CS 319 / Object-Oriented Software Engineering Term Project:

“Mastering Bilkent”

Design Report

Project Group 1.A:
Ertan ADAY
Furkan Arif BOZDAĞ
Rümeyza DİNÇER
Course Instructor:
Bora GÜNGÖREN
Teaching Assistant:
Gülden OLGUN

1. Introduction

1.1 Purpose of the System

Mastering Bilkent is a desktop application intended to help students of Bilkent University to study and understand the content of their scheduled courses with supplementary materials such as pdf documents, videos and quizzes provided by their instructors of each course. Although, the main target of the application is Bilkent student, with the permission of instructor courses shall be accessed by public. Therefore, our main aim is to become a desired place for self-education with academic level of quality.

1.2 Design Goals

We first examine the design goals of the system before move on to system architecture.

User Friendliness & Ease of Use: Since Mastering Bilkent is an educational application, users shall adapt our application easily. We don't aim to serve a complicated and not understandable application. Thus, our application considers user friendliness as main criteria. We will provide an easy to learn user interface. Our application is planned to have different layers for instructors and students as we discuss in the analysis stage. Users shall easily access the courses they are looking for via the search bar. A student can access a course which s/he registered from the list on his/her homepage and instructor shall access a course which s/he created from her/his homepage. Users shall easily understand how to use course-related activities (attending/creating a quiz, reading/uploading documentation or watching/uploading video files) by the simple menus and instructive buttons clickable via mouse.

Modifiability: In Mastering Bilkent, students should be able to change their personal information such as profile pictures, addresses, and other general information about themselves. Moreover, Instructors will also change personal information about themselves, and also they will modify their course pages very often. Therefore, Mastering Bilkent must provide a modifiability.

Availability: Mastering Bilkent is a non-commercial enterprise application, it must provide round-the-clock services. That is, our application need to work 24 hours, 7 days, with low amount of downtime per year. It should have a maximum downtime of 90 hours per year. This is why our application designed to satisfy high availability criteria.

Reliability: Mastering Bilkent must keep the data, which is uploaded by both students and instructors, without corrupting it. Since this application can be used by an instructor as an official course homepage, any error that appears related to the course documents can't be accepted. If a student attempts an assignment quiz, its result should be sent to the instructor correctly. Moreover, Mastering Bilkent is an educational application, any corruption in the course documents or course activities would cause the latency in educational progress. Thus, Mastering Bilkent will be implemented in a reliable way.

Good Documentation: Our application will contain substantial amount of documents which are in audio, video, or text formats. In order to provide a good service to users, Mastering

Bilkent should be documented in a good manner. That is why the design should include good documentation.

Scalability: Since Mastering Bilkent is an educational application that will separate the contents by its semesters and years, it should handle the growing amount of data and work. In order to be an efficient educational application, Mastering Bilkent should a good scalability property.

1.3 Trade-offs

Scalability vs. Long Term Reliability: Mastering Bilkent will not store all the growing data in its database and manage it in terms of scalability efficiency. In order to provide sufficient scalability, there will be some trade-offs from long term information.

Good Documentation vs. Memory: In order to provide a good documentation for different type of documents in the forms of text, video, or audio, our application will use substantial amount of memory. So there will be memory tradeoffs in order to provide a good documentation.

User Friendliness & Ease of Use vs. Functionality: It is very important for Students and Professors to be effective and not waste their time navigating around the application. Therefore, Mastering Bilkent will be a user friendly and easy to use application. The system should not be too complex to engage in. It means that the functionality of the system will be basic. Since the purpose of the system is to provide users a friendly and easy to use learning environment, we will focus on the usability of the system rather than making it functional more than necessary. The system has a simple interface and familiar instructions to act instead of complex menus and various features. Thus, the users can spend time studying rather than struggling with complex features.

