



JSK Company

Division of Software Sales

Version 3.0.0

DESIGN DOCUMENTATION

Min Sup Kim, Junmo Kim, Sanil Khamkar

JSK COMPANY

TABLE OF CONTENTS

I. Introduction	3
I.I. Purpose of Documentation.....	3
I.II. Major problems	3
I.III. Project goals	3
I.IV. Proposed System Overview	4
II. Hardware/Software.....	5
II.I. Hardware	5
II.II Software	5
III. Design Priority	6
III.I. Alternatives	6
III.II. Design Priority Table	7
III.III Development and Execution Environment	8
III. IV. Programming Tools	9
III.V. Naming and Coding Standards.....	10
III.VI. Error Handling.....	12
III.VII. Design Constraints	13
III.VIII. Hardware and Software Constraints	13
III.IX. Agile Methodology	14
IV. Data Flow Diagram	15
IV.I. Data dictionary.....	23
V. Design Legend	25
V.I. Sales Report Format.....	25
V.I.I. Weekly Sales Report Format.....	25
V.I.II. Monthly Sales Report Format.....	26
V.I.III Annual Sales Report Format	28
V.II. Non-Functional Requirements.....	29
VI. Top Level Design.....	32
VII. Medium Level Design	33
1. Login	33
1. Sales	35
2. Accounts Payable/ Receivable	38
3. Inventory Control	41
4. Advertisement/Promotion	49

5. Reservation System	52
VIII. Files and Tables	56
VIII. I Database Tables	56
IX. Definition, Meeting Log, FWBS, References.....	58
IX.I Definition.....	58
IX.II. Meeting Log	61
IX.III. FWBS	63
IX.IV References	68

I. INTRODUCTION

I.I. Purpose of Documentation

The purpose of this document is to provide simple and easy design features of the International Restaurant Smart System to our client, Dr. Shin, and give brief understanding about our design levels to user, which are administrator, employees, and customer. The main goal of this design to deliver less time and cost than previous manual system. And we expect to provide best design and features that directly connects between our design with users and fits the client's needs.

