

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

**PROJECT CHARTER
CSE 4316: SENIOR DESIGN I
SUMMER 2020**



**THE BREW CREW
BEVERAGE MANAGEMENT**

**BISHAL PAUDEL
NIRJAL PHAIJU
SIMA RAYMAJHI
LOKENDRA B. CHHETRI
KUNAL SAMANT**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	08.10.2020	BP	document creation
0.2	08.11.2020	BP	Added architectural layer diagram and section 2
0.3	10.12.2015	AT, GH	release candidate 1
1.0	10.20.2015	AT, GH, CB	official release
1.1	10.31.2015	AL	added design review requests

CONTENTS

1	Introduction	5
2	System Overview	6
2.1	Presentation Layer Description	6
2.2	Application Layer Description	6
2.3	Database Access Layer Description	6
3	Subsystem Definitions & Data Flow	8
4	Presentation Layer	9
4.1	Request	9
4.2	XML	10
4.3	Response	11
5	Application Layer	12
5.1	API Subsystem	12
6	Z Layer Subsystems	13
6.1	Subsystem 1	13
6.2	Subsystem 2	13
6.3	Subsystem 3	14

LIST OF FIGURES

1	A simple architectural layer diagram	6
2	A simple data flow diagram	8
3	Example subsystem description diagram	9
4	Example subsystem description diagram	10
5	Example subsystem description diagram	11
6	Example subsystem description diagram	12
7	Example subsystem description diagram	13

LIST OF TABLES

2	Subsystem interfaces	10
3	Subsystem interfaces	11
4	Subsystem interfaces	11
5	Subsystem interfaces	12
6	Subsystem interfaces	13

1 INTRODUCTION

Your introduction should describe your product concept in sufficient detail that the architectural design will be easy to follow. The introduction may include information used in the first sections of your SRS for this purpose. At a minimum, ensure that the product concept, scope and key requirements are described.

2 SYSTEM OVERVIEW

The diagram (Figure 1) below shows the basic architectural layer diagram of the Beverage Management app. The overall structure of our app can be described using the popular three-layer architecture which consists of presentation layer, application layer and data access layer. The presentation layer is the top-most layer of our system which allows user to interact with the system. Application layer acts as an interface between the presentation layer and data access layer. This layer supports all of the core functions of our application. The data access layer is the layer where all the data and information are stored or retrieved from the database. In other words, the presentation layer takes input from the user and pass it to the application layer. The application layer then process those commands and pass the information to the data access layer. The data access layer either store the information on the database or retrieve the requested information from the database and pass it back to the application layer, and eventually to the presentation layer where the result is displayed in a user understandable format.

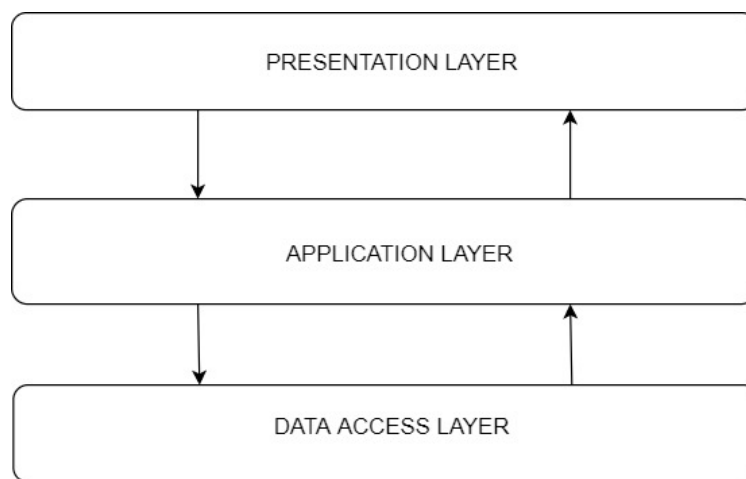


Figure 1: A simple architectural layer diagram

2.1 PRESENTATION LAYER DESCRIPTION

This layer will allow our application to successfully communicate with the user. The features will include the display of login page, page for storing new products in the inventory, and also displaying the desired output for the user. This user level layer is connected with our application layer which will help to display or retrieve the required information that our user is looking for.

