



Bilkent University

Department of Computer Engineering

CS319 Term Project

Section 1

Group 1G

ReadyCodeGo

Design Report

Group Members:

- Mehmet Kaan Örnek (21901637)
- Göksu Şenerkek (21803341)
- Cem Apaydın (21802270)
- Ercan Bahri Nazlıoğlu (21903151)
- Mehmet Ali Öztürk (21703425)

Instructor: Eray Tüzün

Teaching Assistant(s): Mert Kara

Table of Contents

1. Introduction	2
1.1 Purpose of the System	2
1.2 Design Goals	2
1.2.1 Usability	2
1.2.2 Visual Appeal	3
1.2.3 Security & Privacy	3
1.2.4 Maintainability	3
1.2.5 Performance	3
2. High-Level Software Architecture	4
2.1 Subsystem Decomposition	4
2.2 Hardware/Software Mapping	5
2.3 Persistent Data Management	5
2.4 Access Control and Security	6
2.5 Boundary Conditions	7
2.5.1 Initialization	7
2.5.2 Termination	7
2.5.3 Failure	7
3. Low-Level Design	8
3.1 Object Design Trade-offs	8
3.2 Final Object Design	9
3.3 Layers	10
3.3.1 User Interface Management Layer	10
3.3.2 Web Server Management Layer	11
3.3.3 Data Management Layer	14
3.4 Packages	16
3.4.1 Packages Introduced By Developers	16
3.4.1.1 Entities	16
3.4.1.2 Security	16
3.4.1.3 LoginAndSignup Management	16
3.4.1.4 WaitingBin Management	16
3.4.1.5 University Management	16
3.4.1.6 Task Management	16
3.4.1.7 Application Management	16
3.4.1.8 Document Management	16
4. Glossary and References	17

1. Introduction

1.1 Purpose of the System

The website will be designed for the application of Erasmus and Exchange. The main goal of the website is to reduce the paperwork by making the process of application web based. All different users will have an account and users will be able to follow the process. Also, it aims to facilitate communication between users when there is an available space in a university for the students who are not placed or during the post approval, if the uploaded documents are invalid or insufficient, it can be clarified by the instructors.

1.2 Design Goals

As it mentioned that the goal of the Erasmus Exchange Program, to ease and digitalize the process, after placements are done, to decrease the paperwork of the users. One of the goals of the website is that the interface should be clear and neat, that is why usability and visual appeal is crucial for the project. Since some private data will be stored in the system, it should be securable. Additionally, the system should meet the needs of the users for maintainability.

1.2.1 Usability

The website should be easy to use even for the ones who have never used before because it is the main purpose of the website for all kinds of users to complete the tasks properly and more easily. Otherwise, if the website has poor usability, it decreases efficiency and harms the maintainability.

1.2.2 Visual Appeal

People check out the visual appearance when comparing things with the same features. Therefore, it is important to have the attention of the users and to make a good first impression. It is possible that visual features can outweigh the functional requirements, so a system that is pleasing to the eye has great advantages.

1.2.3 Security & Privacy

This website not only will contain a lot of private documents that are especially relevant to the academic careers of the students, but also personal information and interaction of the different users, so the website should be securable and reliable. Besides, a secure system helps to prevent people who want to access private data.

