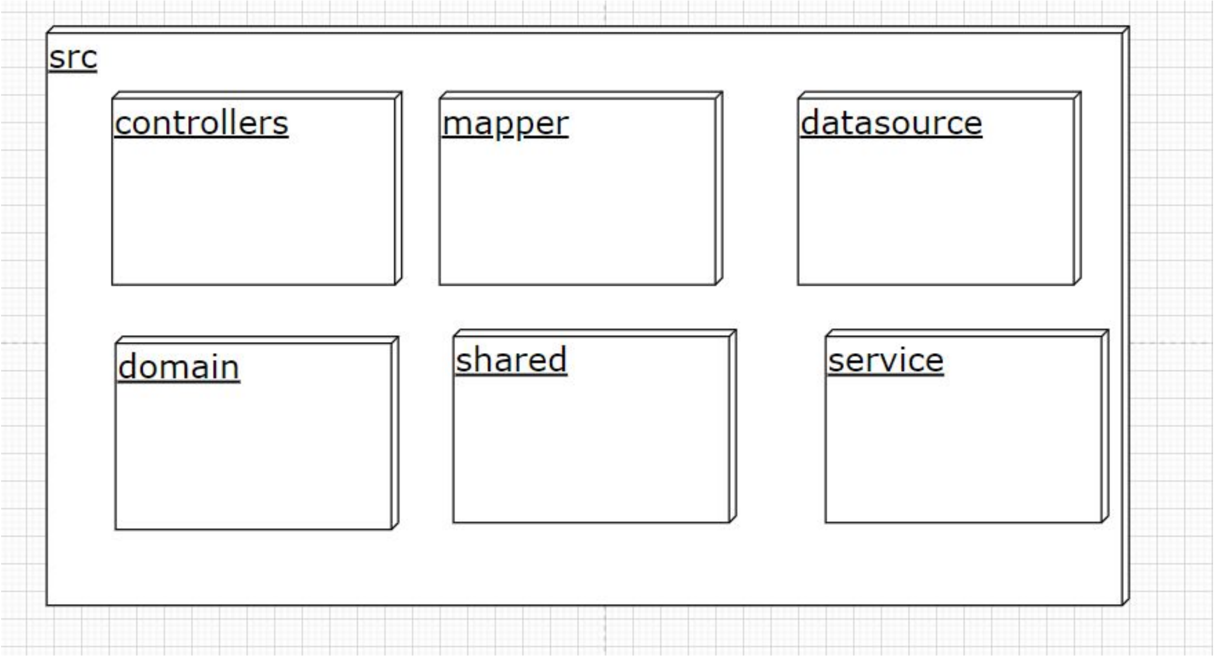
Figures: System execution diagram

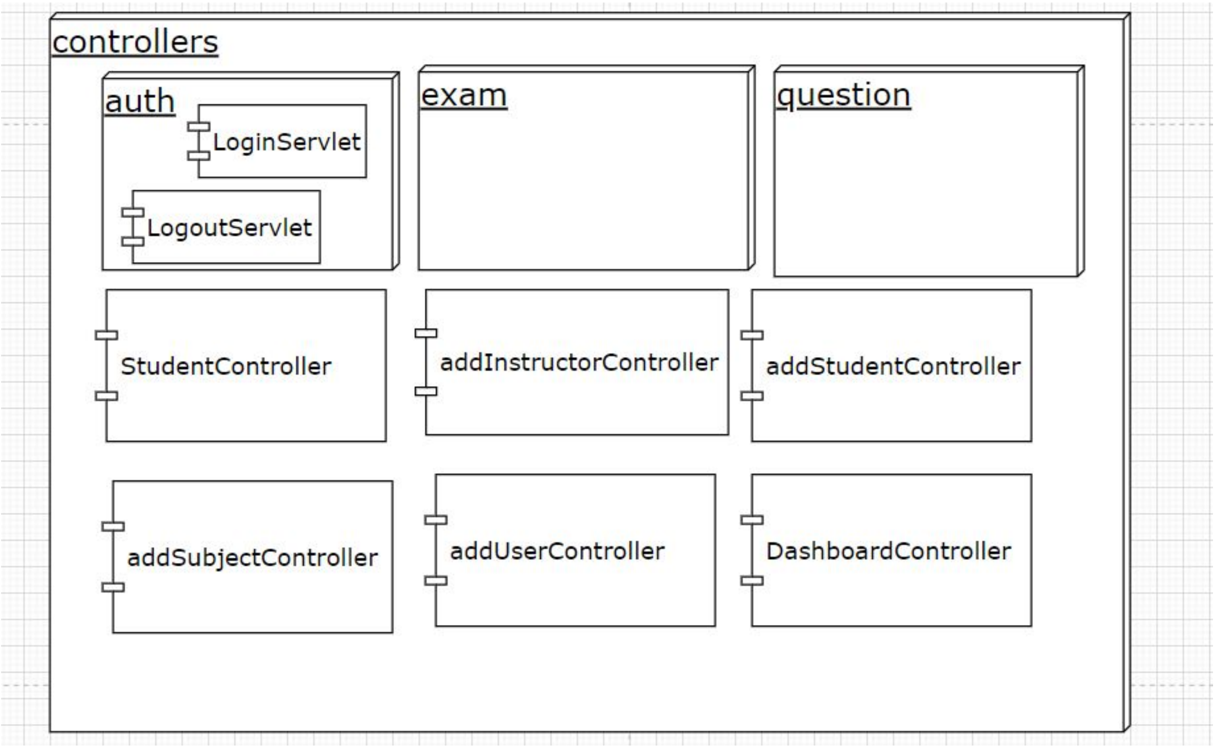
Note: Dashboard.jsp contains the list of subjects, hence there were no sequence diagrams for viewing subjects, since all users will automatically see the list of subjects upon logging in.

