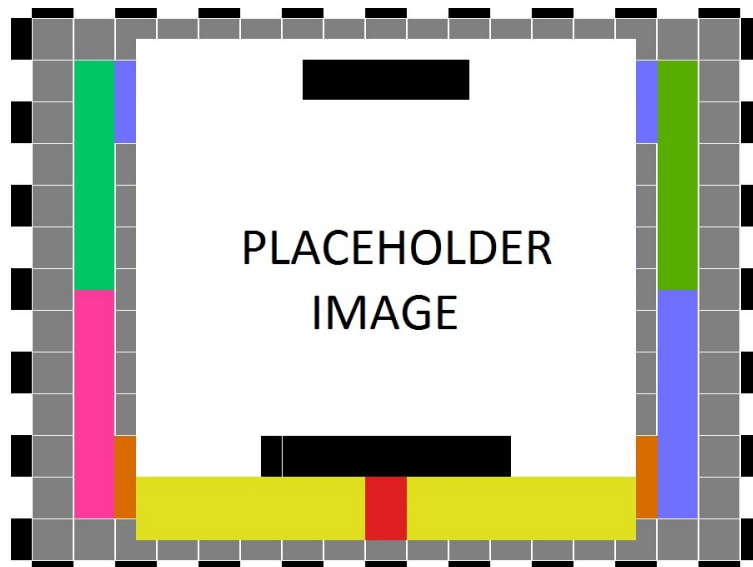


DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

ARCHITECTURAL DESIGN SPECIFICATION
CSE 4316: SENIOR DESIGN I
SUMMER 2019



ENGIMAVS
SWIFT SCAN

JACOB WILKINS
ALEX MEJIA
ZOHAR MORENO
SIRJAN KHANAL
UMANGA SHRESTHA

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.01.2015	GH	document creation
0.2	10.05.2015	AT, GH	complete draft
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 Mobile Application Layer Description 6
 - 2.2 Web Application Description 6
 - 2.3 Database Layer Description 6

- 3 Subsystem Definitions & Data Flow** **7**

- 4 X Layer Subsystems** **8**
 - 4.1 Manage Inventory 8
 - 4.2 Private Database Connector 9

- 5 Y Layer Subsystems** **12**

- 6 Z Layer Subsystems** **14**
 - 6.1 Subsystem 1 14
 - 6.2 Subsystem 2 15

LIST OF FIGURES

1	The architectural layer diagram of Swift Scan	6
2	Data Flow Diagram of the Swift system	7
3	Example subsystem description diagram	8
4	Example subsystem description diagram	10
5	Example subsystem description diagram	14

LIST OF TABLES

2	Manage Inventory Interfaces	9
3	Subsystem interfaces	11
4	Subsystem interfaces	14
5	Subsystem interfaces	15

1 INTRODUCTION

Swift Scan is composed of 3 major components: a mobile application, web application, and database system.

The mobile app will contain 3 tabs: profile, scan, and saved. The profile tab will ensure that all user settings are saved and can be synced with the web app. The scan tab will allow users to scan products using the camera and view information about them. If they choose to save the product information, it will be stored in the saved tab. The saved tab lists every product save by the user. These products can be sorted by type of alcohol or any other category. The user can also make collections with products of their choice.

The web app will include a profile tab and saved tab with the same features as the mobile app. The information will be synced from the mobile app so the user can access their products anywhere.

The database system is the driving force of the project. When a user scans a product, the database will be queried for the barcode number. Information such as name, brand, price, ABV, type, country, and category will be returned the user. In addition, the database will store the information of all users and will be monitored by the maintenance team.

2 SYSTEM OVERVIEW

The product will have 3 layers: The mobile application layer, web application layer, and database layer. This section will describe what each of these layers do.

