

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**ARCHITECTURAL DESIGN SPECIFICATION  
CSE 4316: SENIOR DESIGN I  
SUMMER 2019**



**TEAM MINTS  
UTA ADVISING**

**TUFAN ACHARYA  
NAWARAJ BHURTEL  
SAMEER CHAULAGAIN  
MAHESHWOR RAUT  
ISHOR RIJAL**

## REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	07.30.2019	TA, IR	document creation
0.2	08.05.2019	NB, SC	complete draft
0.3	08.08.2019	MR, IR, TA, NB	release candidate 1

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>System Overview</b>	<b>6</b>
2.1	COMMUNICATION LAYER Description . . . . .	6
2.2	CONTROL LAYER Description . . . . .	7
2.3	DATA LAYER Description . . . . .	7
2.4	VIEW LAYER Description . . . . .	7
<b>3</b>	<b>Subsystem Definitions &amp; Data Flow</b>	<b>8</b>
<b>4</b>	<b>COMMUNICATION LAYER Subsystems</b>	<b>9</b>
4.1	LOGIN . . . . .	9
4.2	AUTHENTICATION . . . . .	10
4.3	FILTER . . . . .	10
4.4	Response . . . . .	11
<b>5</b>	<b>CONTROL LAYER Subsystems</b>	<b>12</b>
5.1	VALIDATION . . . . .	12
5.2	ROLES . . . . .	13
<b>6</b>	<b>DATA LAYER Subsystems</b>	<b>14</b>
6.1	STORAGE . . . . .	14
6.2	DATA CONTROL . . . . .	15
<b>7</b>	<b>VIEW Layer Subsystems</b>	<b>16</b>
7.1	INTERACTIONS . . . . .	16
7.2	GRAPHICAL USER INTERFACE . . . . .	17

## LIST OF FIGURES

1	System Overview for UTA Advising . . . . .	6
2	Data-flow of UTA Advising . . . . .	8
3	Communication Layer Subsystem . . . . .	9
4	Control layer Subsystem . . . . .	12
5	Data layer Subsystem . . . . .	14
6	Example subsystem description diagram . . . . .	16

## LIST OF TABLES

2	Subsystem interfaces . . . . .	10
3	Subsystem interfaces . . . . .	10
4	Subsystem interfaces . . . . .	11
5	Subsystem interfaces . . . . .	11
6	Subsystem interfaces . . . . .	13
7	Subsystem interfaces . . . . .	13
8	Subsystem interfaces . . . . .	15
9	Subsystem interfaces . . . . .	15
10	Subsystem interfaces . . . . .	16
11	Subsystem interfaces . . . . .	17

## 1 INTRODUCTION

The advising applications that we are targeting as of now is going to be a web application. The application will have three end points each for the advisor login, student login and another for the admin/developer to track the issues and the changes that we are going to implement.

The first and the most important page is for the students who are targeting to schedule their advising with one of the advisors. The advisors could either faculty advisors or the staff. The students who are targeting to schedule an appointment can simply input their basic details like UTA ID, majors and the classes they have taken along with the semester for which they are looking to get advised. The students will also be able to choose particular advisors like faculty/staff if they are available and can put their name in the respective advisors wait list.

The second end point would be for the advisors. Advisors could see the list of the students waiting to get advised. Also the advisors will be able to see the UTA ID of the respective students and will also have the form already filled by the students waiting to get advised. The form includes the basic details necessary for the advisors like major information, classes taken, classes they are planning to take in the respective semester. This way they will be able to advise the students more effectively. We will have the databases that stores all the data of the students that have been advised and store their login credentials in cases they came back for advising in near future.

We will be making the advising process completely online by signing up students online and getting rid of the paper based sign-up sheet. The students will also be able to choose the advisor who advised them in the past or pick a new advisor every time they make appointment. The advisor will also have a very easy interface where they can see who is coming next and if they are new or someone who was already advised by them in the past. The advisors will also be able to re-schedule the appointments and change schedule which can be seen by the students before making appointment.

Our system will be very simple to use by any new or existing user because we will be using GUIs and figures to represent options and features. Everything happening in the back-end is secure and safe which takes care of user's privacy.