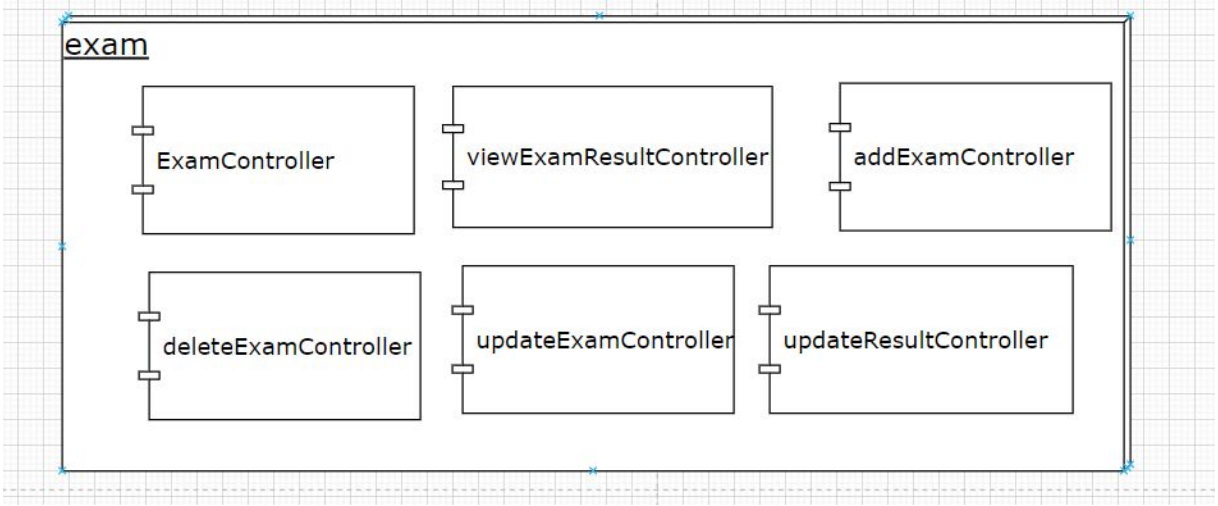
Students will be able to view their marks in the list view of exams. Hence, no sequence diagram for viewing marks is implemented.