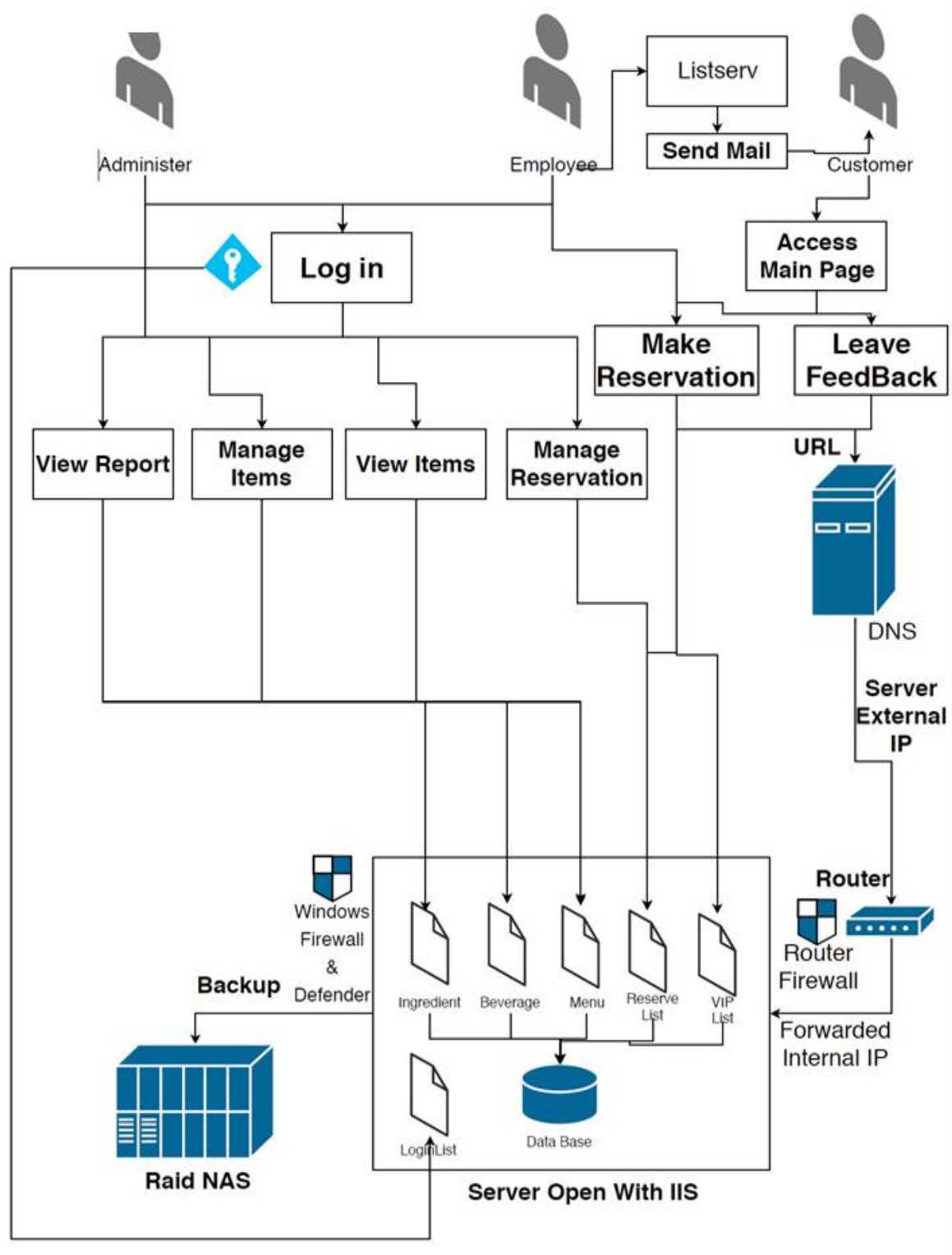
I.II. Major problems

- The software should be tested for bugs and fixed before deploying the system to the customer.
- Requirement gathering. If the requirements gathered are not proper, it may lead to incorrectness of a program which leads to project failure.
- A plan is required before starting with the development of a software application. Without a plan it may lead to wrong cost estimates and duration of a project.
- Collaboration with the customer. Requesting for a meeting and asking for requirements are essential when it comes to development of a project.

I.III. Project goals

The goal of this project is to design an International Restaurant Management System which will be deployed to the customer. The application should allow the restaurant manager to provide faster service in a less amount of time and make twice the amount of revenue. The application should be developed with an interactive user-friendly interface to provide the ease of operations for the administrator, staff as well as the customer. It should be easy for the administrator to manage the inventory and keep track of existing stock with weekly or monthly sales report. The reservation system provided by the application should allow the customers to reserve the table prior coming to the restaurant.

I.IV. Proposed System Overview



II. HARDWARE/SOFTWARE

II.I. Hardware

Name	Condition
Display	1280x720 or higher
CPU	2.0 Ghz of Intel Core architecture 2 nd Generation or higher
RAM	4GB of DDR3 or higher
Storage	1TB of HDD/SSD or higher
Keyboard	US-101Key USB 2.0 or higher
Mouse	USB 2.0 Laser mouse or higher
Hardware Backup	RAID 1 Capable 2bay RAID enclosure
Router	TCP/UDP Port openable with ac profile with NAS Backup Support or higher

II.II Software

Name	Condition
OS	Microsoft Windows 10 PRO
Network Speed	10Mbps or higher
Anti-virous	Windows Defender 4.18.1807 or higher
Firewall	Windows Firewall 1.275.1176.0
Program management	Visual Studio 2017 Basic
Program Package	.NET CORE 2.0 .NET Framework IIS Control
Database	Microsoft SQL
Database management	SQL manager studio

III. DESIGN PRIORITY

III.I. Alternatives

· Front-End Languages:

Number	Language	Description
1.	HTML	Hyper Text Markup Language is used for designing web pages with colors, fonts text style etc.
2.	CSS	Cascading Style Sheet is used for controlling the style of a web document.
3.	JavaScript	JavaScript is used for network-centric applications which is implemented with HTML.