1.2.4 Maintainability

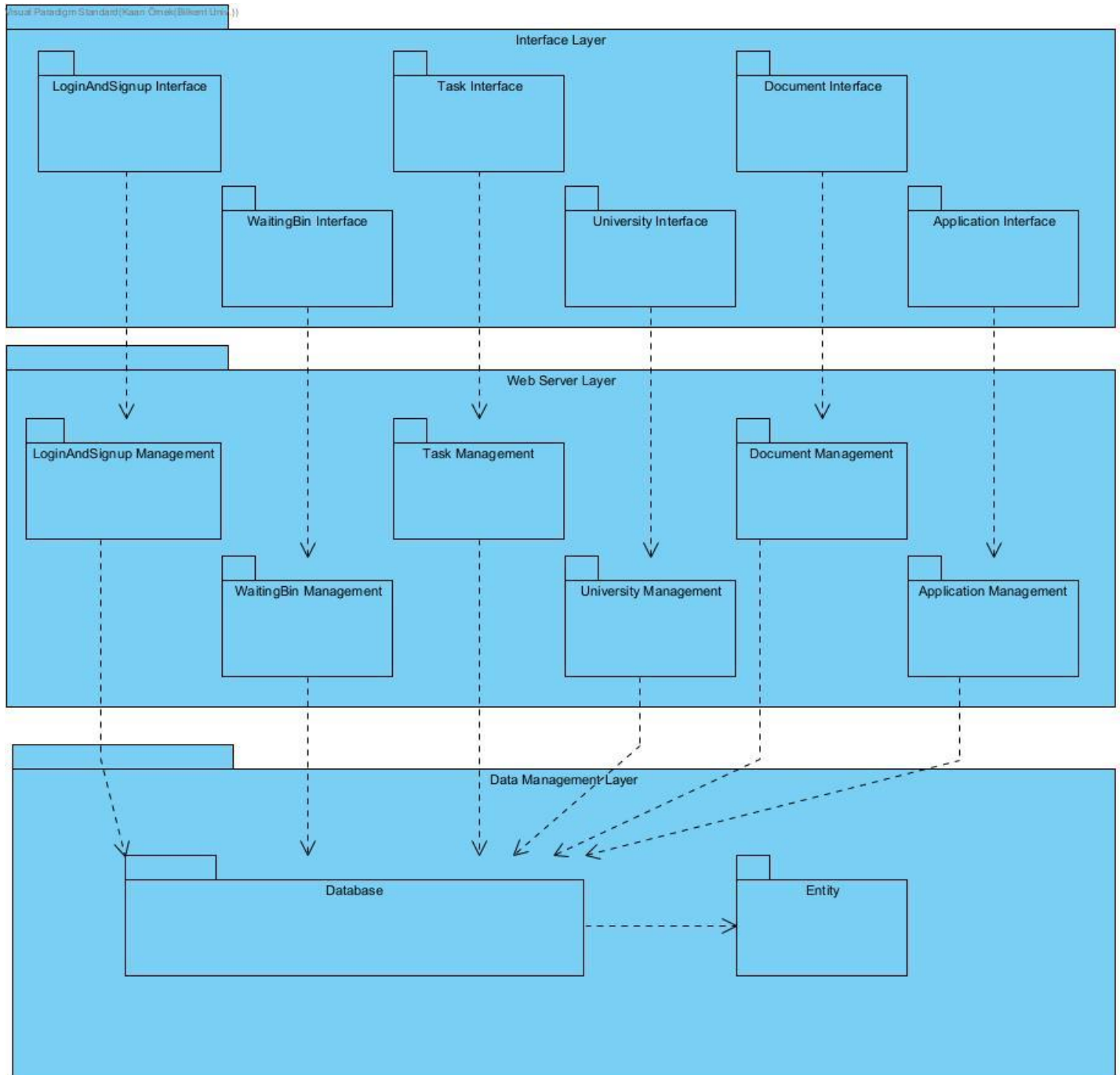
Over time, the interests and requests of the users may change. That is why it is important to ensure the continuity of the project. Maintainability gives the system the ability to change the functionality of the site. Any unwanted part can be removed or a part can be added simply to increase impact on the users and improve the ability of the system.

1.2.5 Performance

The performance of the design should be qualified since the “speed”, “load time” and “accessibility” affect the efficiency of the website. Websites with a good performance, a better quality product can be offered to users.

2. High-Level Software Architecture

2.1 Subsystem Decomposition



In order to get a well-connected system, we consider implementing a new system which has three layers. The Data Management Layer in Figure 2.1 includes Entity and Database subsystems. The Entity subsystem has all classes of the website and the

Database subsystem has some interfaces which inherit JpaRepository to make database connection between classes and entities.

The Web Server Layer includes all back-end processes which are working in the background. These back-end processes are different from database processes. The back-end systems have different subsystems which are used to manage and control core functionalities of the website. For instance, a web server layer component can have a controller and entity classes as a management part of that functionality.

The UI Interface Layer is configured as a boundary object of the system. This layer has interfaces to create connections between user and website pages. This layer has a crucial role for the front-end process because a user who wants to use website services makes contact with the website pages directly. The interface layer is a helper for functions and website pages' connection.

2.2 Hardware/Software Mapping

While doing the project, there is a mandatory implementation which is that the system should work on a web browser. According to this, we implement this project for phones and computers which have access to web browsers via the internet. A user who wants to use our web services should have a keyboard, a mouse, a monitor, essential hardwares of the computer or a mobile phone. This project will be used in web browsers like Mozilla, Google Chrome, and etc. Although the system services will be suitable for modern web browsers, some browsers which are not updated may not be suitable tools.

2.3 Persistent Data Management

In our project, we decided to use MySQL. The main reason for this selection is that many of the group members have knowledge about MySQL and its usage. Our entities and their determined attributes will be held in the database server as data. By using SELECT, UPDATE, DELETE and INSERT commands, we are planning to use and edit data without encountering any challenge. By making dynamic access to objects thanks to MySQL, every user can change the object attributes by using website interfaces. We will

be working on creating proper queries to obtain correct syntax of the PULL data from the database and PUSH data to the database.

2.4 Access Control and Security

Class	Instructor	Coordinator	Student	Faculty Board Member
LoginPage	X	X	X	X
InstructorSignUpPage	X	X		X
StudentSignUpPage			X	
MainPage	X	X	X	X
Navbar	X	X	X	X
ApplicationsPage		X	X	
CreateApplicationPage			X	
ChangeApplicationPage			X	
TaskListPage	X	X	X	X
UploadTaskPage		X	X	
CreateTaskPage		X		
UniversityPage		X		
WaitingBinPage		X		

Our ReadyCodeGo project aims to service different access designs according to the user type. In our project we have three types of users which are students, coordinators and teachers. Students have the ability to create an application and after that overall access the info of that application. However, in order to change their application's specifications they need to take permission from coordinators. Coordinators overall access all the permissions in the system. We are determining the user type in the login page. We are aiming to establish security in our website with the help of SSL protocol. All the connections from our website to the user will be secured by SSL.