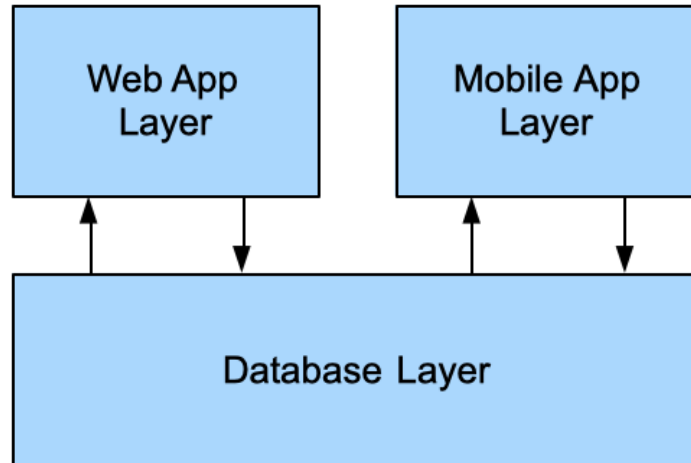


Figure 1: The architectural layer diagram of Swift Scan

2.1 MOBILE APPLICATION LAYER DESCRIPTION

The mobile app will contain 3 tabs: profile, scan, and saved. The profile tab will ensure that all user settings are saved and can be synced with the web app. The scan tab will allow users to scan products using the camera and view information about them. If they choose to save the product information, it will be stored in the saved tab. The saved tab lists every product save by the user. These products can be sorted by type of alcohol or any other category. The user can also make collections with products of their choice.

2.2 WEB APPLICATION DESCRIPTION

The web app will include a profile tab and saved tab with the same features as the mobile app. The information will be synced from the mobile app via the database system so that the user can access their products anywhere. When the web page is refreshed, a request will be sent to the database for the up-to-date information on the user.

2.3 DATABASE LAYER DESCRIPTION

The database system is the driving force of the project. When a user scans a product, the database will be queried for the barcode number. Information such as name, brand, price, ABV, type, country, and category will be returned the user. In addition, the database will store the information of all users and will be monitored by the maintenance team. To login, the user will enter their username and password which will be queried in the database to confirm their identity and all of their setting setting will be returned.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