· Back-End Languages:

Number	Language	Description
1.	C#	C# is a general-purpose, multi-paradigm programming language used for development of Web applications.
2.	Ruby	Ruby is general-purpose, object-oriented programming language used for web development.
3.	Python	Python is a general-purpose programming language used for both web and desktop application development.
4.	PHP	Hypertext Preprocessor is a server-side scripting language designed for web development.
5.	Java	Java is a general-purpose object-oriented programming language used for creating applications that can be distributed among servers and clients in a network.
6.	SQL	Structure query language is a domain specific language used for designing and managing data in a database system.

· Usability of language in software development:

Number	Language	Priority (Importance)
1.	HTML	10
2.	CSS	10
3.	JavaScript	7
4.	C#	10
5.	Ruby	6

6.	Python	7
7.	PHP	8
8.	Java	9
9.	SQL	10

*Priority from 1 to 10: 1 being the lowest and 10 being the highest.

III.II. Design Priority Table

The Priority Table resembles the priority of each design. Our developer team has created three designs based on different design criteria's.

Number	Design Priority Criteria	Priority (Importance)	Design 1	Design 2	Design 3
1.	Performance	9	8	6	7
2.	Readability	8	8	7	6
3.	Writability	7	6	7	5
4.	Reliability	7	7	6	7
5.	Cost	9	8	6	4
6.	Maintainability	5	6	5	3
7.	Modifiability	6	7	3	3
8.	Usability	4	7	3	5
9.	Memory Efficiency	7	7	4	5
10.	Security	8	9	7	7

Number	Design Priority Criteria	Weight of Design 1 C# Language	Weight of Design 2 PHP Language	Weight of Design 3 Java Language
1.	Performance	72	54	63
2.	Readability	64	56	48

3.	Writability	42	49	35
4.	Reliability	49	42	49
5.	Cost	72	54	36
6.	Maintainability	30	25	15
7.	Modifiability	42	18	18
8.	Usability	28	12	20
9.	Memory Efficiency	49	28	35
10.	Security	72	56	56
11.	Total	520 (Best)	340	375

III.III Development and Execution Environment

1. Development Environment

The Development Environment refers to the tools and application used for designing and building the software.

1. Operating System

- Windows 10 Home Edition 64bit
- Intel core i5 generation 5 or equivalent

2. Integrated Development Environment

- Microsoft Visual Studio 2017
- Sublime Text 3

3. Programming Language

- HTML5
- CSS3
- C#
- JavaScript
- SQL

4. Database

- Microsoft SQL Server 2017

- Microsoft SQL Server Management Studio 17

5. Word Processor

- Microsoft Word 2016
- Microsoft PowerPoint 2016
- Microsoft Excel 2016

2. Execution Environment

The Execution Environment refers to the environment in which the software/hardware is executed.

a) System Configuration

- Windows 10 Home Edition 64bit or higher
- Intel core i3 generation 2 or higher
- RAM: 4 Gigabytes DDR3
- Memory Space required: 30 Gigabytes or higher
- Intel HD420 graphics or equivalent

b) Browser

- Google Chrome v60.0.3497.100
- Firefox v61.0.2
- Microsoft Edge v42.17134.1.0
- Internet Explorer v11.345.17134.0

III. IV. Programming Tools

1. Operating System

- Windows 10 Home Edition 64bit
- Intel core i5 generation 7

2. Integrated Development Environment

- Microsoft Visual Studio 2017
- Sublime Text 3

3. Programming Language

- HTML5
- CSS3

- C#
- JavaScript
- SQL

4. Database

- Microsoft SQL Server 2017
- Microsoft SQL Server Management Studio 17

5. Word Processor

- Microsoft Word 2016
- Microsoft PowerPoint 2016
- Microsoft Excel 2016

6. Browser

- Google Chrome v60.0.3497.100
- Microsoft Edge v42.17134.1.0

III.V. Naming and Coding Standards

1. Naming Conventions

In short examples that do not include using directives, use namespace qualifications. A namespace is imported by default in a project, you do not have to fully qualify the names from that namespace. Qualified names can be broken after a dot (.) if they are too long for a single line, as shown in the following example.

Example: C#

```
var currentPerformanceCounterCategory = new System.Diagnostics.  
PerformanceCounterCategory();
```

The names of objects that were created need not be changed by using the Visual Studio designer tools to make them fit other guidelines.

2. Coding Conventions

Coding conventions serve the following purposes:

- They create a consistent look to the code, so that readers can focus on content, not layout.