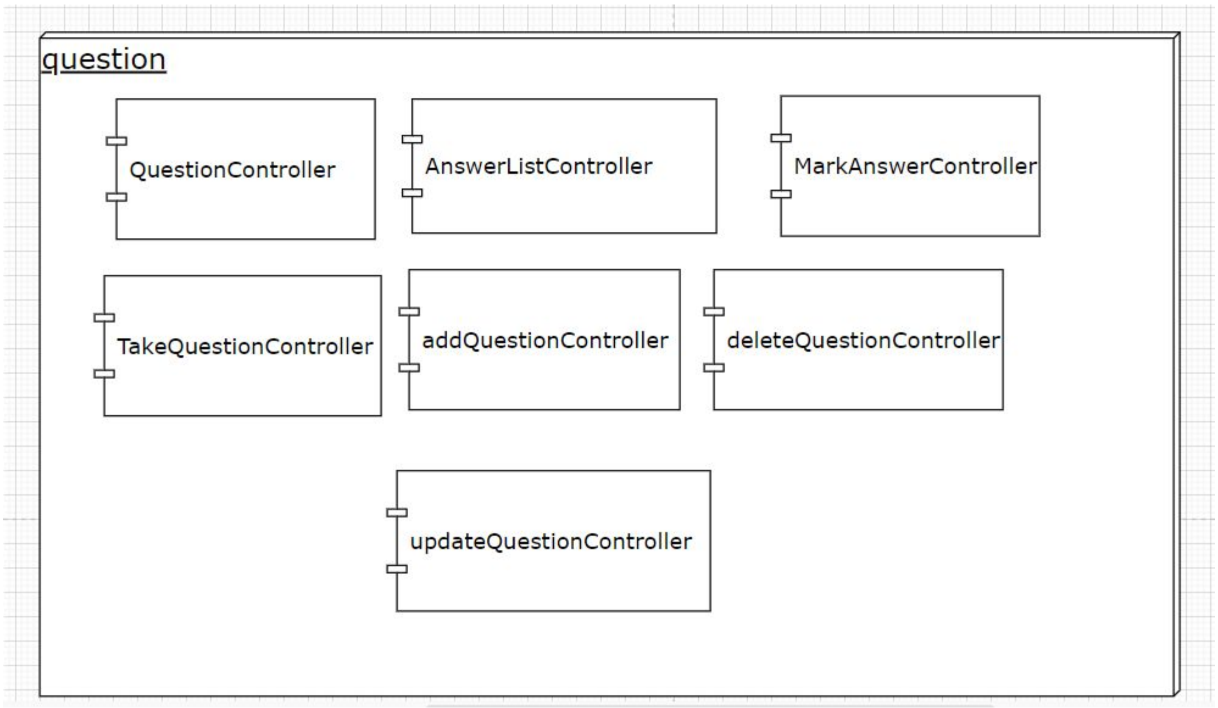
■ Development View

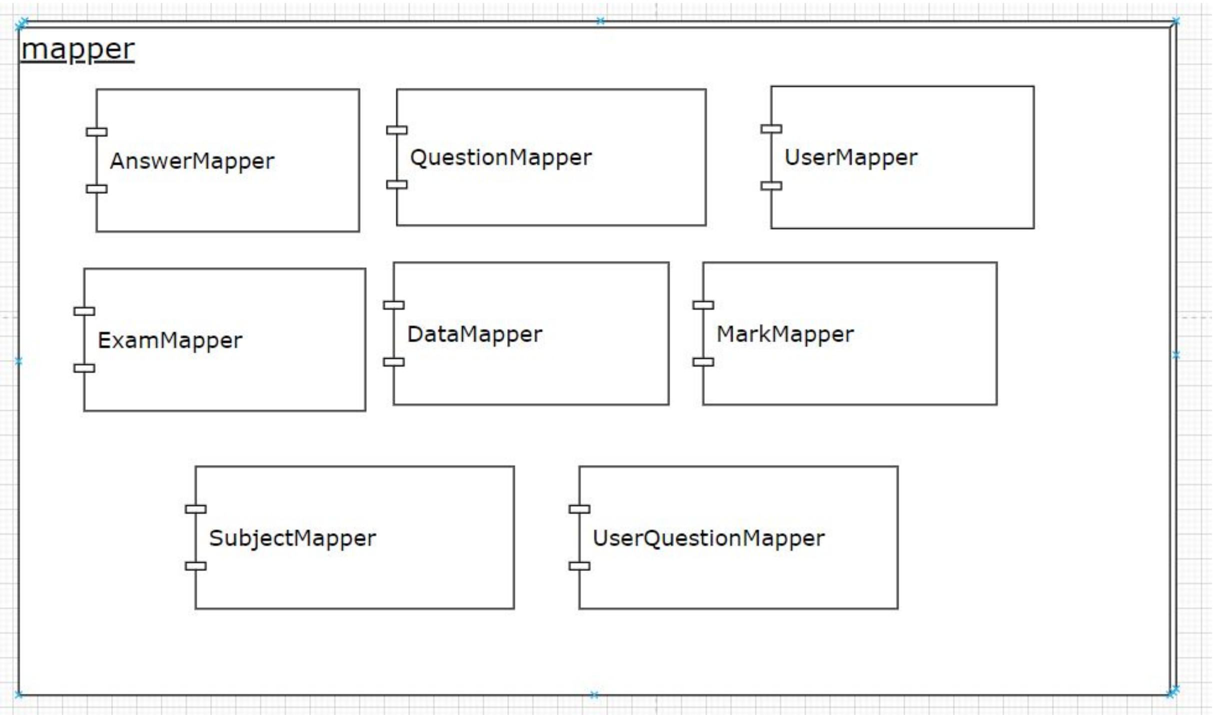
This section provides orientations to the project and system implementation in accordance with the established architecture. Figure 3 illustrates the implementation view of system architecture.

