

File-Sharing

SYSTEM DESIGN DOCUMENT

1.0

1.04.2022

Alparslan Yazıcı 216CS2029

Prepared for

COMP4902

Graduation Design Project



IŞIK UNIVERSITY
**COMPUTER
SCIENCE AND
ENGINEERING**

SYSTEM DESIGN DOCUMENT

1. Introduction

A software design document helps to ensure the design specs of the software are understood and it's clear to all. It specifies what is possible with the product and how it can be accomplished.

1.1 Purpose of the System

The purpose of this web application is to enable file sharing between users. Users who visit the website must first become a member of the site. After member users login to the system, they must find the people they want to share files with using the user search feature and add them as friends. Using the file share feature, users upload the file which they want to share to the system and then select the people to share. Users with whom files are shared can view information such as sender, file name, file size, etc. if they wish after viewing information they can download the file. Also, users who upload the file can delete the file or end up sharing it.

1.2 Design Goals

- **Memory:** The system should have 1Gb memory for per user.
- **Robustness:** The system should be catching the invalid user inputs and should be inform the user.
- **Security:** The system should be secure. Users should be login to use all website. Also, users cannot reach documents which are not shared with him.
- **Development Cost:** There is no development cost. There is not any license fee required to develop the system.
- **Readability:** .Net naming convention should be followed. Should be coding in accordance with the clean code approach.
- **Usability:** Users should be using the website without any tutorial. User interface should be soft and understandable.

1.3 Definitions, Acronyms, and Abbreviations

- DB: Database
- RAD: Requirement Analysis Document
- SDD: Software Design Document
- MVC: Model-View-Controller architecture.
- Visitor: Not logged in user.