## 2 SYSTEM OVERVIEW

The advising system is structured as traditional dynamic web application which would be primarily be decisive based on the communication layer, control layer, data layer and view layer. These layers would function highly individually as well as interact with each other to perform the required functionality for the system. The communication layer would perform the role of bridge to connect between the view layer and control layer, whereas data layer would be helping to handle the data storage and access. Even though data layer might be functional in data access, control layer would be highly administrative in term of roles and data access process.

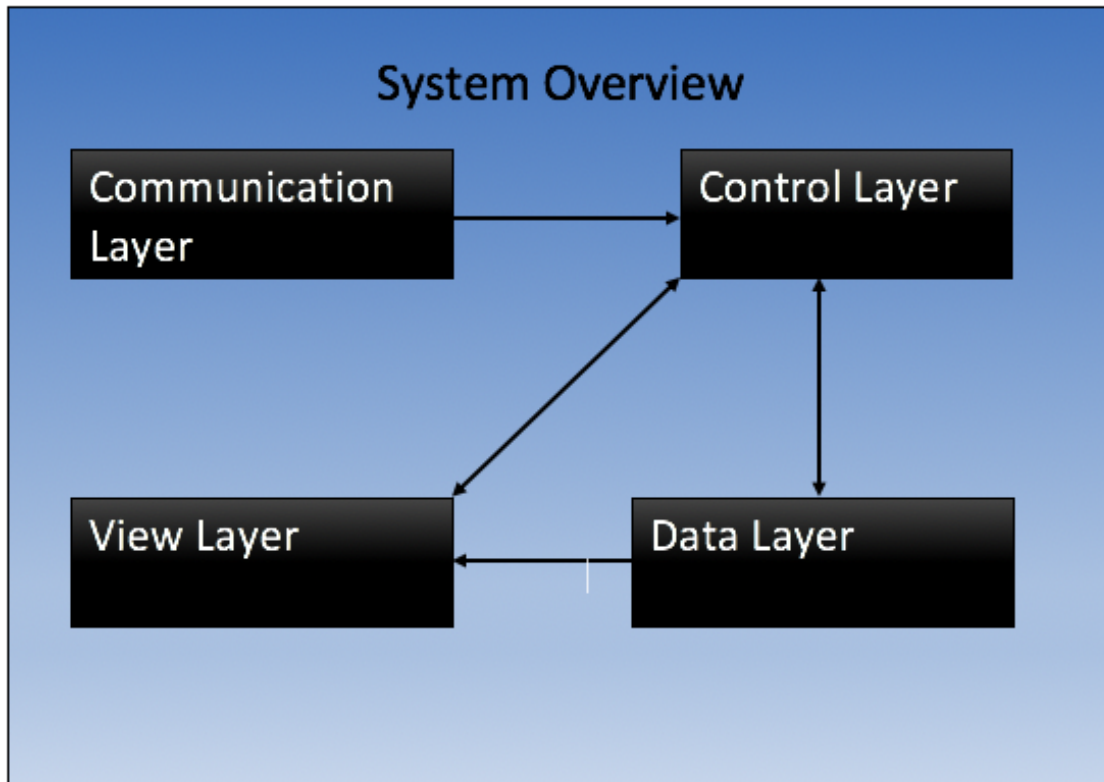


Figure 1: System Overview for UTA Advising

### 2.1 COMMUNICATION LAYER DESCRIPTION

This layer in the system act as a bridge between the control layer and view layer indirectly whereas how this layer acts determines the response to the data layer as well. This layer will be highly functional in sending the correct, legal, filtered and requested data from user to the control layer so that manipulation required for the application would be processed to the view layer of the program. This layer will primarily take care of the data taken from the user to log-in to the system and check the correctness of data, and accurateness to the requests by the user, and hence send the response to the control layer for further speculations.

## **2.2 CONTROL LAYER DESCRIPTION**

Control layer is primarily functional for controlling the administrative part of the application. This layer would be hence a main part to define the types of data that could be processed, and roles of the user would also be controlled by this layer. This layer would control the administration by labeling the users as a student or administrator or advisor and send the required authoritative power to the data layer or the view layer as required. These layer would also set the standards for the types of actions and data accepted by the application.