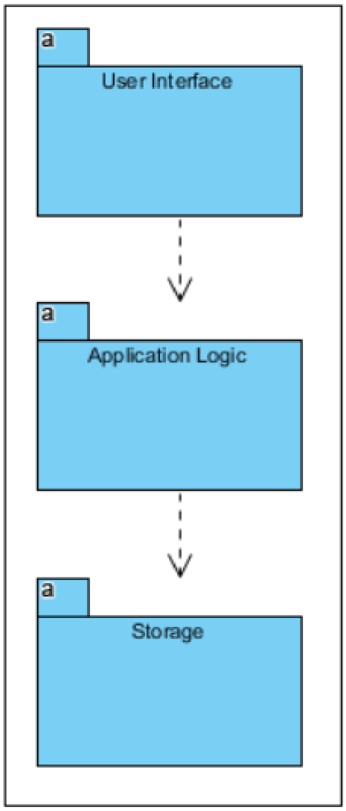
2. Software Architecture

Overview

Our main goal in this section is to decompose whole system into cohesive subsystems. In order to achieve this we will be using Three Layer Architectural style.

2.1 Subsystem Decomposition

In order to achieve the goals that are defined in 1st section, the system should be broken into three main pieces. This act of breaking into pieces is subsystem decomposition. Those so-called pieces are subsystems provide us which layers are associated with each other in which hierarchy, and also which classes are associated with each other in what terms. The classes with similar jobs belongs to the same subsystem components.



2.1.1 First Layer - User Interface:

Our first layer is user interface, which includes User package in it. This layer is responsible for user interface creation and providing user friendliness and ease of use for the users. This layer scans the input from user’s keyboard and mouse, and conveys this input to the second layer, which is application logic. Further, this layer is also responsible for differentiating interfaces for different roles of users, which are student and instructor.

2.1.2 Second Layer - Application Logic:

This layer’s job is to do the desired job that is directed from the above layer, and also interpret the data coming from below layer, which is storage. This layer mostly deals with the main activities in the application. Inside this layer, there are profile management and content management packages. Profile management deals with the logic behind users’ profiles, and content management deals with the logic of contents of application related issues, such as quiz management and course management.

2.1.3 Third Layer - Storage:

In this layer, there is a component that handles and simplifies the connection of database related issues. This layer’s job is to provide course related information to the users, and handle the authentication-related issues of a user, according to his/her role. This one will insert and delete data to or from the database such as inserting a new quiz to the database, or deleting an existing one. It is also responsible for saving and updating users’ personal information, course list, and other storage-related problems. Moreover, 3rd layer is also responsible with stopping the user interaction with application if their connection is lost, or any other system problems.