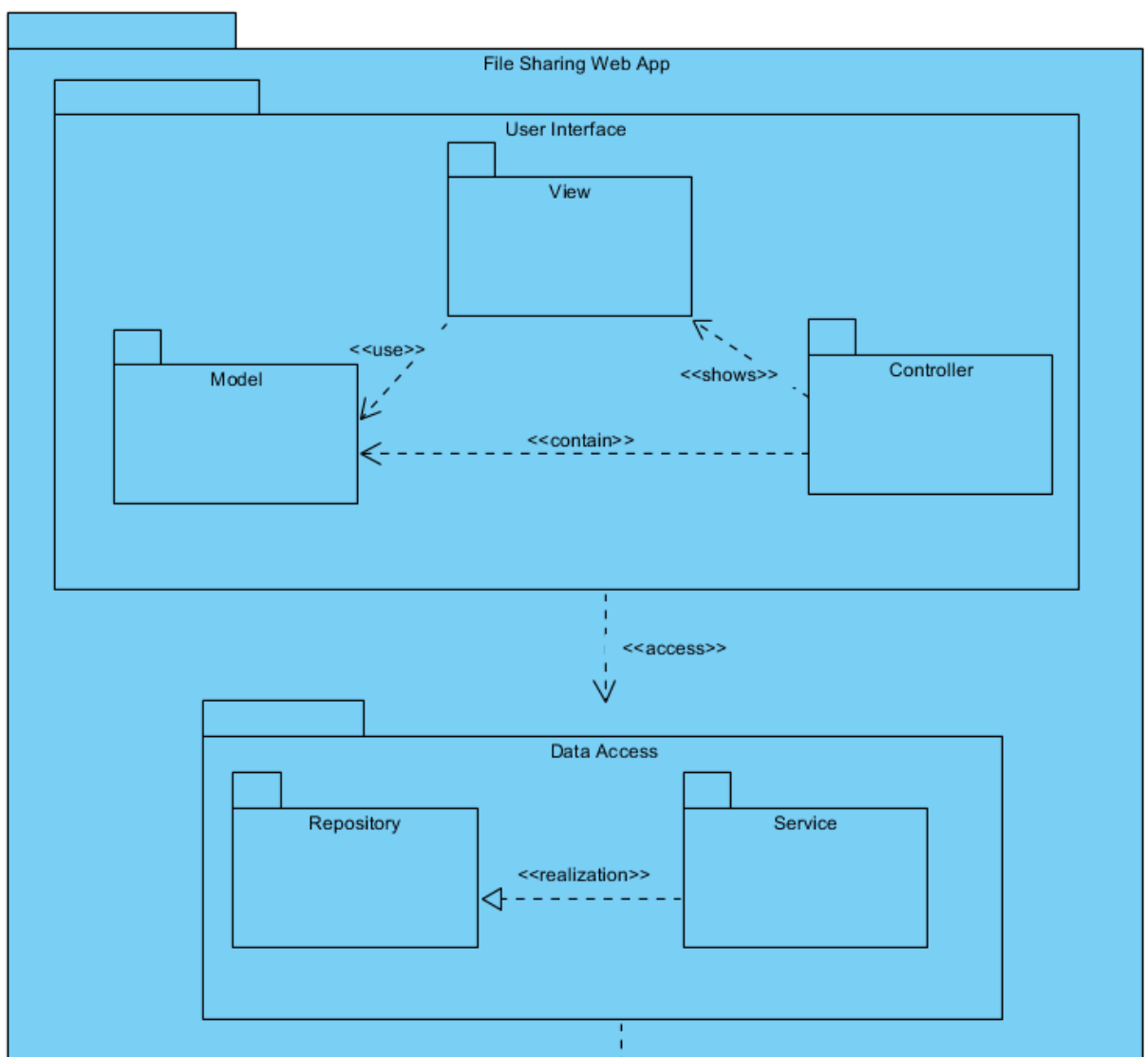
2. Proposed Software Architecture

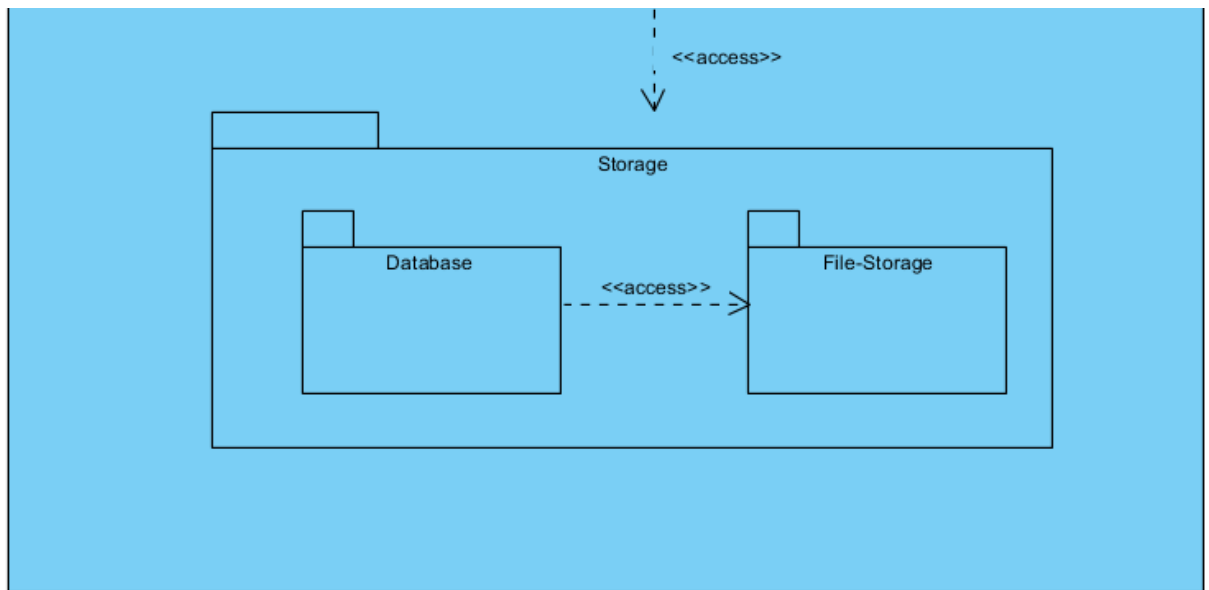
Model-View-Controller (MVC) architecture used for the system design. In this architecture, there are 3 main subsystems. The View subsystem is responsible for how data are shown to the users, Model subsystem stores data that are the application need. The controller subsystem is where controls, calculations, updates... etc. are performed.

2.1 Overview

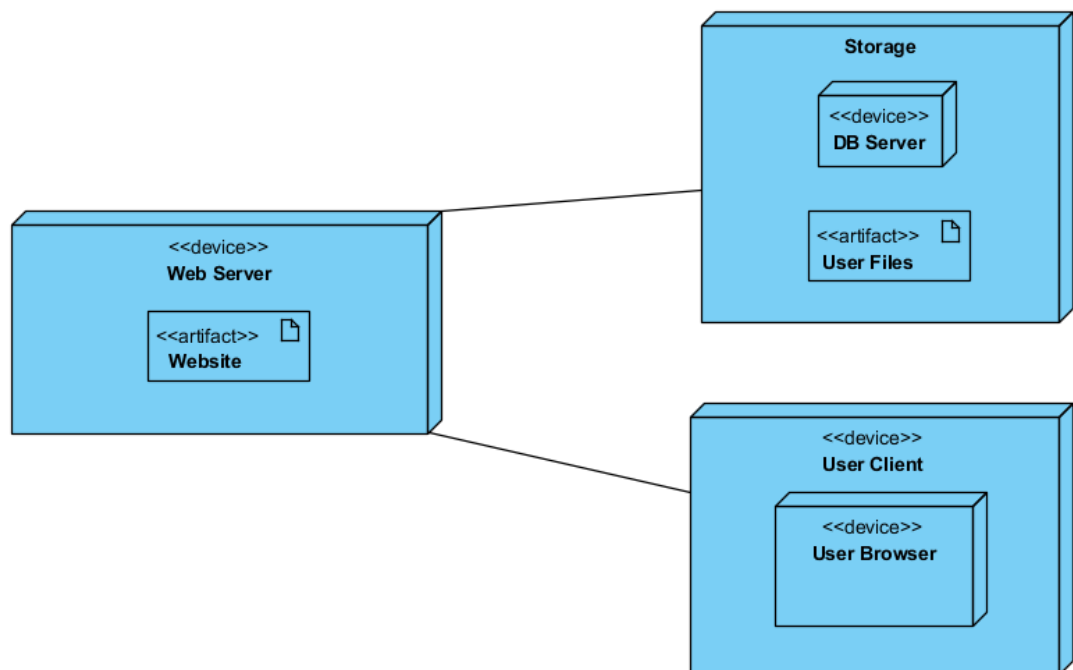
The system build with 3-tier architecture; Representation layer, Business Layer, and Data access Layer. System decomposition part shows relations these 3 components.