## **2.3 DATA LAYER DESCRIPTION**

This layer is primarily meant for the data storage and help the control and interaction process to exchange the data required for the functions. After the requests has been processed by the roles set up the the control layer to the data layer, the data layer would help in storing the data store in the server. The data would be available for access for the control layer and view layer through the data control mechanism which would essentially evaluate the necessity and authenticity of the need of the data.

## **2.4 VIEW LAYER DESCRIPTION**

This layer is the front end of the application. This layer is very functional on helping the user interact with the application. This layer will be helping send user interactions to the control layer directly and also the communication layer indirectly. This layer will be present the actions requested by the user to user directly, primarily this is the only layer that would be visible to the user. This layer would provide the notifications, requests, and responses in general.

### 3 SUBSYSTEM DEFINITIONS & DATA FLOW

The four layer of the UTA advising web application work together to successfully schedule appointment with the advisors. The communication layer and the Control layer work together for the authentication and the data validation from the data layer. Whenever a user enters the system the Communication layer authenticates the login information and the control layer works together with the data layer to validates the login information stored in the database. Data layers consists of the databases of the wait times, login and some more documents used in the application. All the four layers work together from view layer passing to the communication and control layer and accessing the data from the data layer.

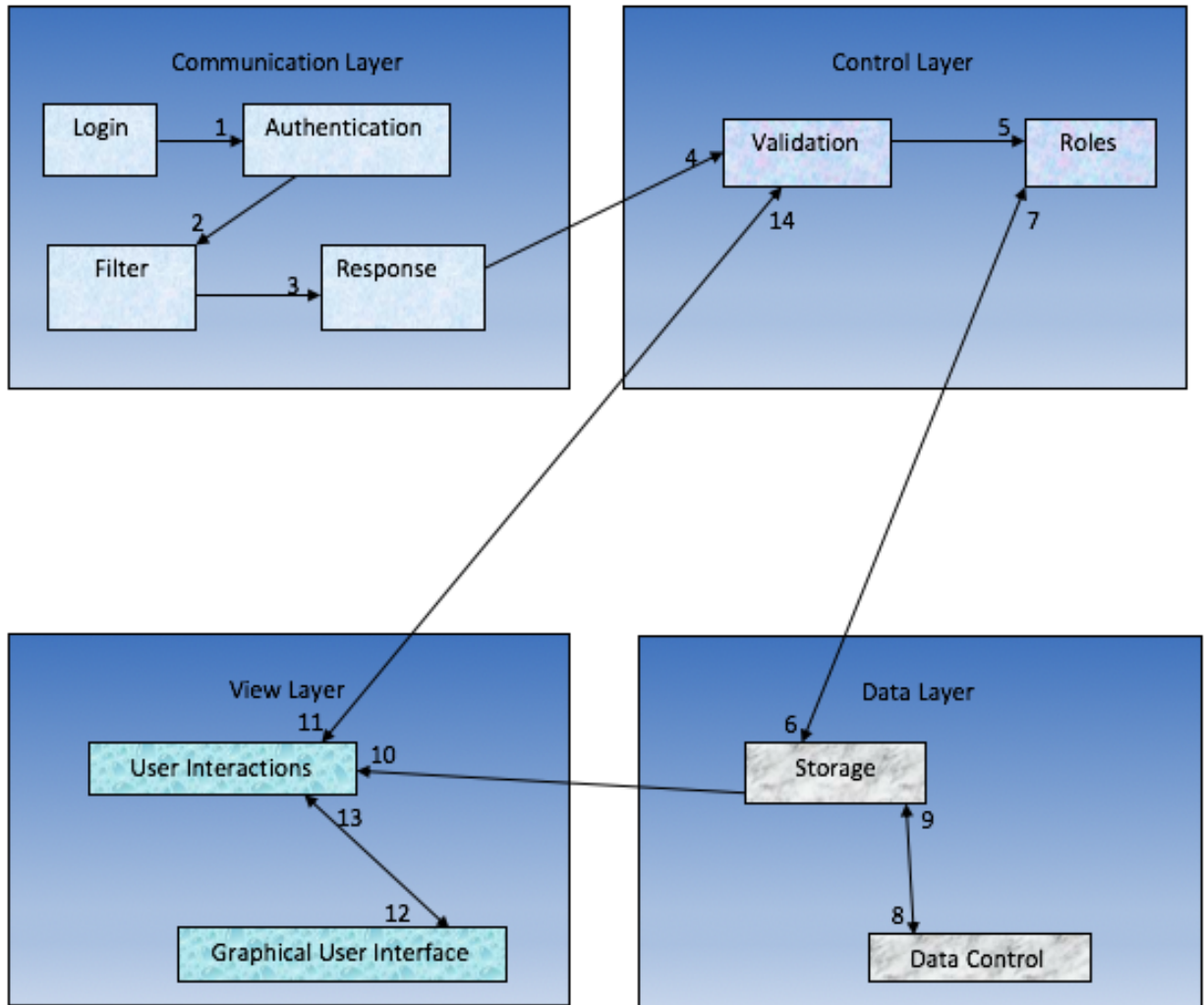


Figure 2: Data-flow of UTA Advising