2.5 Boundary Conditions

2.5.1 Initialization

Since the system will be implemented for web browsers, there are not any setup requirements for general users. All parts of the system(back-end, front-end and database) will be held in web servers and these servers are going to work in %99.9 uptime guarantee. These uptime guarantees will be provided by web server companies and will not be linked with us directly. Basically, a user who has an account in our databases can reach our services via the internet. An important restriction for unregistered users is that they can see only the main page, login and register pages rather than our functional pages. All data including our pages are going to be held in our database. Users can log into their accounts by using different web browsers simultaneously.

2.5.2 Termination

Our website has many subsystems and helpers as a whole architecture. If some parts of it lose its functionality and stop working, our website and services will be completely affected. In addition, to make maintenance, admin users can suspend our services for a short time. Before doing this, all user actions and data will be saved in the database by the admin. All data and actions will be preserved in our database services without any data loss.

2.5.3 Failure

Our website will be held on online servers. The companies which provide us this source may encounter some critical errors. While these errors arise, our web services can be restarted or offline for a short time. If this failure can not be resolved, our team members which have admin account privileges send a notification for our users, and some unsuccessful HTTP/S response of the web servers may be occurred. Our team will try to fix these critical errors as soon as possible.

3. Low-Level Design

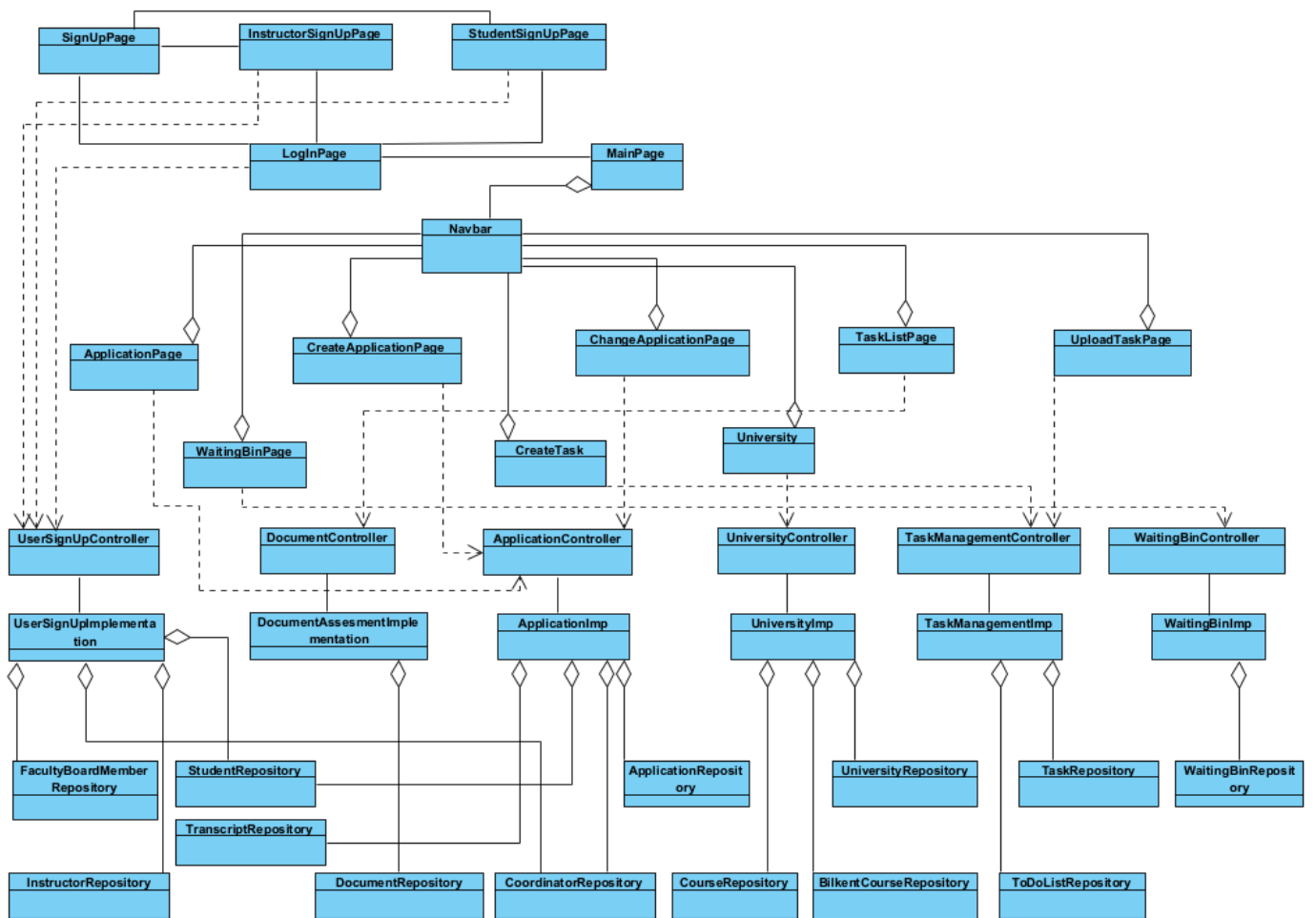
3.1 Object Design Trade-offs

Maintainability over Performance: Websites will be designed for different users and the interest or requests of the users may change. Since it is a complex project, there will be lots of classes and changes will be made in the components which will ensure the maintenance. On the other hand this will lower the performance level.