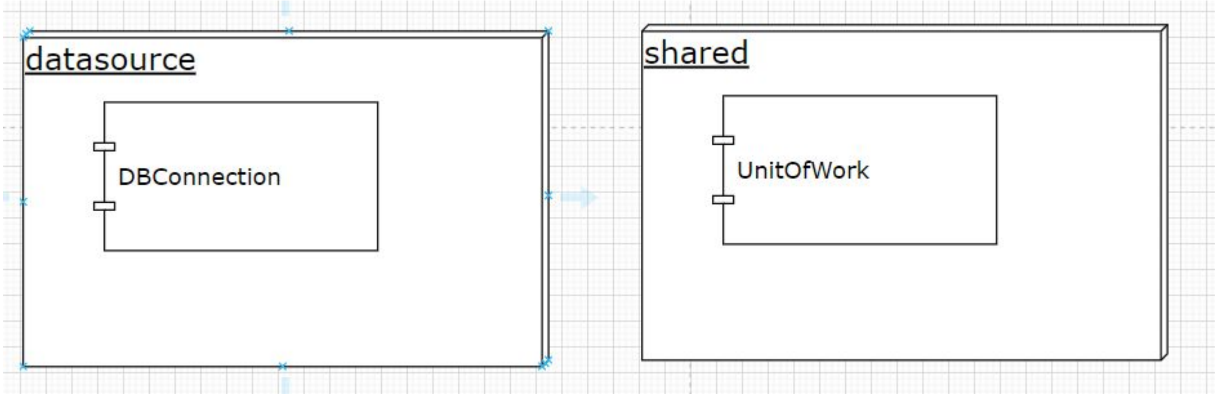


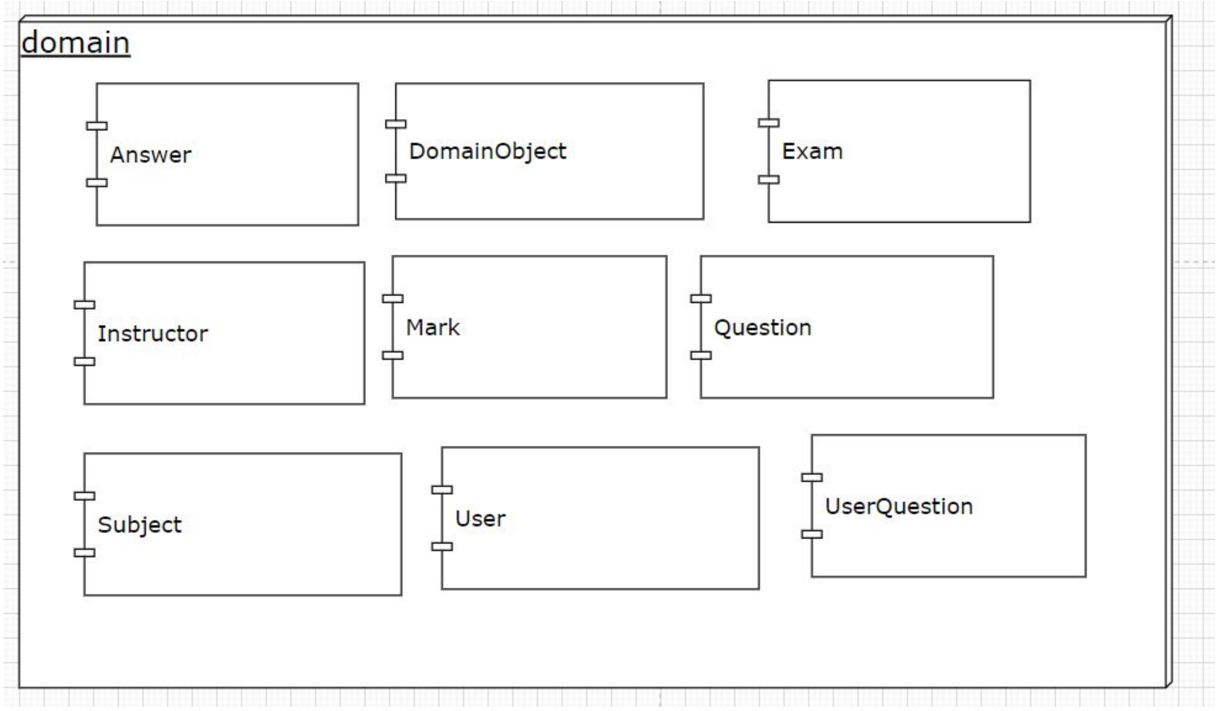

controllers
auth
LoginServlet
LogoutServlet
exam
question
StudentController
addInstructorController
addStudentController
addSubjectController
addUserController
DashboardController

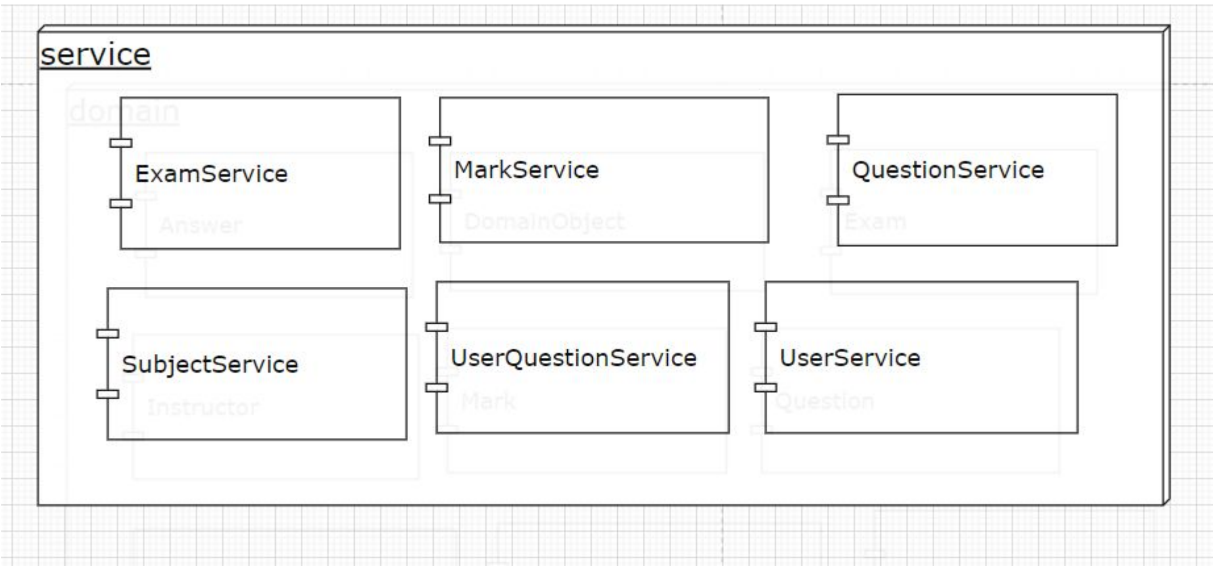

exam
ExamController
viewExamResultController
addExamController
deleteExamController
updateExamController
updateResultController