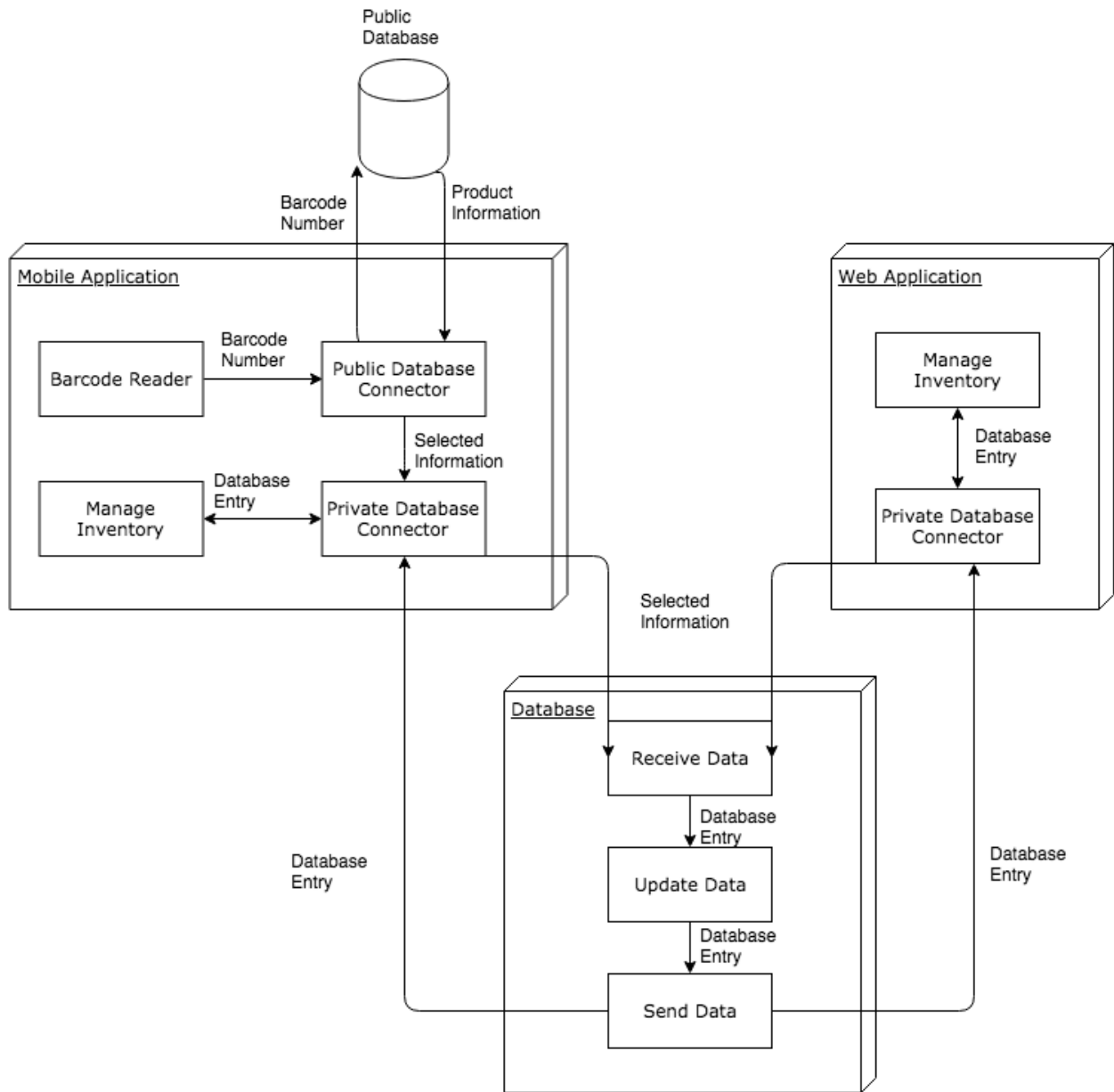


Figure 2: Data Flow Diagram of the Swift system

4 X LAYER SUBSYSTEMS

The web application, which in this case is Layer X, is the overall web portal that will be used to manage all information for any brewery products. The web application's sole purpose is to have administrative privileges and more control of the information that is stored in the brewery's database. Unlike the mobile application, the web portal will allow the user to add, remove, and modify any records of any brewery product that is stored in the database.

4.1 MANAGE INVENTORY

In the Manage Inventory subsystem of the web app layer, the subsystem will allow for information to be managed and stored into the brewery database. The user will be allowed to add, delete, or modify any brewery records stored in the database by having administrative privileges for easy management of the information stored in the database.

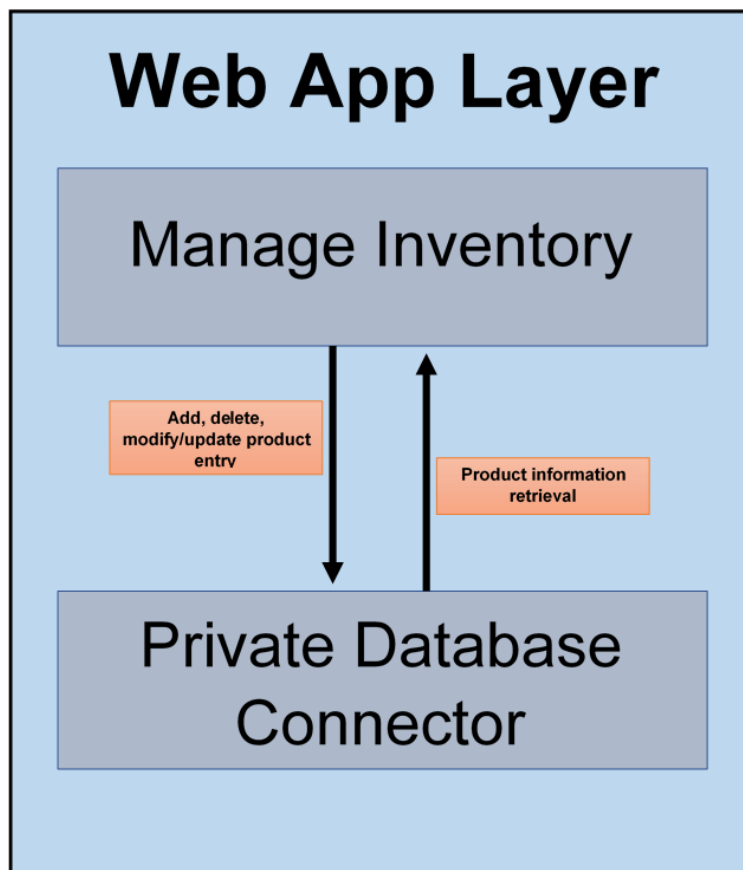


Figure 3: Example subsystem description diagram

4.1.1 ASSUMPTIONS

Assuming that a brewery product is not available on the public database when the mobile app scans the barcode of the brewery product, then the web portal will allow the user to add/input the product information into the brewery database manually and become available for retrieval. On the other hand, assuming the need for a product to be updated/modified arises, the web portal will allow the user to select the product from the database to be modified/updated and also will prompt the user which attribute of the product needs to be updated.