- They enable readers to understand the code more quickly by making assumptions based on previous experience.
- They facilitate copying, changing, and maintaining the code.
- They demonstrate C# best practices.

3. Layout Conventions

Good layout uses formatting to emphasize the structure of your code and to make the code easier to read.

- Use the default Code Editor settings (smart indenting, four-character indents, tabs saved as spaces).
- Write only one statement per line.
- Write only one declaration per line.
- If continuation lines are not indented automatically, indent them one tab stop (four spaces).
- Add at least one blank line between method definitions and property definitions.
- Use parentheses to make clauses in an expression apparent, as shown in the following code.

Example: C#

```
if ((val1 > val2) && (val1 > val3))
{
    // Take appropriate action.
}
```

4. Commenting Conventions

- Place the comment on a separate line, not at the end of a line of code.
- Begin comment text with an uppercase letter.
- End comment text with a period.
- Insert one space between the comment delimiter (//) and the comment text, as shown in the following example.

Example: C#

```
// The following declaration creates a query. It does not run
// the query.
```

- Do not create formatted blocks of asterisks around comments.

III.VI. Error Handling

1. The Try Block

A `try` block is used by C# programmers to partition code that might be affected by an exception. Associated `catch` blocks are used to handle any resulting exceptions. A `finally` block contains code that is run regardless of whether an exception is thrown in the `try` block, such as releasing resources that are allocated in the `try` block. A `try` block requires one or more associated `catch` blocks, or a `finally` block, or both.

2. The Catch Block

A `catch` block can specify the type of exception to catch. The type specification is called an *exception filter*. The exception type should be derived from `Exception`. In general, do not specify `Exception` as the exception filter unless either you know how to handle all exceptions that might be thrown in the `try` block, or you have included a `throw` statement at the end of your `catch` block.

Example: C#

```
try
{
    // Code to try goes here.
}
catch (SomeSpecificException ex)
{
    // Code to handle the exception goes here.
    // Only catch exceptions that you know how to handle.
    // Never catch base class System. Exception without
    // rethrowing it at the end of the catch block.
}
```

3. The Finally Block

A `finally` block enables you to clean up actions that are performed in a `try` block. If present, the `finally` block executes last, after the `try` block and any matched `catch` block. A `finally` block always runs, regardless of whether an exception is thrown or a `catch` block matching the exception type is found.

Example: C#

```
try
{
    // Code to try goes here.
```

```

}
finally
{
    // Code to execute after the try block goes here.
}

```

Example: C#

```

try
{
    // Code to try goes here.
}
catch (SomeSpecificException ex)
{
    // Code to handle the exception goes here.
}
finally
{
    // Code to execute after the try (and possibly catch) blocks
    // goes here.
}

```

III.VII. Design Constraints

- The application was developed using Visual Studio 2017 and used C# for developing the software
- The user must login prior reserving the time, date for a table in the restaurant
- Only the admin and a few staff who are registered with the system can gain login access
- Internet connection is required for backing up the data to the server and retrieve any other information

III.VIII. Hardware and Software Constraints

1. Hardware Constraints

The application can be deployed on the following specification or higher.

- Intel core i3 generation 2 or higher
- RAM: 4 Gigabytes DDR3
- Memory Space required: 30 Gigabytes or higher
- Intel HD420 graphics or equivalent

2. Software Constraints

- Windows 10 Home Edition 64bit

- Google Chrome v60.0.3497.100
- Firefox v61.0.2
- Microsoft Edge v42.17134.1.0
- Internet Explorer v11.345.17134.0
- The system is available through web browser only

III.IX. Agile Methodology

Our team has implemented the agile project management process. The process will follow four main key steps: team member interactions and meetings on a daily basis, working software over documentation, client collaboration and responding to change over the following plan.

1. Setting a plan with strategy meeting

At the beginning of the project, we defined a clear business plan of what will be the outcome of this project and how to gather tools required for implementing in the project.

2. Sprint Cycle

Our team has implemented short cycles of development in which specific tasks and goals will be performed. Our Sprint cycle will last between 1-2 weeks and will remain the same throughout the entire project to plan future work more accurately. Our team created a list of items that need to be completed in the given timeframe to create the functional software.