Functionality over Usability and Performance: Erasmus Exchange program will have different types of users. Each user will have different functionalities which will lead to a great complicity that results with a decrease in usability and a poor performance.

Usability over Visual Appeal: The main goal of the system is to reduce the workload of the users by digitalizing the application process. Thus, it is important to have a clear and easy to understand interface. That is why the system will have simpler interfaces instead of aesthetic and complex interfaces.

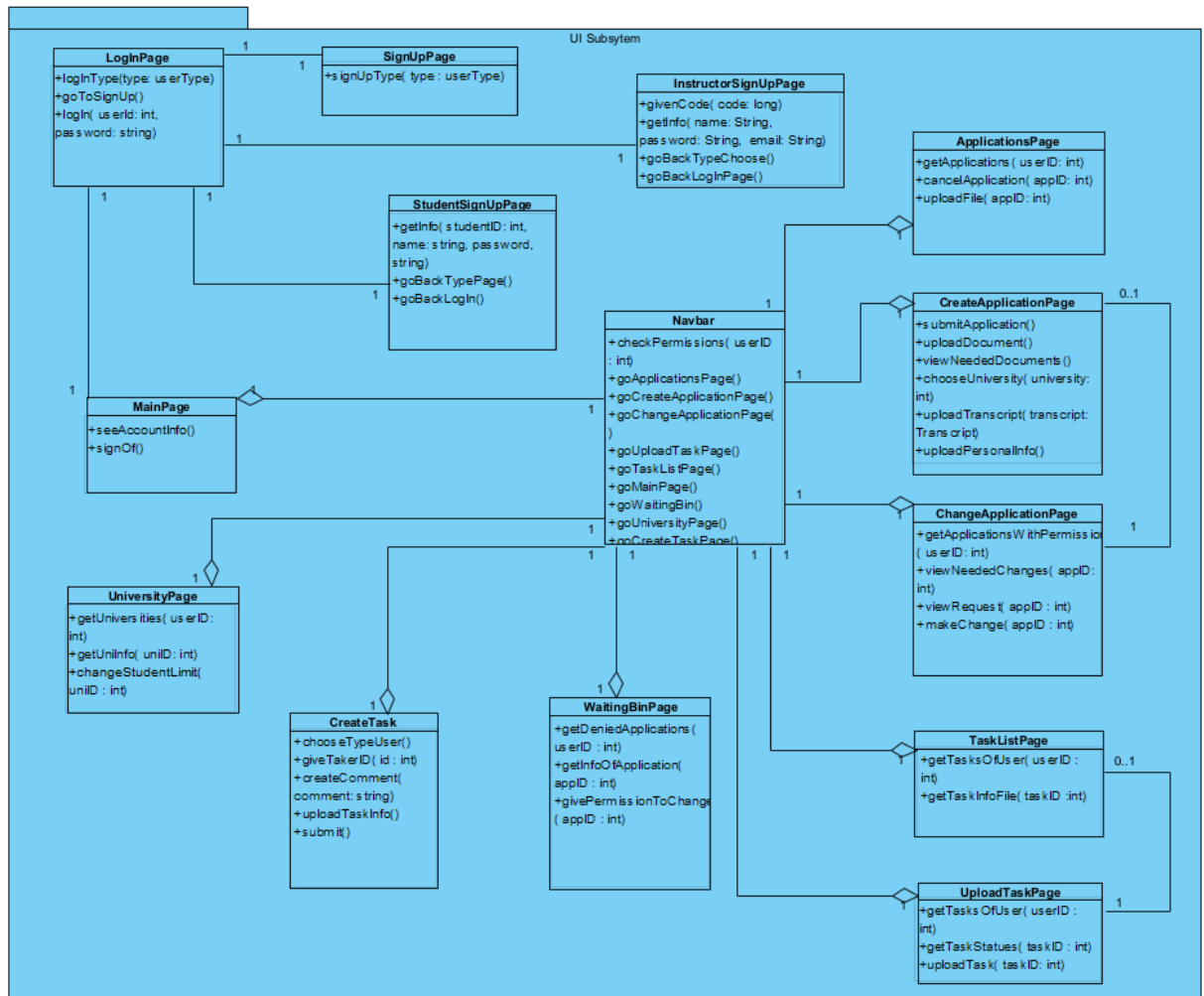
3.2 Final Object Design



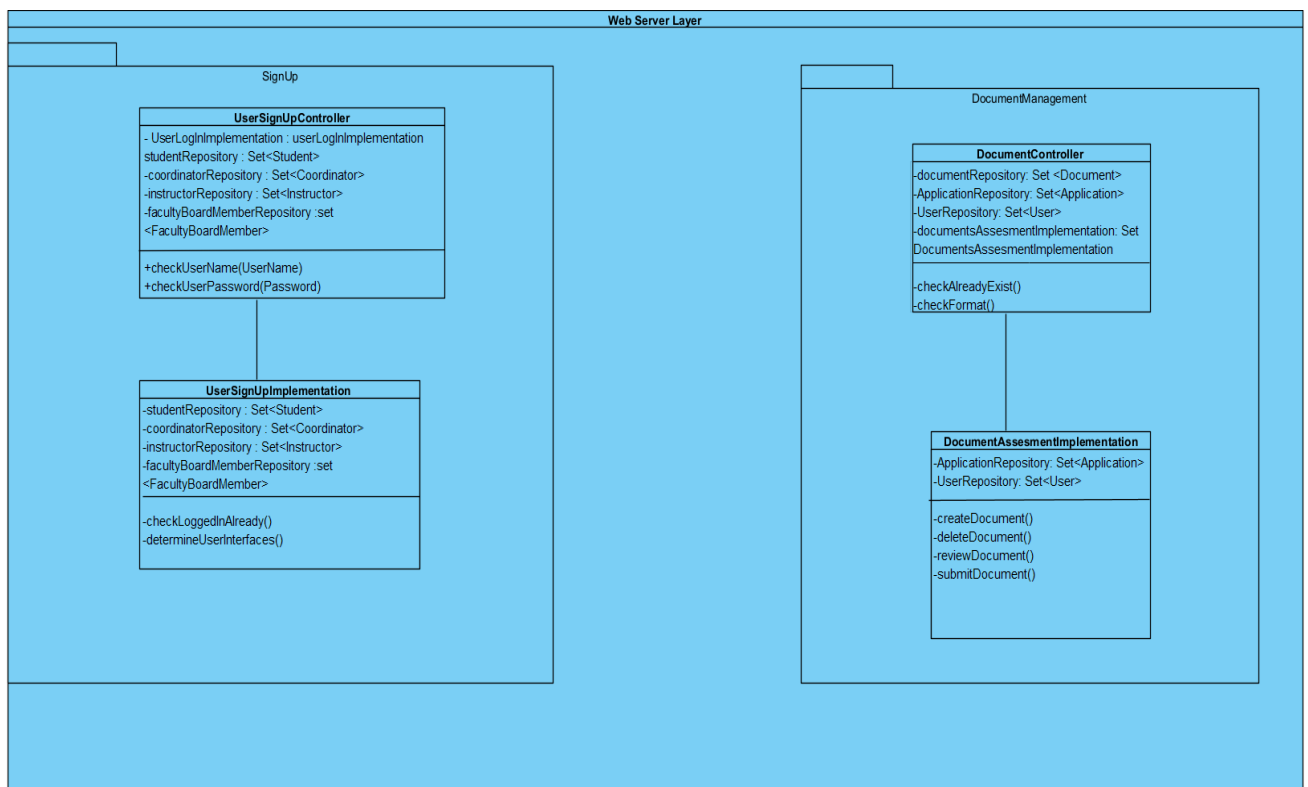
3.3 Layers

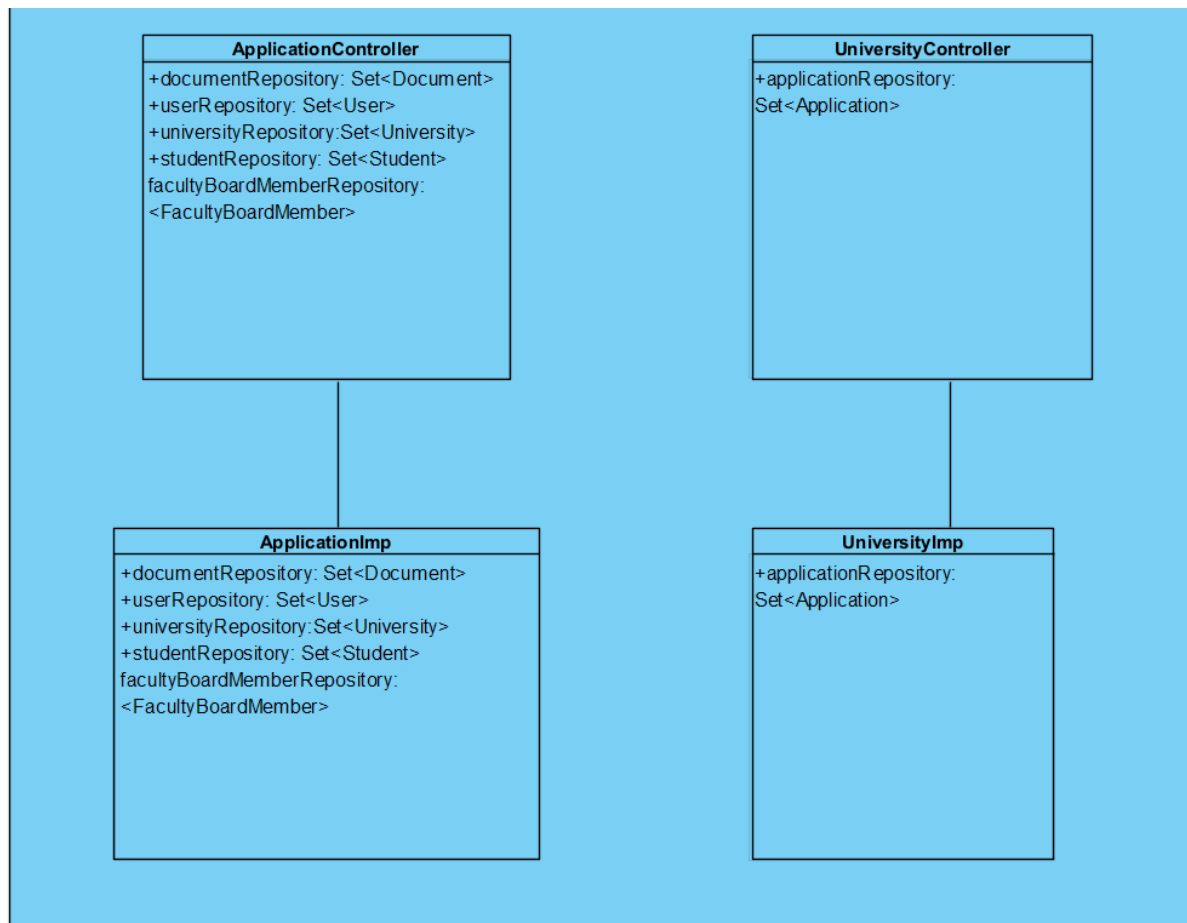