The Manage Inventory subsystem will interact directly with the central brewery database in the background. Any record that is added, deleted, or modified will be stored and saved in the database once the user approves it when using the web portal.

4.1.2 RESPONSIBILITIES

The Manage Inventory subsystem primary responsibility is for the user to interact with the user interface (UI) that'll be accessible through a web browser in order to manage any and all information stored in the central brewery database. The user will be able to add, delete, and modify any records that are currently stored and saved on the database. In addition, the information that'll be made available of any brewery product will include the following: Brewery Name of beer Style Date of purchase Best by date Format (can, bottle, keg, crowler, growler, etc.) Location (cabinet, closets, kegerator, etc.) ID

To further explain the format and location of the brewery, the user will be able to generate a two-dimensional figure to describe the location where the products are stored physically and set the location of the brewery product in the exact row and columns of the two-dimensional figure from the cabinet, closet, kegerator, etc.

4.1.3 SUBSYSTEM INTERFACES

Table 2: Manage Inventory Interfaces

ID	Description	Inputs	Outputs
#01	Web portal	brewery, name of beer style	expanded information on brewery product

4.2 PRIVATE DATABASE CONNECTOR

In the Private Database Connector subsystem of the web app layer, the connector will allow a direct and secure connection to the central brewery database for information to be stored in the database layer and for information to be retrieved from the database. The Private Database Connector will interface both with the Manage Inventory subsystem of the web portal and the central database layer.

4.2.1 ASSUMPTIONS

The Private Database Connector will establish an interface between the web portal and the central database. The connector will make sure that information between the web portal and the central database is exchanged accordingly. When the web portal requests for information on a particular brewery, the connector will make the request using MySQL queries to retrieve the information from the database. In turn, the database will receive the MySQL query and will return the information to the connector and the connector will send the information to the web portal to display it to the user.

4.2.2 RESPONSIBILITIES

The connector's sole responsibility is to make sure that data flow between the web portal and the central database layer are well-established and for the brewery information to be securely stored and retrieved from the database. First, the connector will establish connection to the database once it receives an input from the web portal regarding a specific brewery product. Once the connector receives the information of the brewery product, then the connector sends the information over to the database and the database will search for the brewery product and return additional information of the product to the connector.

