Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
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	09.18.2020	BP	document creation
0.2	09.18.2020	BP	Added architectural layer diagram and section 2
0.3	09.20.2020	SR	Added section 1
0.4	09.24.2020	LC	Working on section 6
0.5	09.24.2020	NP	Added section 4
0.6	09.24.2020	KS	Added section 5
0.7	09.24.2020	SR	All sections Revision
1.0	09.24.2020	BP	Official Release

CONTENTS

List of Figures

List of Tables

1 Introduction

Beverage Management is an Android OS based application designed to manage the beverages in the household environment with significantly large collection in multiple setups. The product will virtually manage the collection allowing the user to store the beverages in different locations with customized racks.

Main intended purpose including keeping track of name, storage location, brewery, style, volume, manufactured date, best before date, and some other functionality including search and sort the beverage according to the date, style, etc.

Additional product concept includes the bar-code scanning capability, which ensure the initial setup of the inventory, and saving it the local data-set,a friendly user interface to keep it sleek and notification system to notify the user about certain important information.

Conclusively, the initial application release will on be Android platform, however the team will also be working on web platform and will be released in future releases.

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.

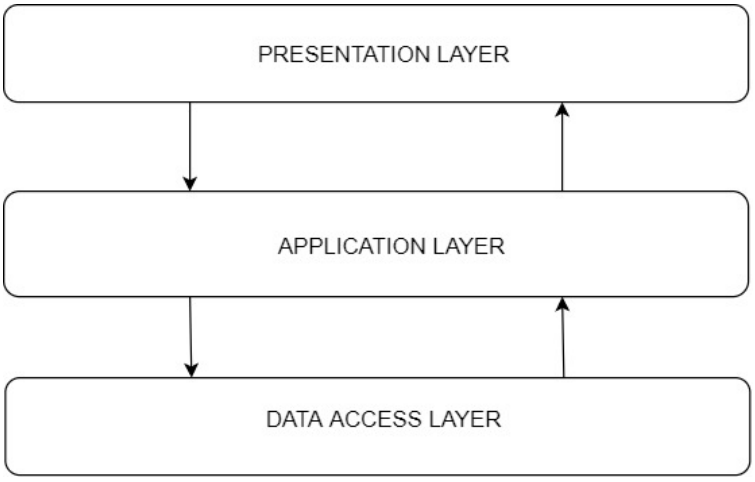

Figure 1: A simple architectural layer diagram

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 Data 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 subsystems such as Firebase Authentication to manage user's authentication, Firebase Cloud Firestore to store all the information about the beverage, Firebase Storage to store photos and Global Barcode Inventory to retrive infromation about the beverage using it's barcode number. This layer will be the fulcrum to our team creating and effective management system. Multiple information will be recorded in this layer. This includes login credentials, a productâs name, style, 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 Presentation Layer

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.

3.1 Layer Hardware

Since, we are developing an android application, our application will need a cellphone to operate. Apart from that, our application will receive and display information via the cellphone screen.

3.2 Layer Operating System

Our application will run efficiently on android version higher than Android 4.0.3(IceCreamSandwich).

3.3 Layer Software Dependencies

Not Applicable

3.4 Request

Request is a fundamental subsystem of the Presentation layer. This subsystem will help to process the userâs 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.

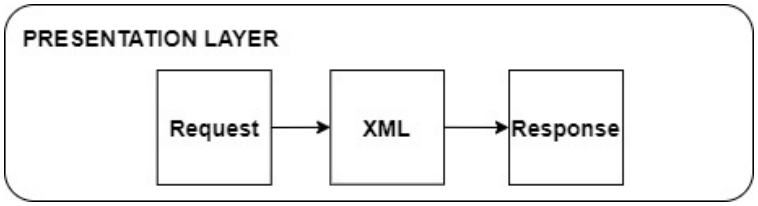

Figure 2: Example subsystem description diagram

3.4.1 Subsystem Hardware

Cellphone Camera that will help the user to click pictures of the inventory they are storing.

3.4.2 Subsystem Operating System

Any

3.4.3 Subsystem Software Dependencies

Not Applicable

3.4.4 Subsystem Programming Languages

Java and Android Studio

3.4.5 Subsystem Data Structures

Not applicable

3.4.6 Subsystem Data Processing

Not applicable

3.5 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.

3.5.1 Subsystem Hardware

Not applicable

3.5.2 Subsystem Operating System

Any

3.5.3 Subsystem Software Dependencies

V7 Support Libraries

3.5.4 Subsystem Programming Languages

Java and Android Studio

3.5.5 Subsystem Data Structures

Not applicable

3.5.6 Subsystem Data Processing

Firbase adapter was used proccess the data from real time Database to Response Subsystem.

3.6 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.

3.6.1 Subsystem Hardware

Not applicable

3.6.2 Subsystem Operating System

Any

3.6.3 Subsystem Software Dependencies

V7 Support Libraries

3.6.4 Subsystem Programming Languages

Java and Android Studio

3.6.5 Subsystem Data Structures

Not applicable

3.6.6 Subsystem Data Processing

Recycler View and Card-based layout were used to process the data and display the out put to the user.

4 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 userâs beverage database and send the updated data back to the presentation layer for further input. There is no use of hardware in this layer.