2.2 APPLICATION LAYER DESCRIPTION

This layer will serve as a bridge between the presentation layer and data access layer of our application. No matter what command the user gives in the presentation layer level, these commands will be interpreted by the application layer. Depending on the type of command, this layer will execute the command by accessing the database layer and then channels the accurate and expected output to the presentation layer.

2.3 DATABASE ACCESS LAYER DESCRIPTION

This layer is the most fragile yet the most critical aspect of our application. This is where the information from the user is stored so that it can be accessed when required in future. We will be using subsystem such as Firebase, SQLite and Global barcode inventory. This layer will be the fulcrum to our team creating an effective management system. Multiple information will be recorded in this layer. This

includes login credentials, a product's name, flavor, type and so on. In brief, this layer will execute the query given by the user in presentation layer and process the result back to application layer which will finally process the information to be displayed in the presentation layer.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

This section breaks down our layer abstraction to another level of detail. Altogether, our system is divided into 3 different layers, Presentation Layer, Application Layer, Data Access Layer. Each of these layers are further divided into multiple subsystems. The Presentation Layer has Request, XML and Response. The Application Layer is divided into Java, Database Connection and Data Access Query whereas the Data Access Layer consists of Firebase, SQLite and Global Barcode Inventory. When an user opens the app, they will see the presentation layer consisting of text, buttons, forms, etc made using XML. When user makes some request like checking whether "India pale ale" is present in the inventory, this layer communicates with the application layer which formats a query and passes it to the data access layer. The data access layer then runs this query and returns the results (list of all beverage matching that style) to the application layer which then converts these results into appropriate format so that it can be displayed in the presentation layer.