## 4 COMMUNICATION LAYER SUBSYSTEMS

Communication layer is the first layer of the subsystem which is responsible for Login, Authentication, Filter and Sending Response. For this layer of sub-system the system will first get the login credentials from the user in the form of keyboard input from a computer or a cellular device. The authentication system will check what kind of data is coming and what kind of data is being requested. It also makes sure that the data coming in is in correct format. After that the Filter will analyze what type of data is being requested.

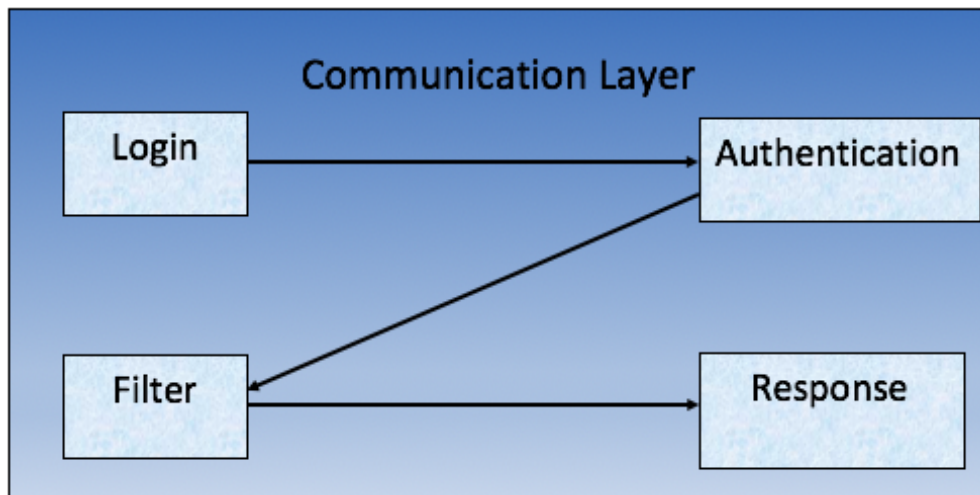


Figure 3: Communication Layer Subsystem

### 4.1 LOGIN

This section is responsible for receiving inputs from the users to send the data to Authentication system. There are two ways for logging in to the system. The advisor can login with administrator rights whereas the students will be able to login as users with some limited rights. Based on the login credentials provided our system will direct the users to their pages.

#### 4.1.1 ASSUMPTIONS

The user id and password provided by the users are correct.

#### 4.1.2 RESPONSIBILITIES

The login subsystem will send the data received from the user to Authentication system. The authentication subsystem analyze what kind of data is coming in and based on that the system will respond. If a student logs in and does not want to make an appointment, the student's name will not be in the appointment list. This subsystem will stop the system to send any useless information to and about the user.

### 4.1.3 SUBSYSTEM INTERFACES

Table 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Advisor's Login	Username/Password	Admin Screen
#01	Student's login	Username/Password	Student Interface

## 4.2 AUTHENTICATION

This subsystem checks if the format of the data entered is valid. We have our own standard set for authentication so this system will not interact with database for validity. The authentication will also make sure that fields and requests are correct according to the system. There are certain standard for the authentication process.