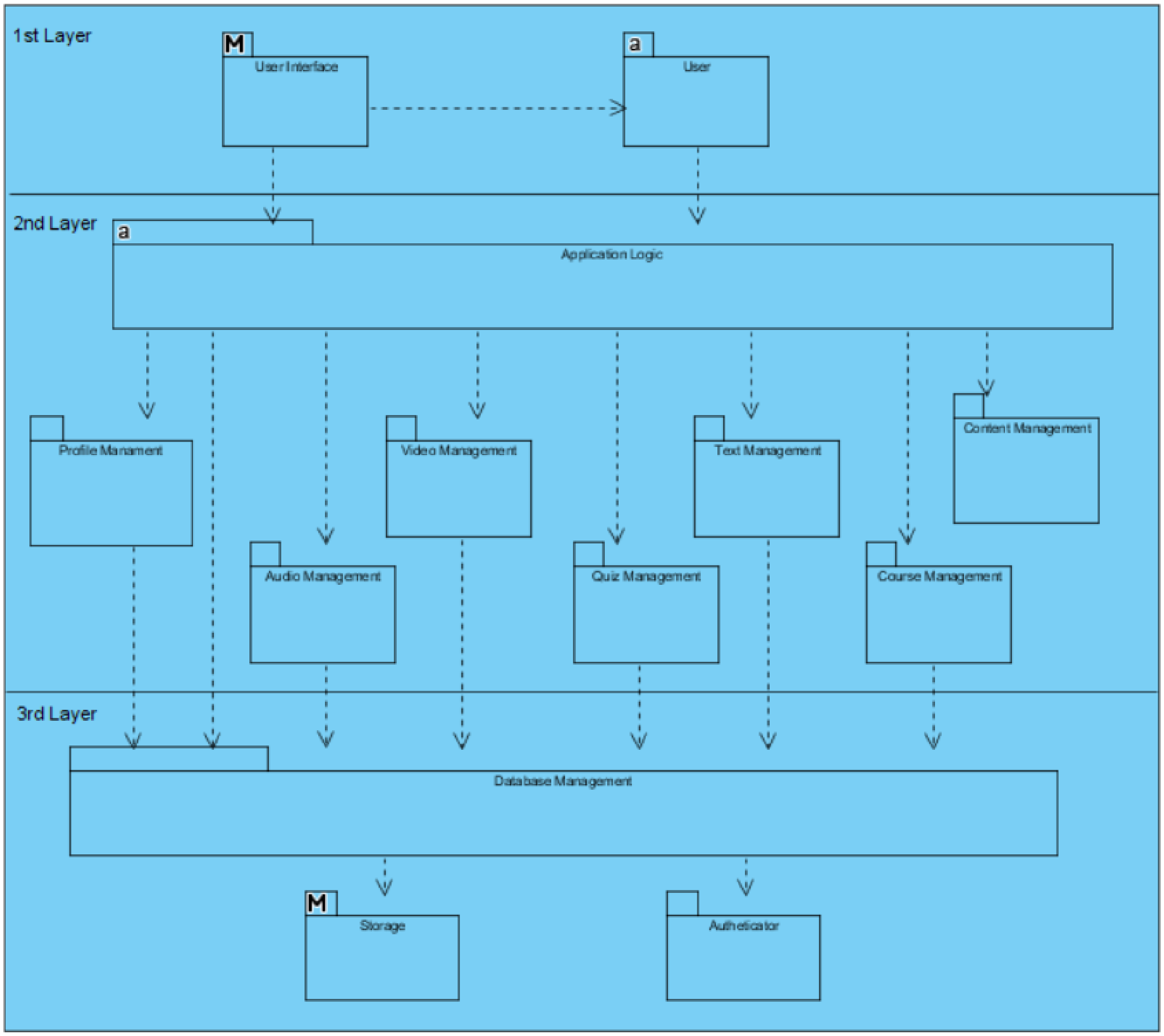


Figure 1 (Basic Subsystem Decomposition)