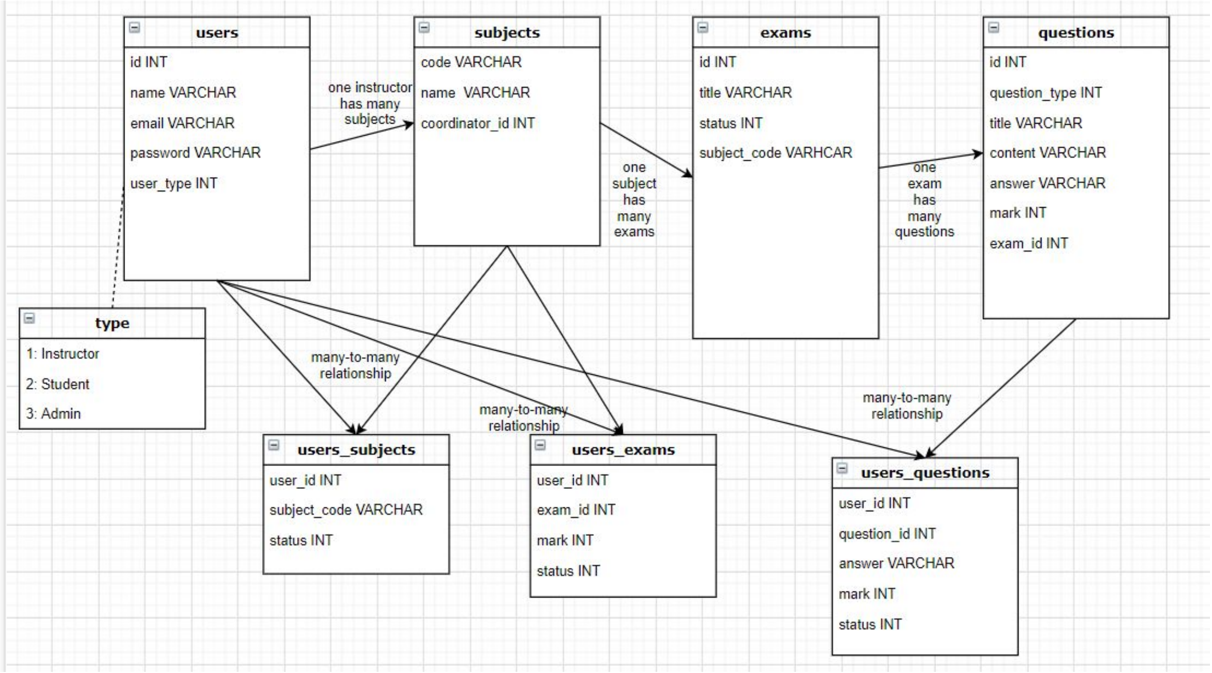

question
QuestionController
AnswerListController
MarkAnswerController
TakeQuestionController
addQuestionController
deleteQuestionController
updateQuestionController


mapper
AnswerMapper
QuestionMapper
UserMapper
ExamMapper
DataMapper
MarkMapper
SubjectMapper
UserQuestionMapper


datasource
DBConnection
shared
UnitOfWork


domain
Answer
DomainObject
Exam
Instructor
Mark
Question
Subject
User
UserQuestion

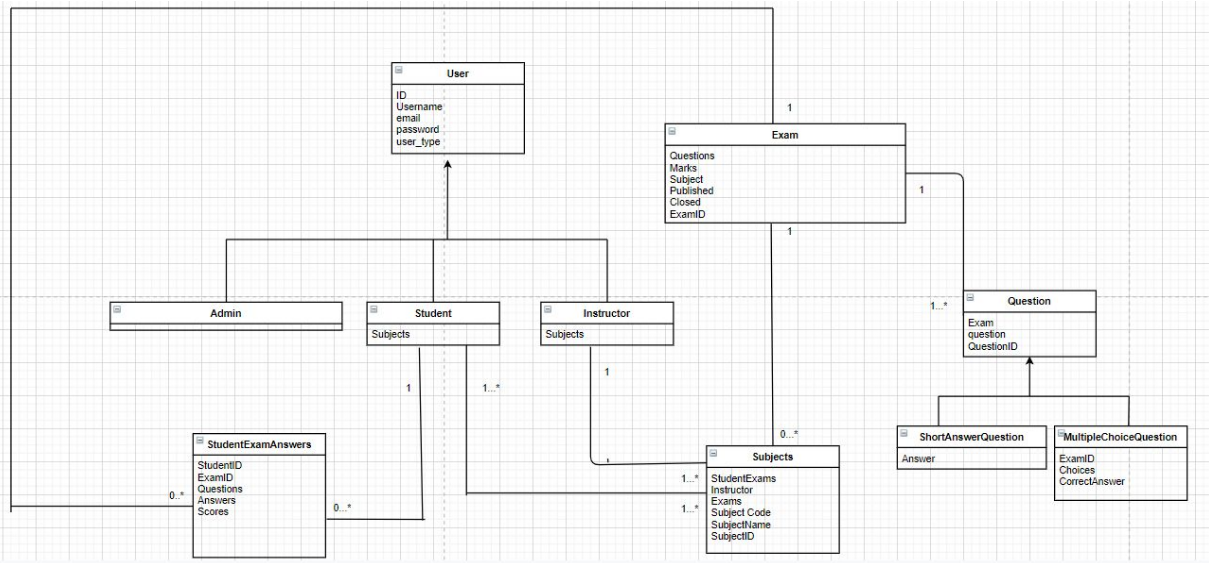

service
ExamService
MarkService
QuestionService
SubjectService
UserQuestionService
UserService