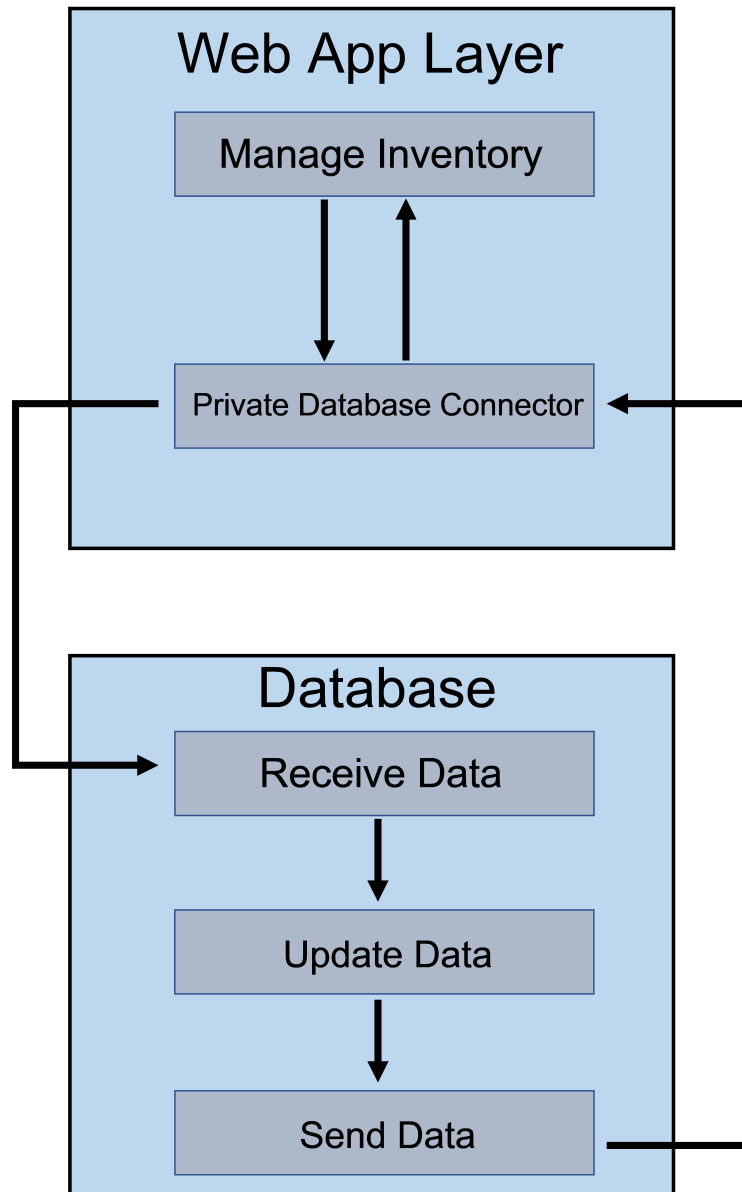


Figure 4: Example subsystem description diagram

The connector will then send the information to the web portal and from there the web portal will display the brewery product information to the user. On the other hand, if the user decides to modify or delete a record from the database, the connector will send the request to the database and the database will delete or modify the record from the user's request.

4.2.3 PRIVATE DATABASE CONNECTOR INTERFACES

Table 3: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Web portal	brewery, name of beer style	expanded information on brewery product
#02	Central brewery DB	Information on brewery product	detailed information on product sent to the connector

5 Y 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.

Layer Y is mobile application layer. The subsystem layers are barcode reader, public database connector, private database connector and manage inventory. In these subsystem users can scan the barcode number and get the product information from database with respective connectors (public and private). User can get the information of beer on the basis of their type, order, style and price available in database. Barcode reader is connected to public database through its connectors; and if user enters the barcode number of selected products, then user can get through the information from manage inventory through private database connectors. Public database conn helps to get the product records stored in public database. Private database connectors help in database entry. It promotes to change the records, upgrade, delete the information in brewing database. Manage Inventory allow for information to be managed and stored into the brewery database.

5.1(Subsystem 1) The subsystem of the Mobile app layer is Barcode reader. This subsystem scans the barcode number and get the product information like type, price, style and date from the database with respective private and public connectors. Barcode reader is connected with public database connector through barcode number. 5.1.1(Assumption) The subsystem Barcode reader is connected to public database connector through barcode number. It can access the information from the public database. Similarly, when user wants to access the information from the private database then probably, user have to do manually whatever stored in system. They cannot access private information directly and without permission granted. 5.1.2 (Responsibilities) As our subsystem is barcode reader. It is responsible to scan the beer products through barcode number. It is representation of data of related products. It performs to get the information of beer based on its types, origin, price, style and date. It extracts the information by its barcode number available in database. 5.1.3 (Subsystem Interference) ID Description Inputs Outputs xx Barcode reader Brewing products Barcode number

5.2(subsystem 2) Another subsystem of mobile app is public database connector. Public database conn helps to get the product records stored in public database. In public connectors, you can get the public information from public database. Like what type of product, it is (brand), its origin, best by date and type. Public database connectors are connected with public database through barcode number; as database send information to public on the basis of its number. Similarly, it is also connected with the private database through selected information.

5.2.1(Assumption)

As public database connector is linked up with private database connector still it cannot modify and edit any records form the mange inventory unless the user provides the approval. It cannot access the information that is managed and stored in brewery database without the permission granted. It's hard to access the database entry of the product. Therefore, they will do manually.

5.2.2 (Responsibilities) Public database connector helps to get the product records stored in public database. In public connectors, you can get the public information from public database. Like what type of product, it is (brand), its origin, best by date and type. In public database connectors, anyone can access database and can get the information regarding the product. No need of permission of Users to search the product on public database.

5.2.3 (Subsystem Interference) ID Description Inputs Outputs xx Public database connector Barcode number, Product information Selected information

5.3(subsystem 3) Another Subsystem of mobile app is private database connector. Private database connectors help in database entry. It promotes to change the records, upgrade, delete the information in

brewing database. Private database connector is connected with the manage inventory through database entry. It helps to manage records of product and store in brewing database.