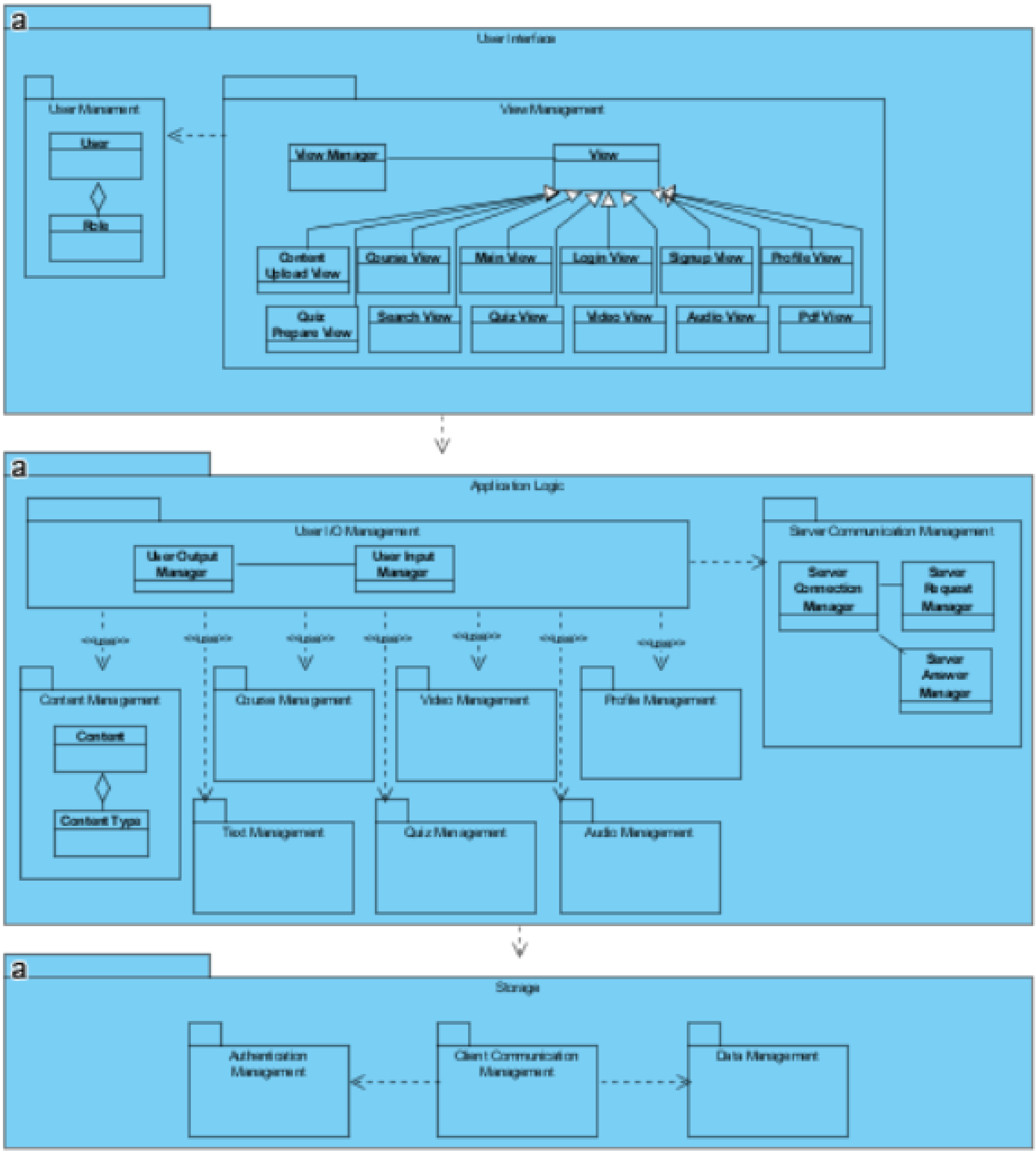


Figure 2 (Detailed Subsystem Decomposition)

2.2 Hardware/Software Mapping

Mastering Bilkent will be implemented with Java programming language via JDK 8 and for GUI of our application JavaFX 8 will be intended to use. Thus, any computer having a platform which has JVM can be used to run our application.

For hardware, keyboard is needed for giving inputs to the application such as entering user profile information, entering course related information or searching a specific course name. Mouse is needed for also giving input such as clicking the wished course and wished course activity and a basic monitor is needed for interact with the system via GUI.

For storage, a database created with SQL. In order to bridge the gap between Java code and database, Java Database Connectivity (JDBC) API is needed.

2.3 Persistent Data Management

Mastering Bilkent has to store data related to users and courses. Thus, we can't use a temporary storage technique. A database implemented via SQL will be used.

*List of users with their roles,

*User's profile information,

*Registered course list for each student,

*Created course list for each instructor, *List of all courses with registered student list,

*Courses' documentation will be stored in our database.

2.4 Access Control and Security

Mastering Bilkent is intended to be an application which can be accessed by multiple users at the same time from different computers. Although an enterprise application need to satisfy this condition, because of the limited implementation time and CS 319 Course focuses on object design rather than database and internet connectivity we decide to implement our application to run on one computer for demonstration. Access through many computers will be leaved as future work at this point.

We will provide the security of the users' personal information. No user can access the other users' data. Also, nobody can modify the course's data other than its own instructor. Course's and user's information will be stored safely.

2.5 Boundary Conditions

Enter:

User shall enter the program by clicking the application icon. Application will be opened by .jar file.

Exit:

User shall exit with X button at the right top of the interface frame.

Error:

Until there won’t be a problem related to database, users won’t lose the information and error screen will provide the user. However there is always a little risk to lose data. If data are lost, system will require from user to login again and courses must be created from beginning also. If connection to database is lost due to the user’s system problem an error also will be shown.

3. Subsystem Services