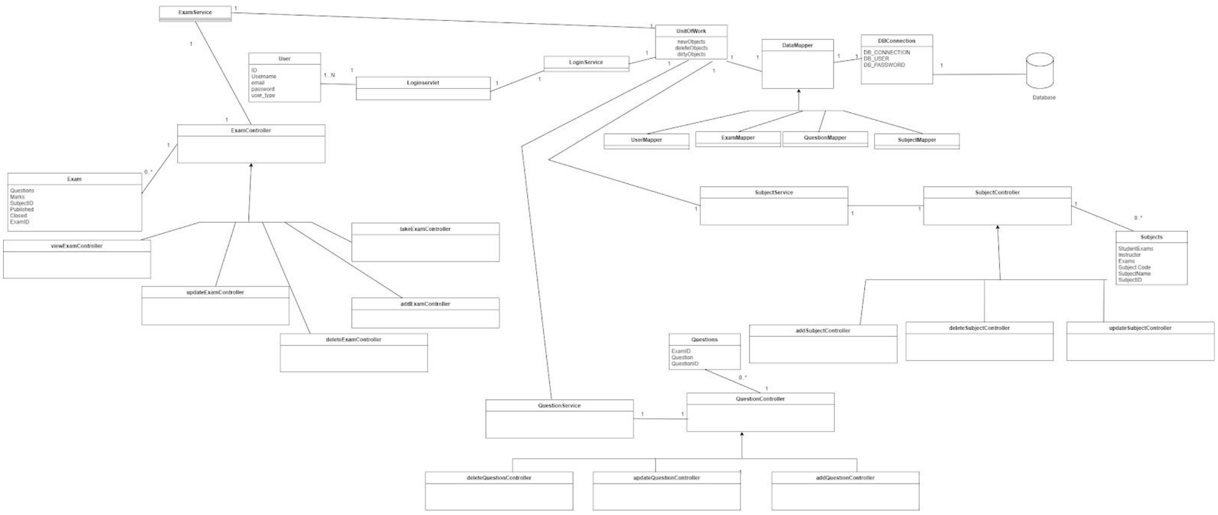


Figures. Development view of system architecture

1.3 Architectural Patterns

The figures below represent the domain model. It is split up into two pictures.





https://drive.google.com/file/d/1NIJPiCkKiheYf4sj13I5D3z7sgKSLyhc/view?fbclid=IwAR01p4FRQ0fxGRW-TKsgjgK7EVyo2qJWOEpm4kFDLpyPxsjRjxv9vfuOVhI

Pattern	Reason	Where it will be used
Domain Model	It is easily extensible, which allows for improvements for the system to be easily made in the future. Transaction scripts is another pattern that can be used to replace domain models, but it is rigid and should not be used on a system that is subject to change.	Domain model is used to show the relationships between all classes.
Data mapper	Data mapper is highly compatible with the domain model, which is being used in this project. Additionally, it allows for loose coupling within the code by decoupling the database access from the domain subjects. Row data gateway is not used due to the high likelihood of the system being changed in the future, which can lead to database schema changing.	all queries, deletes and updates within the database will be made through mappers. The following list of mappers are used: AnswerMapper DataMapper (parent class) ExamMapper MarkMapper QuestionMapper SubjectMapper UserMapper UserQuestionMapper

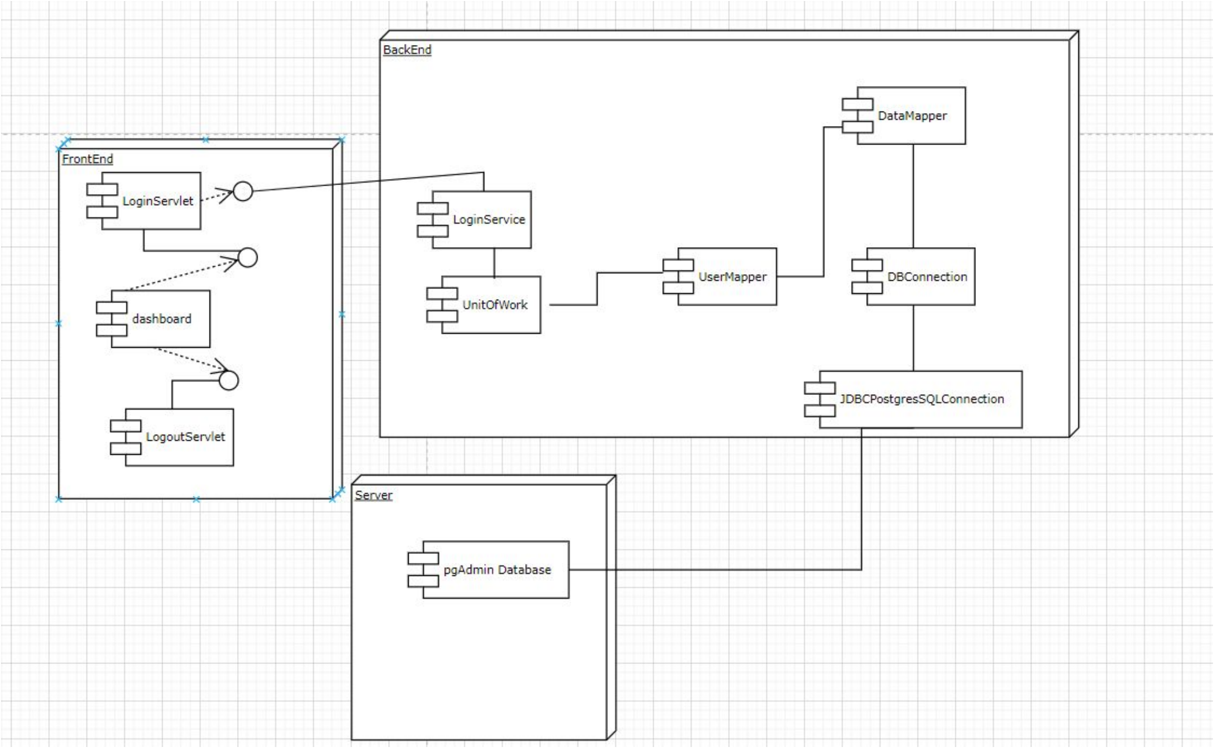
Unit of Work	Unit of work is simple, efficient and cohesive. there will only be minimal changes at a time, which allow for faster commits to the database. It is easily implementable and all information will be contained in a single place.	All database queries will go through the unit of work class, with every update, delete, add, record before applying them to the database.
Lazy Load	Lazy load can help improve the efficiency of retrieving objects from the database.	When retrieving any data from the database, the object will initially be loaded as null. In questions, answers will be initialized as null. In Exams, questions and marks are initialized as null. In Subject class, the list of exams is initialized as null. In User class, the list of subjects is initialized as null. In Exam class, the lists of questions and marks are initialized as null. When these information are needed, the mapper is called and the values will be retrieved from the database.
Identity Field	Easy and simple to to comprehend, allows for data within the system to be easily tracked	All tables will have an auto generated key to keep track of the data.
Foreign Key Mapping	Allows for tables to be easily connected to other tables within the database for one to many relationships	Every relationship within the database will contain foreign keys of the associated tables. For example, a subject will contain a coordinator id for the associated instructor.
Association Key Mapping	Allows for tables to be easily connected to other tables within the database for many to many relationships	Instructor/Student to subject mapping will use the association key mapping due to the many to many relationship.
Embedded value	Allows the database schema to be neater and more easily understandable.	Instead of the questions being an attribute of the exams table in the database table, There will be another table for Questions.
Concrete Table Inheritance	Concrete table inheritance allows for simplicity and requires no joins or multiple queries. Single table inheritance was not used due to the space complexity of the unneeded attributes that are generated for some classes.	Inheritance is used in multiple places. Our data controllers and mappers will all be inherited from one main parent class. Domain object will also be the parent class of the rest of the objects and users will also have one superclass.