### 4.2.1 RESPONSIBILITIES

The primary responsibility will be checking the data validity. For example any user is not allowed to use special characters for user name but special characters are allowed for passwords, so this subsystem will make sure that anything passed by user input is in the correct form.

### 4.2.2 SUBSYSTEM INTERFACES

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#001	Correct Format	User1 / Password!	Success
#002	Incorrect format	User1!/Password	Incorrect format

## 4.3 FILTER

This subsystem mainly focuses on limiting the amount of data is being sent by sending only required data by filtering.

### 4.3.1 ASSUMPTIONS

Any data providing extra information wont be send unless requested.

### 4.3.2 RESPONSIBILITIES

The main responsibility of this subsystem is to filter the type of input coming in and send the output based on the input type. Example: Logging in will only login and not set appointment unless clicked on set appointment button. This will not confuse user as well as the system performance will be improved.

### 4.3.3 SUBSYSTEM INTERFACES

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#03	Enter username and password	Username/Password	Logged in Only

## 4.4 RESPONSE

This subsystem deals with what to respond with. Based on the filtered input the response will be sent out. This is the end point of the Communication layer. After sending the response Communication layer interacts with control layer for validation.

### 4.4.1 ASSUMPTIONS

Response will be only provided if requested and the the response should also be optimized based on the request.

### 4.4.2 RESPONSIBILITIES

The system should be able to provide the proper response to the request layer from view layer to the control layer. It should also be considerate about the response time.

### 4.4.3 SUBSYSTEM INTERFACES

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#04	Request/Response	Filtered data/Request from view layer	Accurate Response to the control layer

## 5 CONTROL LAYER SUBSYSTEMS

Control layer is a heart of the application that we are building. In this section we discuss about various subsystems which assemble a whole control layer. Control layer is basically responsible for the validation and authentication of the two main users, advisors and the students.

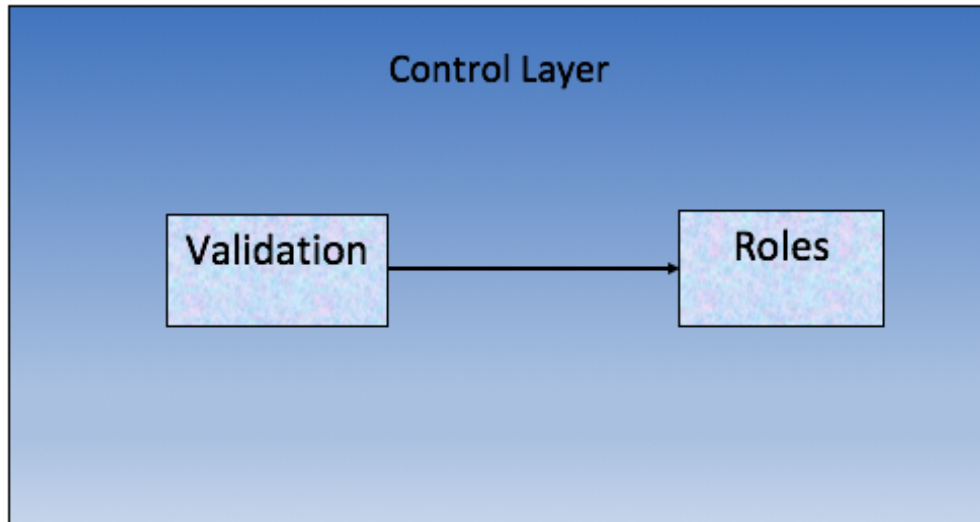


Figure 4: Control layer Subsystem

### 5.1 VALIDATION

Validation basically validates all the login information entered by the users. After the communication layer validates the requirements for the user name and password, the validation in the control layer validated who is entering the systems advisors or the students. Then the system use the database to check who is entering the system.

#### 5.1.1 ASSUMPTIONS

In the validation we assume that the communication validates the username and the password requirements and the system directs the user to their proper view page.

#### 5.1.2 RESPONSIBILITIES

After the communication layer validates the necessary requirements and we have the users entering the correcting information validation checks the username and password match to the database. If the validation API found the data in the database it passes the application with the new view page for each users. For example if the entered user name is of administrator or advisors it check the username in the username table in the database and if it found the advisor user name it returns respective page to the users.