3.3.1 User Interface Management Layer

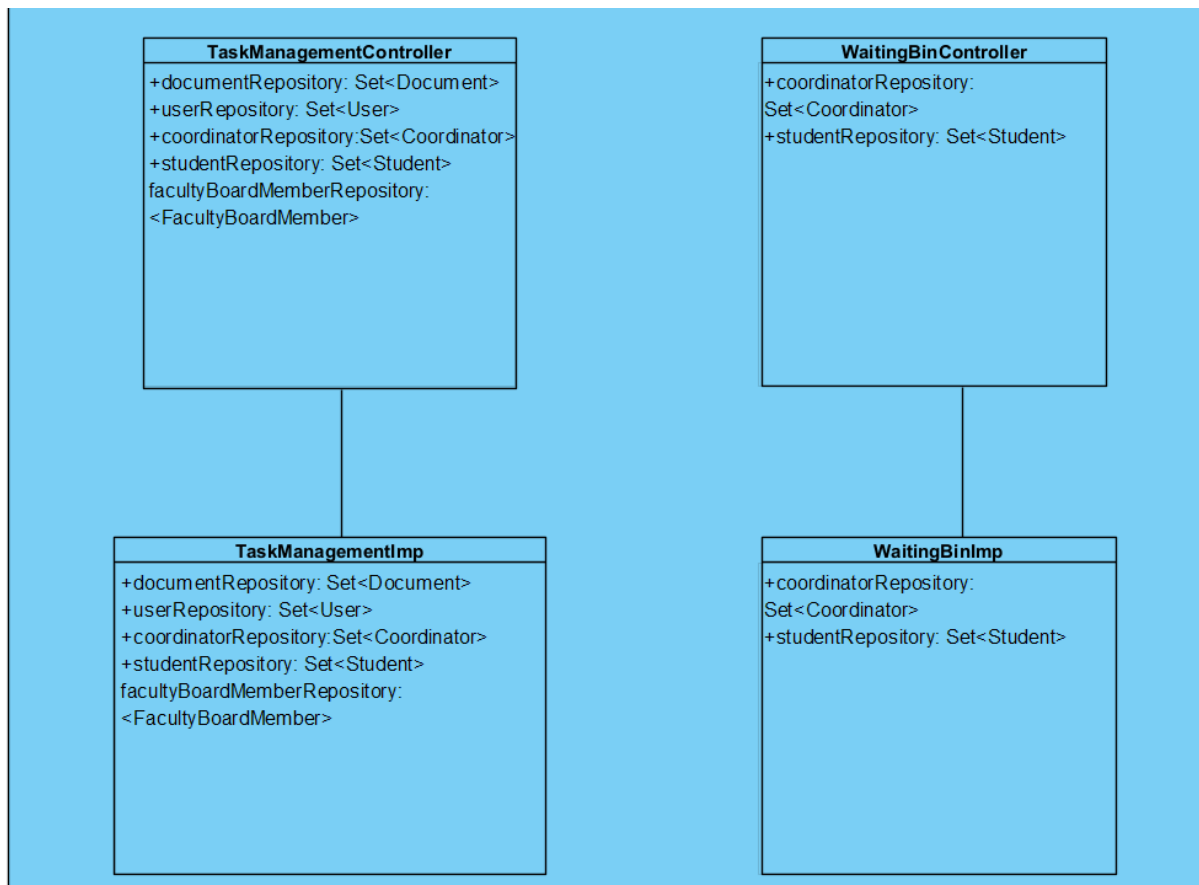
All objects in this layer will be HTML format. With the help of the “div” and “button” elements of HTML, our objects can communicate with each other. All of the objects in this layer are boundary objects therefore, it can communicate with the user (actor), control objects and other boundary objects.