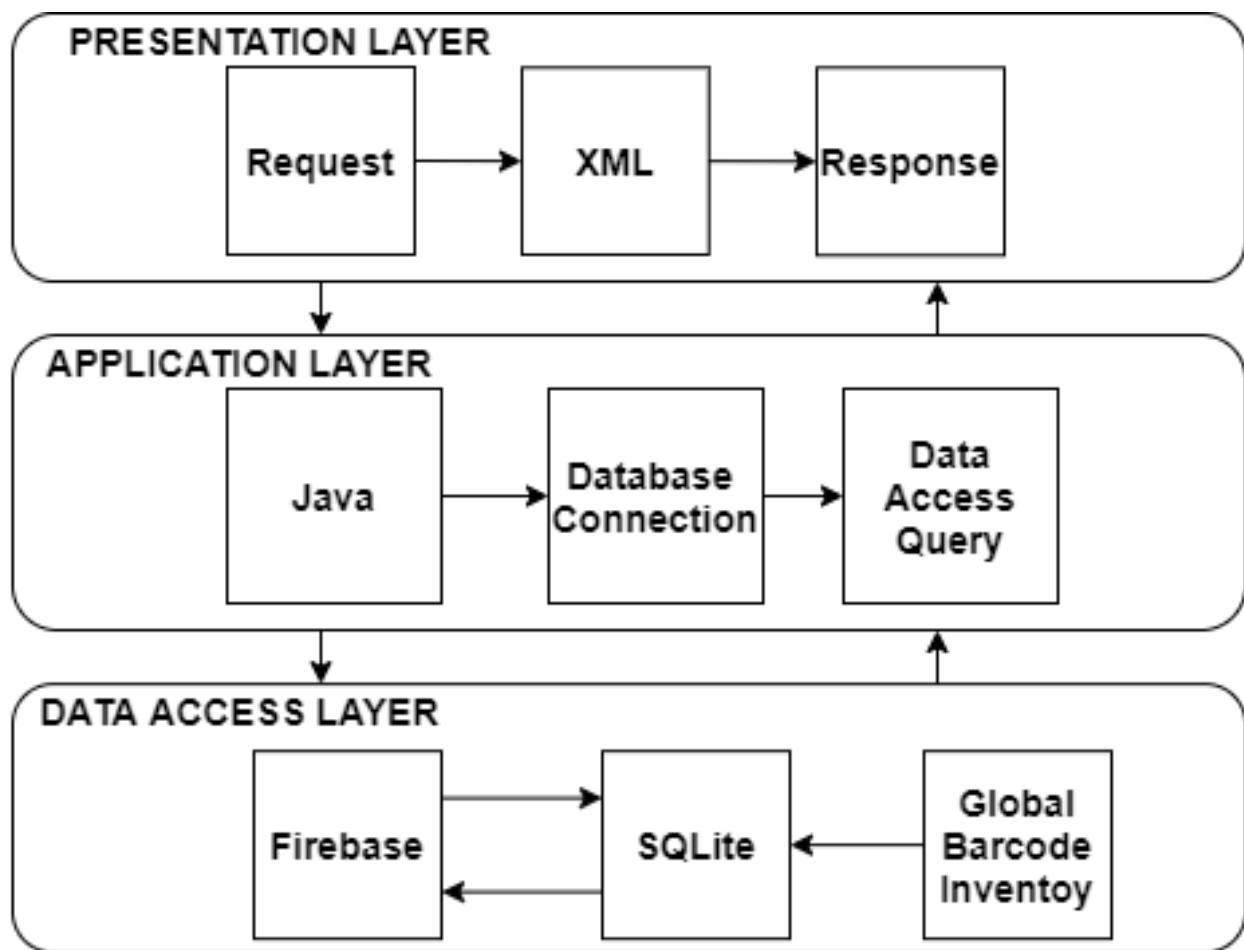


Figure 2: A simple data flow diagram

4 PRESENTATION LAYER

Presentation layer is where the interaction between human and machine takes place. This layer is further divided into 3 subsystems that continuously interact with each other to successfully communicate with the user. The screen of a mobile device will help the user to explore the various features of the application. Request subsystem is involved in receiving the commands of the user, XML subsystem will receive the command from Request subsystem. This subsystem will help to decode the instructions and complete the task. The response subsystem will display the result after the command is executed.

4.1 REQUEST

Request is a fundamental subsystem of the Presentation layer. This subsystem will help to process the user given command. The commands may vary; the variety includes, signing up for new account, logging in to an already existing account, add to item to inventory, delete items from inventory, and search the inventory.

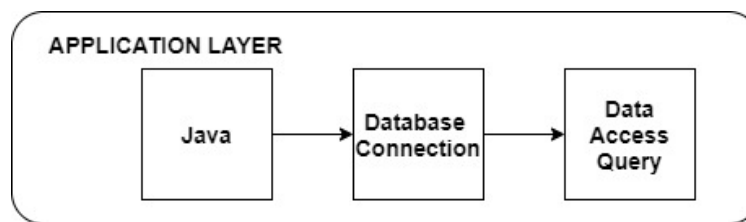


Figure 3: Example subsystem description diagram

4.1.1 ASSUMPTIONS

We assume that all the data entered by the user are accurate and valid for the application layer to process.

4.1.2 RESPONSIBILITIES

This subsystem will be responsible to process the instructions entered by the user to the application layer.

4.1.3 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 2: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	The request subsystem will interact with the user to understand what the wants from the application. This will include SignUp, Login, store information or retrieve information from database	Email ID, User name, password, Age, First Name and Last Name	Registered or denied
#2	The request subsystem will interact with the user to understand what the wants from the application. This will include Login.	username, password	login success or denied due to incorrect credentials

4.2 XML

XML is the unseen mechanics of our Presentation layer. The layouts, images or buttons seen in the application are functional because of XML. There are various kinds of XML files. Manifest XML file will help to define the functionality of buttons, layout XML file will help to determine the layout and many more XML files are available to help make the user-system interaction easy.

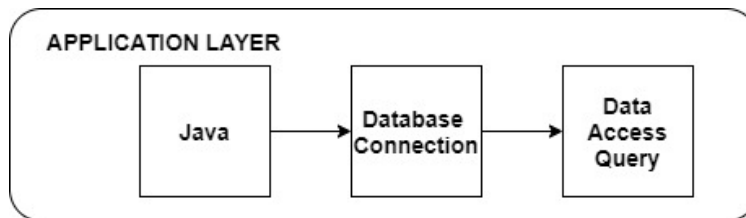


Figure 4: Example subsystem description diagram

4.2.1 ASSUMPTIONS

The user will issue right commands.

4.2.2 RESPONSIBILITIES

The main responsibility of this subsystem is to form a connection with application layer from the presentation layer depending on what the user wants to do.

4.2.3 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 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	User wants to search for an item(s) already in the inventory	Items to search by name, expiry date, flavor	Lists of product that match the search criteria.