#### 5.1.3 SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 6: Subsystem interfaces

ID	Description	Inputs	Outputs
#001	Validates the username and password	user name password	new view page

## 5.2 ROLES

This section covers the roles subsystem that is in our controller system. We mostly have three roles of people, students, advisors and the administrator panel. After validating we can direct the users to their individual user page.

### 5.2.1 ASSUMPTIONS

We assume that the user name and the password entered by the users correctly matches either three of the roles of our systems.

### 5.2.2 RESPONSIBILITIES

After the validation the users can be either the students, advisors or the administrator. Students can put their email as a user name and the validator APIs verify it with the database while the advisors and administrator can have the admin access and see all the information entered by the students. After the validation and verification of the roles the individual UI pages of the users are shown and the data collection for the further process is done.

### 5.2.3 SUBSYSTEM INTERFACES

Roles contain all the users in the database and the validator validates the user name and password.

Table 7: Subsystem interfaces

ID	Description	Inputs	Outputs
#001	User name and Password Verification	user name and password	User Interface Page
#002	User Pages	student email and password advisor email and password	student page advisor page

## 6 DATA LAYER SUBSYSTEMS

Data Layer is one of the most important part of the entire system. It contains all the databases and the persistence layer to access the data from the databases. We have Data Control and the Storage in the data layer subsystems.

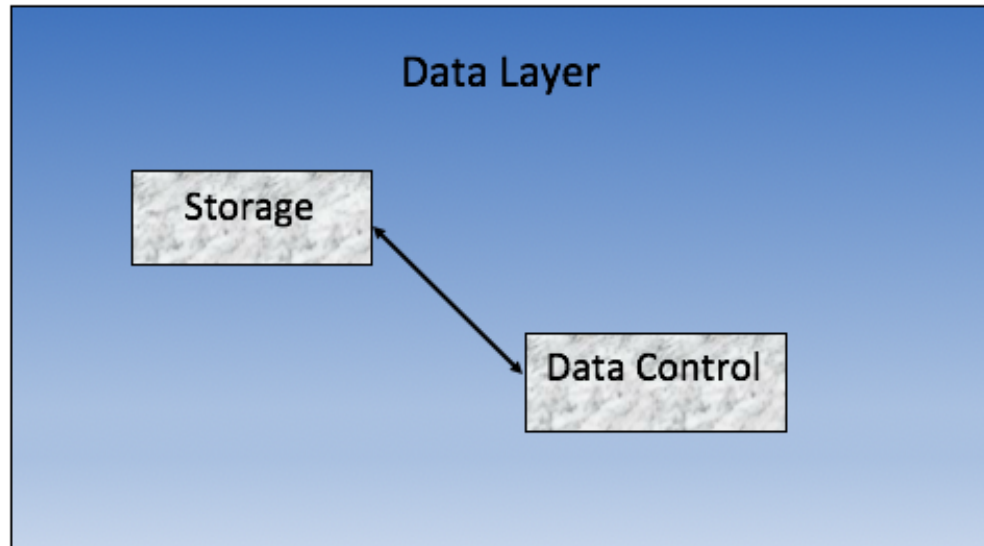


Figure 5: Data layer Subsystem

### 6.1 STORAGE

Storage is a databases that we are using. For this particular web application we are using the MYSQL databases for the storage. We have various tables, figures, encrypted password in the storage. Therefore the proper implementation and the security aspects are given priority.

#### 6.1.1 ASSUMPTIONS

We will be using Relational Databases like MYSQL and also assume that the data we stored will be secure.

#### 6.1.2 RESPONSIBILITIES

Storage is responsible to stores all the data that is coming from and to our network. From the students details like First Name, Last Name, UTA ID, Email and many other important information are stored in one table by using email or accessing a UUID to a particular users. Another table will have the hyperlink of the necessary documents and the advisors scheduled. We will also have a table to store the user name and the passowrd in encrypted form to access in the future.

#### 6.1.3 SUBSYSTEM INTERFACES

Since this is a storage we have various data inputs and outputs constantly coming.

Table 8: Subsystem interfaces

ID	Description	Inputs	Outputs
#001	User Details	First Name Last Name Email	Stored in the databases
#002	Request for the data	Data request end points	All the necessary data request's response.