3.3.2 Web Server Management Layer

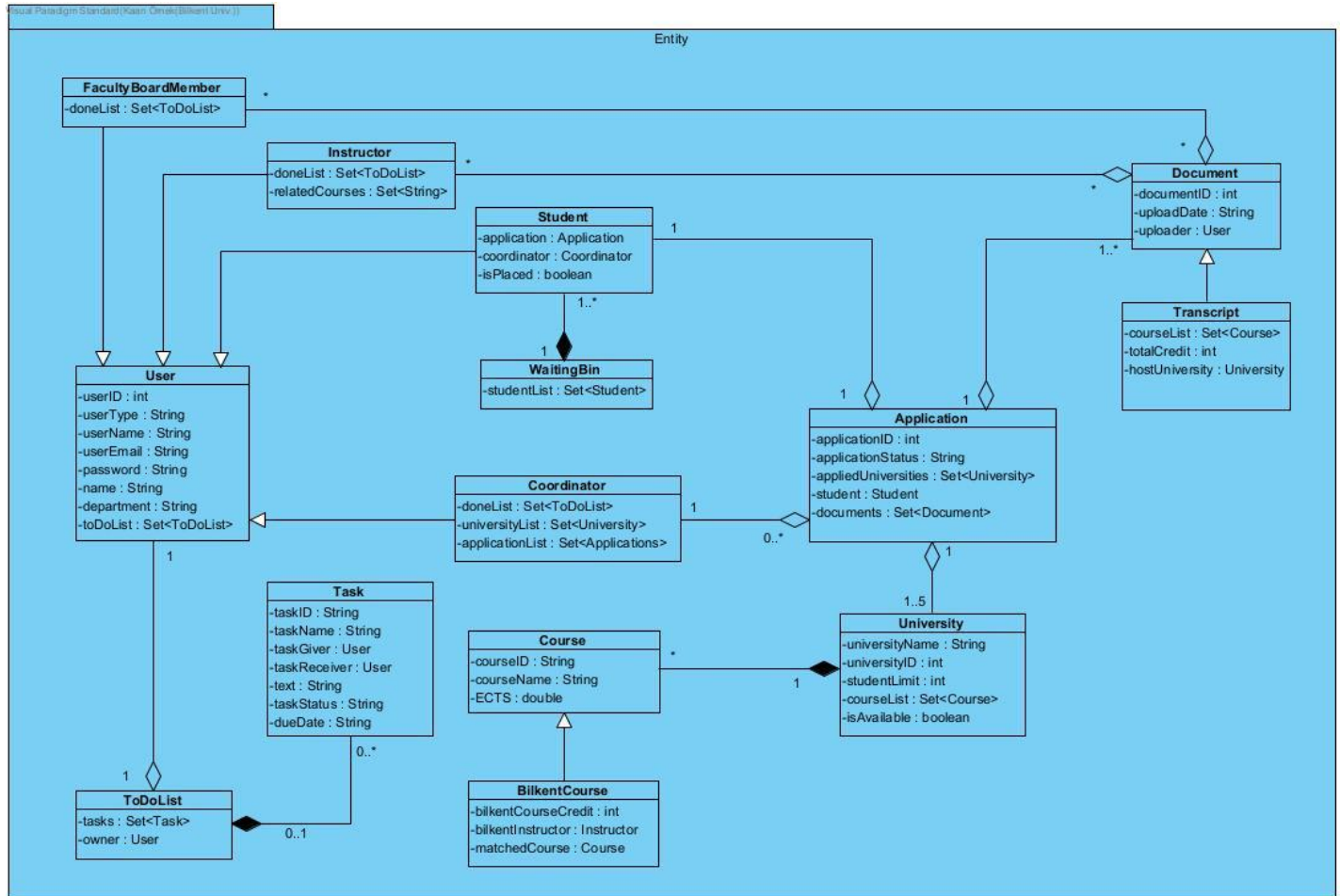


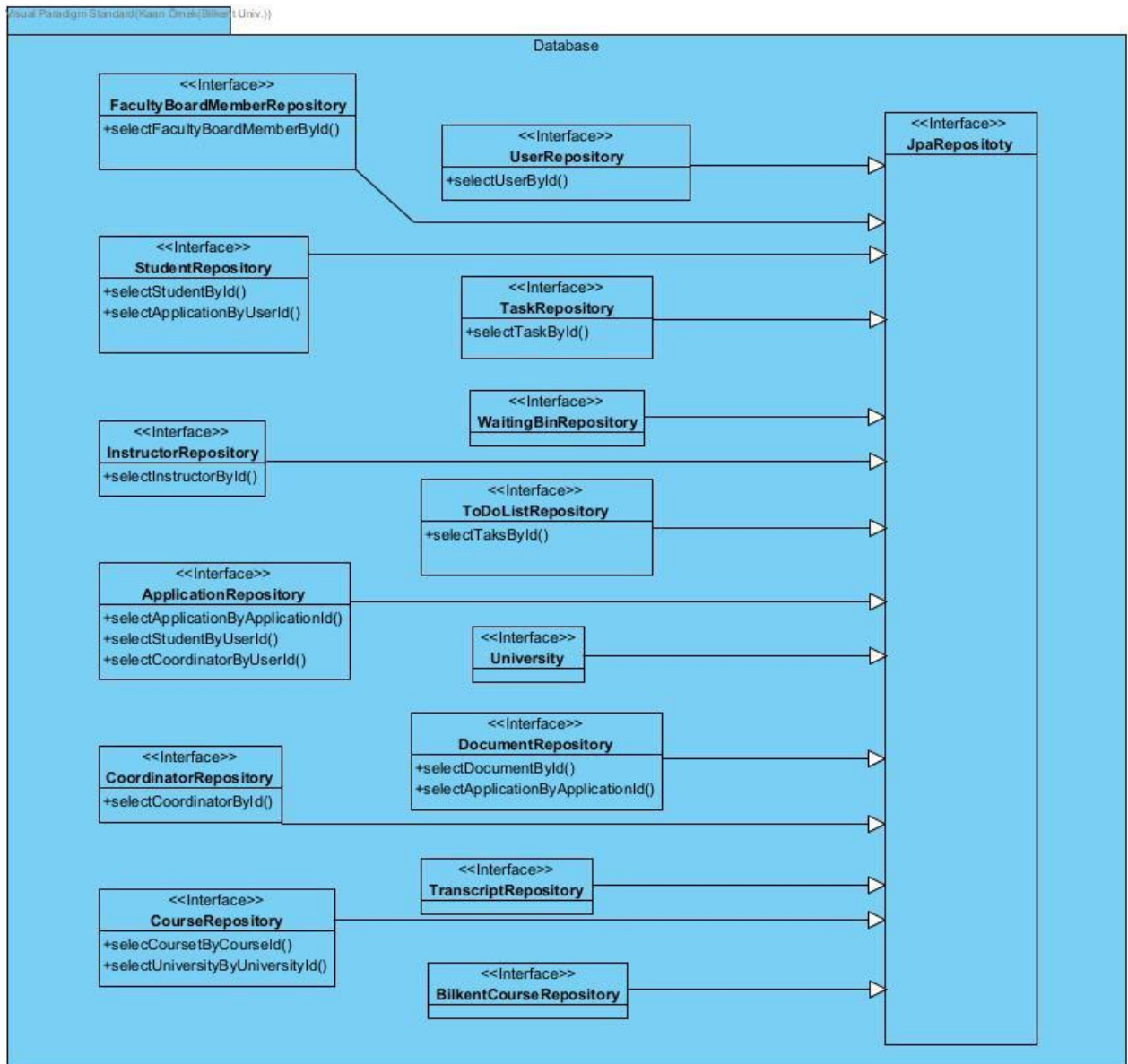




Web Server Layer operates the necessary actions to respond to requests from hosts communicating with Data Management Layer.

3.3.3 Data Management Layer





Our program has Entity and Database subsystems as a whole Database Management Layer. The entity system of our program includes all entity classes of the system. These classes are configured according to the class diagram. Entity subsystems show us how the entities of the system can communicate with each other in the database. As for database subsystems, interfaces of the system entities inherit from JpaRepository to communicate with the database.

3.4 Packages

3.4.1 Packages Introduces By Developers

3.4.1.1 Entities

The package is created to include the Entity class which has all getter and setter methods and objects. These elements can help developers to create and edit all entities of the program.

3.4.1.2 Security

The package is created to include the Security class which handles all security processes of the website.

3.4.1.3 LoginAndSignup Management

The package is created to include login and signup classes to manage all processes and actions of these pages.

3.4.1.4 WaitingBin Management

The package is created to include the student class to manage the waiting list adding and deleting process.

3.4.1.5 University Management

The package is created to include university classes to manage the application process which is about universities and their quotas.

3.4.1.6 Task Management

The package is created to include task class and ToDoList class to make connection between tasks and different users.

3.4.1.7 Application Management

The package is created to include an application class to handle all application processes by editing and creating.

3.4.1.8 Document Management

The package is created to include a document class to manage all document objects from different classes(Transcript, Application etc.)

4. Glossary and References

<http://cs.bilkent.edu.tr/~exchange/procedure.html>

-Object-Oriented Software Engineering, Using UML, Patterns, and Java, 3rd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2010, ISBN-10: 0136066836.

-Alkan, Can (2022) " Applications for Erasmus and Exchange [PowerPoint slides] Computer Engineering, Bilkent University.