4.3 RESPONSE

This subsystem is essential in order to successfully display the output that is asked by the user. The information gained from the request layer is processed by the application layer using database access layer if necessary in order to produce the desired out for the response subsystem.

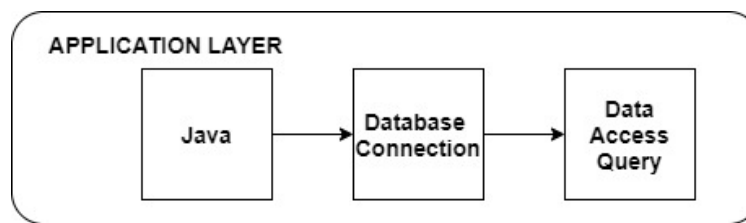


Figure 5: Example subsystem description diagram

4.3.1 ASSUMPTIONS

We assume that the query made by user can be successfully executed by application layer.

4.3.2 RESPONSIBILITIES

This subsystem is responsible to display the result that is required from the users query.

4.3.3 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 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	The user wants to look in to inventory for a certain product.	Name of the product or search by specific requirement.	Detailed description of the product.

5 APPLICATION LAYER

This layer processes the data scanned from the presentation layer and stores the data in local variables, i.e. name of the beverage and expiry date. This data is added to the current user's beverage database and send the updated data back to the presentation layer for further input.

5.1 API SUBSYSTEM

The API subsystem passes the barcode to the Data Access Layer and retrieves the information of the beverage being scanned. The data retrieved is added to the current dataset of beverages which the user owns.

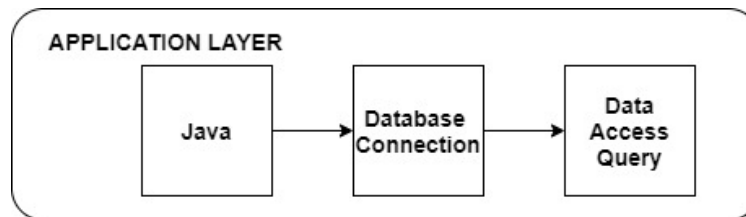


Figure 6: Example subsystem description diagram

5.1.1 ASSUMPTIONS

- The drink is in the database of drinks.
- The data inputted is correctly scanned.

5.1.2 RESPONSIBILITIES

The data must be properly retrieved and stored to avoid glitches in the application and ensure that the user is satisfied.

5.1.3 SUBSYSTEM INTERFACES

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Application to Data Access	Scanned Barcode ID	Product ID
#xx	Application to Presentation	Data	Data in readable format

6 Z LAYER SUBSYSTEMS

In this section, the layer is described in some detail in terms of its specific subsystems. Describe each of the layers and its subsystems in a separate chapter/major subsection of this document. The content of each subsystem description should be similar. Include in this section any special considerations and/or trade-offs considered for the approach you have chosen.

6.1 SUBSYSTEM 1

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

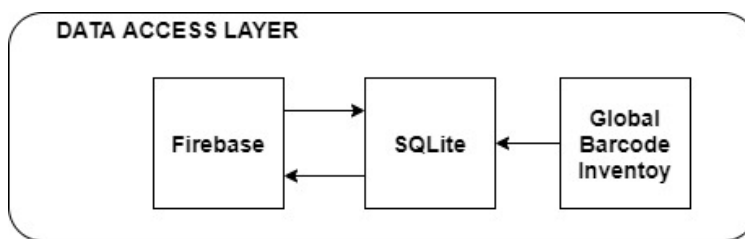


Figure 7: Example subsystem description diagram

6.1.1 ASSUMPTIONS

Any assumptions made in the definition of the subsystem should be listed and described. Pay particular attention to assumptions concerning interfaces and interactions with other layers.

6.1.2 RESPONSIBILITIES

Each of the responsibilities/features/functions/services of the subsystem as identified in the architectural summary must be expanded to more detailed responsibilities. These responsibilities form the basis for the identification of the finer-grained responsibilities of the layer's internal subsystems. Clearly describe what each subsystem does.

6.1.3 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
#xx input 2	Description of the interface/bus 3cmoutput 1	3cminput 1	
#xx	Description of the interface/bus	3cmN/A	3cmoutput 1

6.2 SUBSYSTEM 2

Repeat for each subsystem

6.3 SUBSYSTEM 3

Repeat for each subsystem