4.1 Layer Hardware

No Hardware is used.

4.2 Layer Operating System

Any OS is acceptable.

4.3 Layer Software Dependencies

The software must be able to fetch and catch API calls.

4.4 Java Subsystem

The Java Subsystem is responsible for taking the XML data provided from the Presentation layer and and send it to the API subsystem to connect to the dataset and accurately get the required information. After the data is retrieved from the dataset, this subsystem passes the result to the Response subsystem in the Presentation layer.

4.4.1 Subsystem Hardware

No hardware used in this subsystem.

4.4.2 Subsystem Operating System

Any OS is acceptable.

4.4.3 Subsystem Software Dependencies

The subsystem requires java to be installed as the only dependency to work properly.

4.4.4 Subsystem Programming Languages

The Java subsystem is written in Java.

4.5 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.

4.5.1 Subsystem Hardware

No hardware used in this subsystem.

4.5.2 Subsystem Operating System

Any OS is acceptable.

4.5.3 Subsystem Software Dependencies

The subsystem must be able to establish a connection to the database.

4.5.4 Subsystem Programming Languages

The API subsystem is written in Java.

4.6 Database Connection Subsystem

This subsystem establishes the connection between the database (Firebase) and the application.

4.6.1 Subsystem Hardware

No hardware used in this subsystem.

4.6.2 Subsystem Operating System

Any OS is acceptable.

4.6.3 Subsystem Software Dependencies

The subsystem must be able to establish a connection to the application.

4.6.4 Subsystem Programming Languages

The Database connection subsystem is written in Java.

4.7 Data Access Query Subsystem

This will fetch the data being query as per the API request and send the result back to the application.

4.7.1 Subsystem Hardware

No hardware used in this subsystem.

4.7.2 Subsystem Operating System

Any OS is acceptable.

4.7.3 Subsystem Software Dependencies

No Dependencies.

4.7.4 Subsystem Programming Languages

The Data access query subsystem is written in Java.

5 Z Layer Subsystems

In this section, the layer is described in terms of the hardware and software design. Specific implementation details, such as hardware components, programming languages, software dependencies, operating systems, etc. should be discussed. Any unnecessary items can be ommitted (for example, a pure software module without any specific hardware should not include a hardware subsection). The organization, titles, and content of the sections below can be modified as necessary for the project.

5.1 Layer Hardware

A description of any involved hardware components for the layer. For example, if each subsystem is a software process running on an embedded computer, discuss the specifics of that device here. Do not list a hardware component that only exists at the subsystem level (include it in the following sections).

5.2 Layer Operating System

A description of any operating systems required by the layer.

5.3 Layer Software Dependencies

A description of any software dependencies (libraries, frameworks, etc) required by the layer.

5.4 Subsystem 1

Descibe at a high level the purpose and basic design of this subsystem. Is it a piece of hardware, a class, a web service, or something else? Note that each of the subsystem items below are meant to be specific to that subystem and not a repeat of anything discussed above for the overall layer.

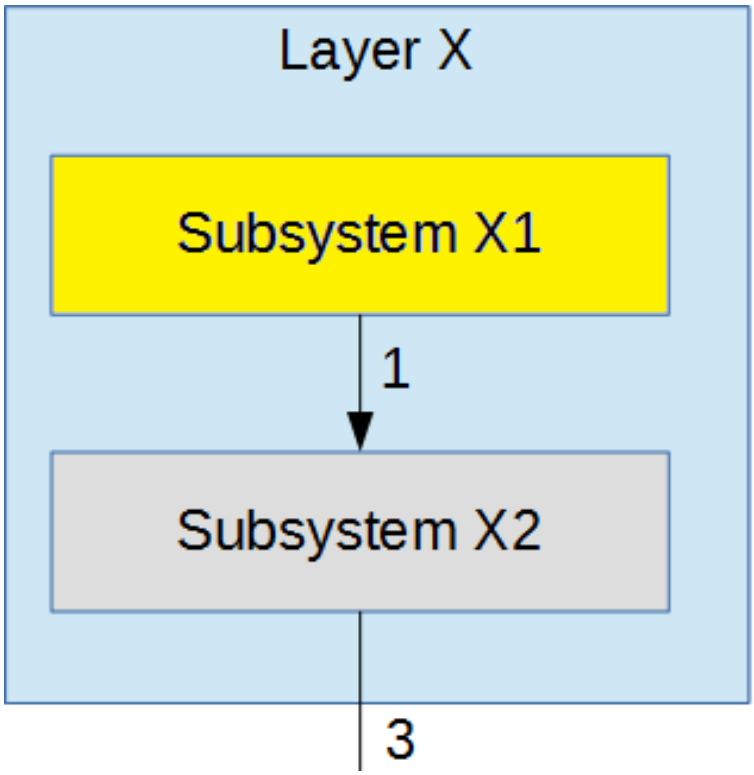

Figure 3: Example subsystem description diagram

5.4.1 Subsystem Hardware

A description of any involved hardware components for the subsystem.

5.4.2 Subsystem Operating System

A description of any operating systems required by the subsystem.

5.4.3 Subsystem Software Dependencies

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

5.4.4 Subsystem Programming Languages

A description of any programming languages used by the subsystem.

5.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

5.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

6 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

References