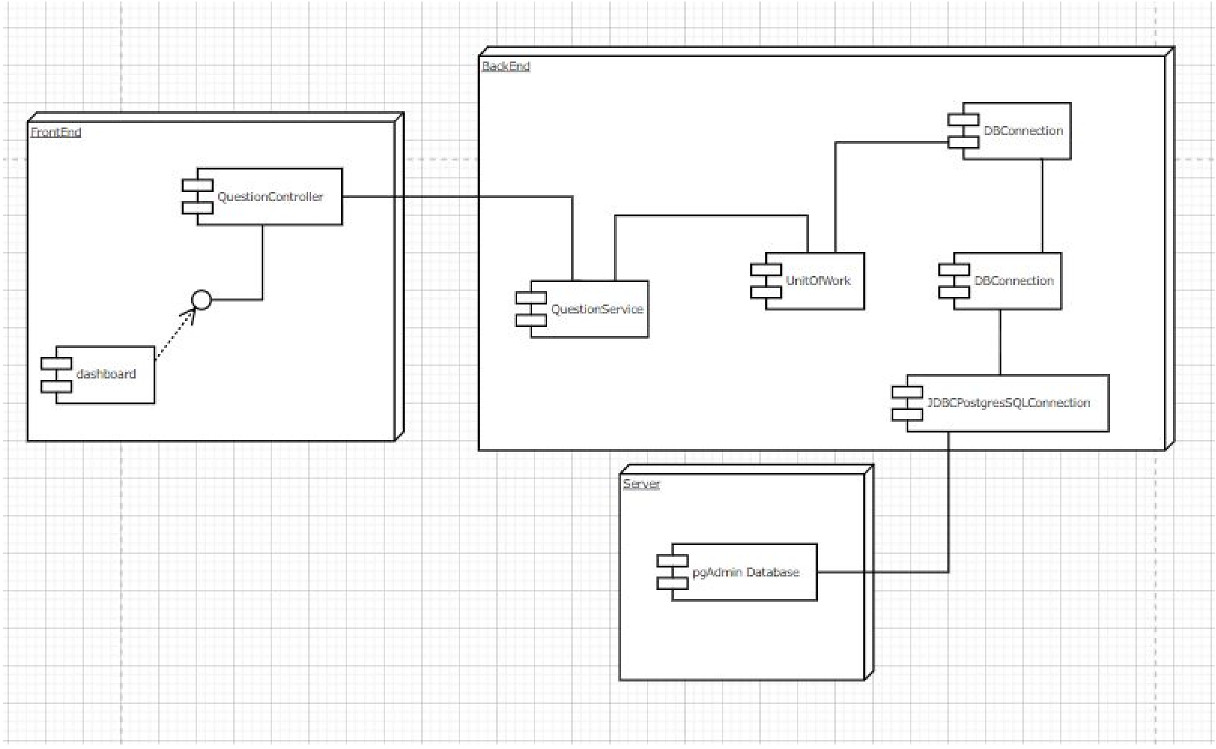
■ Physical View

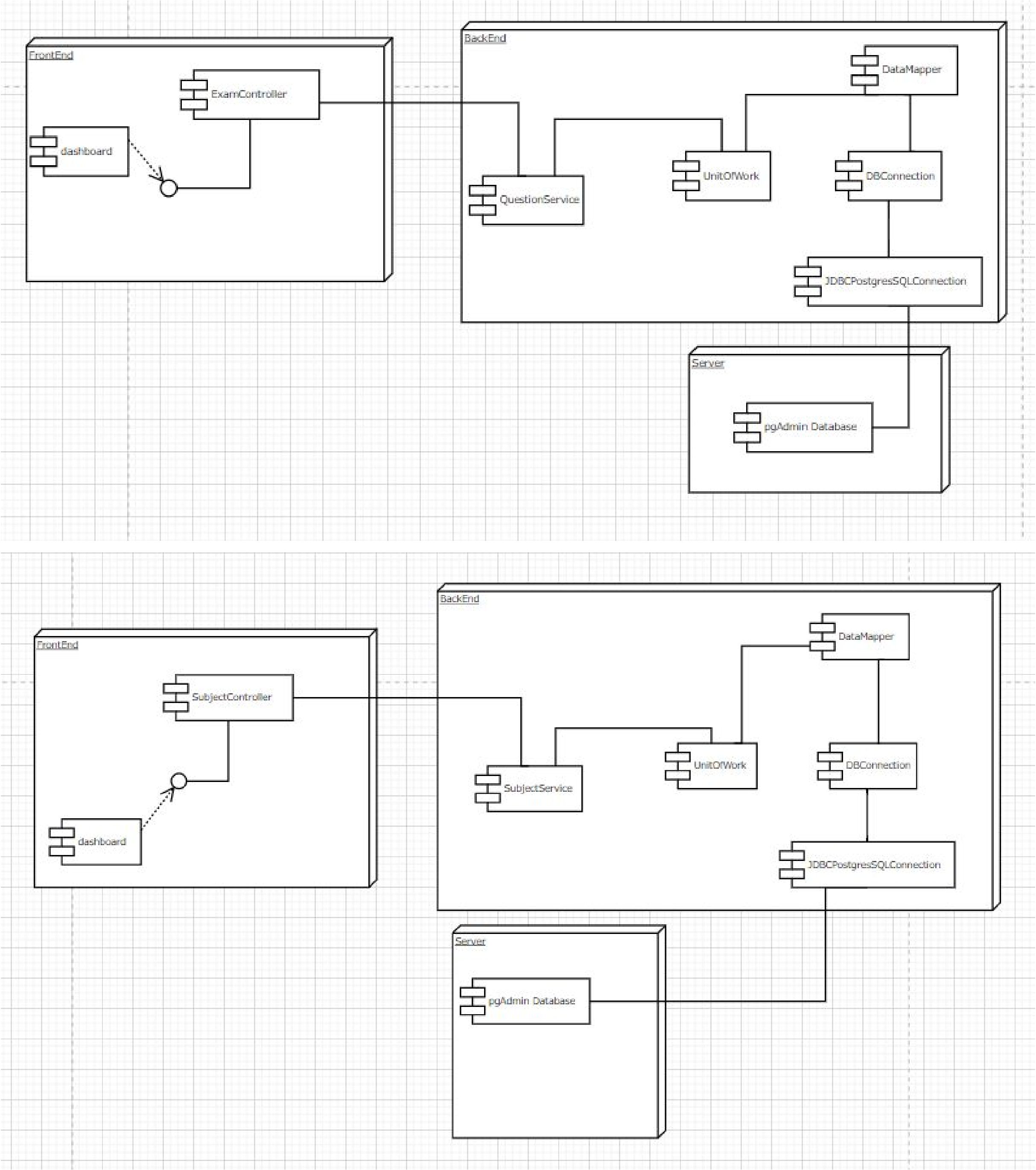
This section describes the hardware elements of the system and the mapping between them and the software elements.

1.4 Production Environment

This section describes the production environment of the system and the mapping between the software elements and the available hardware. The figures below illustrate the physical view of the production environment.




FrontEnd
QuestionController
dashboard
BackEnd
DBConnection
QuestionService
UnitOfWork
DBConnection
JDBCPostgresSQLConnection
Server
pgAdmin Database



Figures. Production environment of the system

Users will be able to interact with only the frontend side of the product. The backend will react to the user’s inputs, and relay instructions to retrieve information from the database.With this system, users can query, update or delete information from the database.

■.1.1 Hardware

The hardware used is heroku cloud.

■.1.2 Software

The software used is Java, JEE for software.

■ Scenarios

Heroku link: https://swen90007-lms.herokuapp.com
The following accounts have been created for the project.

Student

username: Student