## 6.2 DATA CONTROL

Data Control is in the system where we manipulates the data. This layer works as a data persistence layer. In this layer we will have a bunch of APIs to work on the GET,POST, CREATE and DELETE the data from the databases. This layer will have a data connection API/library like JDBC or Hibernate.

### 6.2.1 ASSUMPTIONS

The data connection is established successfully every time.

### 6.2.2 RESPONSIBILITIES

This layer is responsible to access the data from our User Interface Page. Whenever the users enter anything it is stored in our database in the form of object. The stored data is then accessed by the advisors and the algorithm to calculate the best estimated time.

### 6.2.3 SUBSYSTEM INTERFACES

We will have various APIs waiting here for the data mapping. Input are the data that wants to be entered to the databases and the output is the data requested by the system and users.

Table 9: Subsystem interfaces

ID	Description	Inputs	Outputs
#001	Data Request and Response	user data	Database storage

## 7 VIEW LAYER SUBSYSTEMS

View layer is the front end of the application. It consists of the two subsections, subsection interactions generally handling about specific responses and requests, and the subsections graphical user interactions which handles how the application looks, and all other actions that user want to perform in the system.

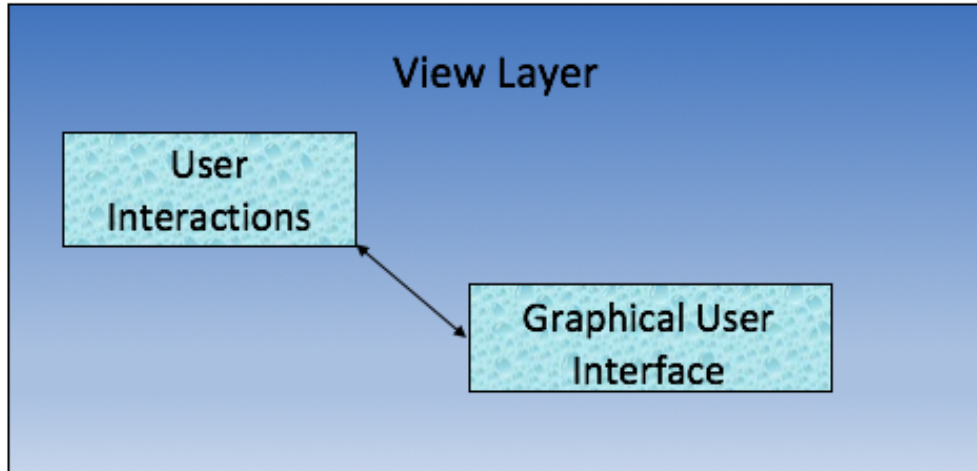


Figure 6: Example subsystem description diagram

### 7.1 INTERACTIONS

Interactions subsystem will handle the specific functional requests and responses requested by the system or the user and help them to redirect to the interfaces for the display or the control layer to perform the action based on the need of the system. The interactions will be primarily focused on adding check-in requests, notifying user about their queue time and other higher priority actions.

#### 7.1.1 ASSUMPTIONS

Even though user interface controls the interactions, this sub system is especially designed for the higher priority interactions to optimize the results.

#### 7.1.2 RESPONSIBILITIES

This subsystem should be able to prioritize the importance of the actions coming requests and process it accordingly.

#### 7.1.3 SUBSYSTEM INTERFACES

Table 10: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Higher Priority Request	Check In Request Advising Complete Notice	Quick Pop Notice to User
#02	Effective Notice	Request Complete	Instant Notice



## 7.2 GRAPHICAL USER INTERFACE

This the main part of the web application of the advising system. This is where users would be directly interacting with the system to make all the requests and see the changes to it. Any functions that they want to perform would be requested in this sub system and results will be displayed in this subsystems as well.

### 7.2.1 ASSUMPTIONS

User would be know to navigate the different features of website based on personal experience and the help provided.

### 7.2.2 RESPONSIBILITIES

This subsystems should be able to help the user to perform any possible functionality of the system. It should be simple and easy to use.

### 7.2.3 SUBSYSTEM INTERFACES

Table 11: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Generalized Interface	User Requests Page Requests	Responses Page Display

## REFERENCES