2.2 System Decomposition





2.3 Hardware/Software Mapping



2.4 Persistent Data Management

File sharing web application provides file transfer to users. To achieve the purpose, application needs two different data store management. The first one is for user files. The second one is system database, it keeps user information (name, email, password, etc.) and relations (file owner, friendship, etc.).

2.5 Access Control and Security

	User	Visitor	Document	Application
Visitor				Register()
User	AddFriend() AcceptFriendshipRequest() RejectFriendshipRequest() DeleteFriend()		UploadFile() DownloadFile() DeleteFile() ShareFile() EndSharing()	

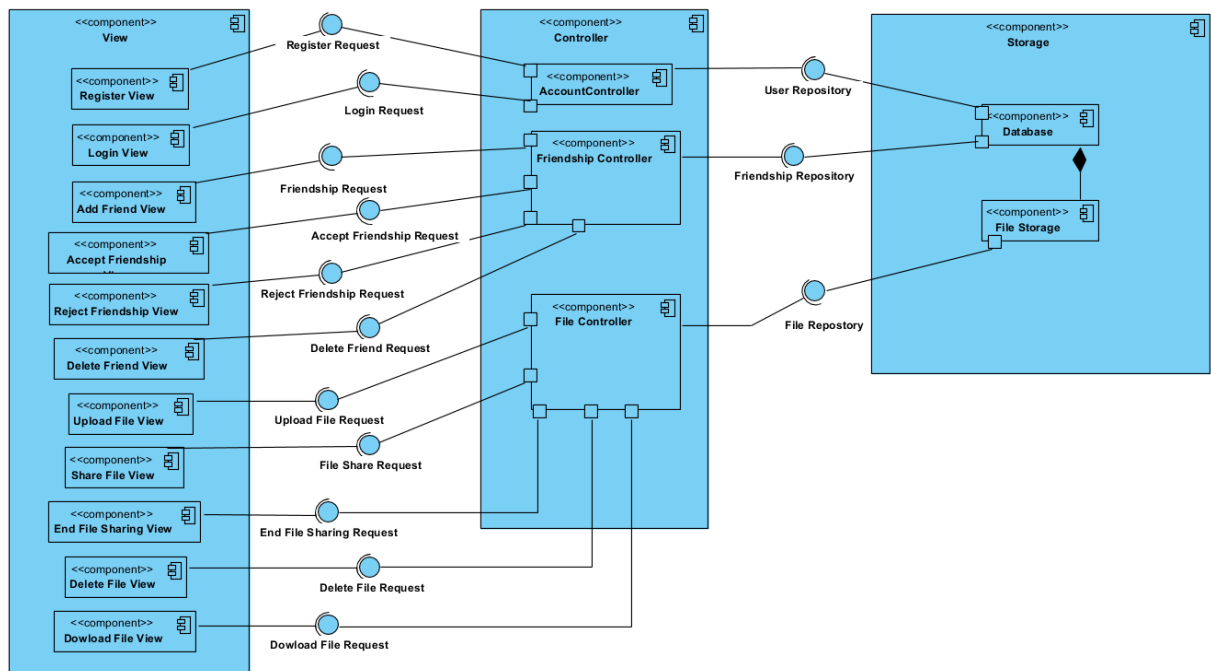
2.6 Global Software Control

Users send HTTP requests using the user interface, controllers catch these requests and control authorization & authentication. After catching the request, controllers invoke associated functions from the Data Access Layer via interfaces. On the data access layer, the system checks business logic and performs DB operations if required.

2.7 Boundary Conditions

The system should not crash as a result of any error. It should catch and manage system errors. To prevent user-induced errors, the inputs sent by the user should be validated, and if there is a mismatch, it should send a message to the user. This process should continue until the user submits the appropriate entry.

3. Subsystem Services



4. References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. Ekin, Emine. “*System Design: Decomposing the System.*” *Software Engineering* Nov. 2020. FMV Isık University, PowerPoint Presentation.