password: 123

Instructor

username: Instructor
password: 123

Admin

username: Admin
password: 123

Scenarios:

Student

- Student logs in (Student is shown a list of subjects)
- Student views exams in a subject
- Student attempts an exam for the first time
- Student views the exam result after the instructor has marked it.

Instructor

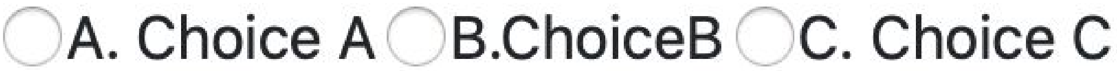
- Instructor logs in (Instructor is shown a list of subjects)
- Instructor views exams in a subject
- Instructor creates a new exam
- Instructor creates a new question*

- Instructor deletes/updates an existing exam
- Instructor deletes/updates an existing question
- Instructor publishes exam
- Instructor marks student answers
- Instructor updates student answers

Admin

- Admin logs in (Admin is shown a list of subjects and users)
- Admin adds a new subject
- Admin adds a new user (For user type: place 2 for student, 3 for instructor)

* For multiple choice questions, place the answer content in the format of: A. Choice A#B.ChoiceB#C. Choice C. This will produce the following result:



○A. Choice A ○B.ChoiceB ○C. Choice C

Note: The database response is slow, please give it some time before refreshing.