5.3.1(Assumption) The subsystem private database connector is connected with the manage inventory. It can only operate the private database. Only the approval people can access the database and can get the information. Therefore, the user will add, change and update the product information into the brewery database manually. 5.3.2(responsibilities) Private database connectors help in database entry. It promotes to change the records, upgrade, delete the information in brewing database. It manages records of product and store in brewing database. It controls the private database and only limited user can access who got approval.

5.3.3 (Subsystem Interference) ID Description Inputs Outputs xx Private database connectors Selected information, database entry Database entry

5.4 (subsystem 4)

Another subsystem of mobile app is Manage Inventory. The system will allow for information to be managed and stored into the brewery database. The user can add, modify and delete information stored in database. Manage inventory is connected with the private database connector through database entry.

5.4.1(Assumption)

As manage inventory app is connected only with the private database connector. The information available in database can be changed only when User provides the approval. Therefore, a brewery product is not available on the public database when the mobile app scans the barcode of the brewery product, will allow the user to add/input the product information into the brewery database manually and become available for retrieval.

5.4.2 (Responsibilities) The subsystem manage inventory allows information to be managed and stored into the database. The user will be able to add, delete, and modify any records that are currently stored and saved on the database. The information of beer products will include name of product, type, its id and best by date.

5.4.3 (Subsystem Interference) ID Description Inputs Outputs xx Manage inventory Database entry Database entry

6 Z LAYER SUBSYSTEMS

Layer z is the database layer. The subsystem will be two databases that is public and a private database where all the informations on the beers will be stored. When the app layer or mobile layer interacts the databases the informations is supplied through web portals or database connectors. And when information is required to be updated it is updated through the database entry.

6.1 SUBSYSTEM 1

This subsystem consist of the local database. So when the web application or mobile layer uses database connectors or the web portal to find the required information if the the inforamtion is stored in the public database the inforamtion is provided and displayed in the requiired layer.

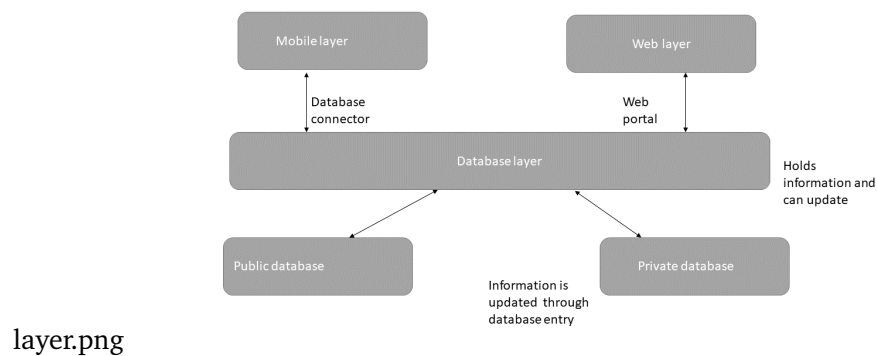


Figure 5: Example subsystem description diagram

6.1.1 ASSUMPTIONS

When the mobile layer or web layer requests the data or seems to change it from public database the database does the required task and displays inthe respected layer or it connects to private database to create or get infromation already created.

6.1.2 RESPONSIBILITIES

When the mobile layer or web layer requests the data from public database the database does the required task and displays inthe respected layer

6.1.3 SUBSYSTEM INTERFACES

It will be simitlar to public database used.

Table 4: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Web portal and databse connector	local database	expanded infor- mation on brewery prod- uct

6.2 SUBSYSTEM 2

This subsystem consist of the private database. So when the web application or mobile layer uses database connectors or the web portal to find the required information if the the inforamtion is not stored in the public database they go to this database which allows to create the required information or the already existinfg one is provided and displayed in the required layer.

6.2.1 ASSUMPTIONS

When the mobile layer or web layer requests the data and is not availbale from pubic database the database does it connects to private database to create or get requested infromation already created in private database.

6.2.2 RESPONSIBILITIES

When the mobile layer or web layer requests the data or seems to change it from database layer the database does the required task and displays inthe respected layer

6.2.3 SUBSYSTEM INTERFACES

Table 5: Subsystem interfaces

ID	Description	Inputs	Outputs
#01	Web portal and databse connector	private database	expanded infor- mation on brewery prod- uct

REFERENCES