3. Weekly Project Meetings

Our team follows two team meetings per week. The meeting discussion consists of goal of the project and the amount of work completed till date. Various tasks are given for each individual member of the team and each team member reports the work he has done since the last meeting. Our team also discusses about the work to be done before the next meeting and the problems encountered with a particular task.

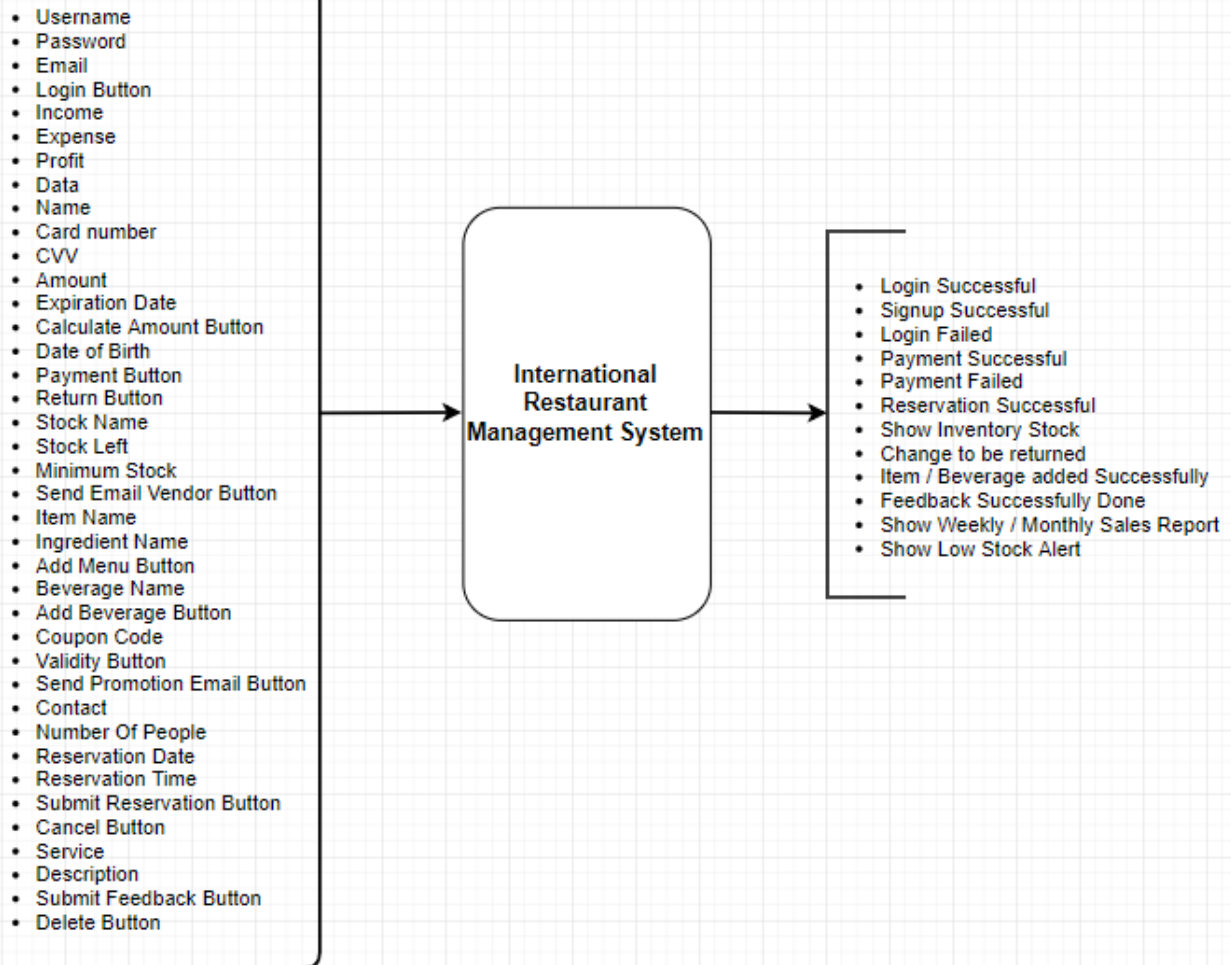
4. Review Meeting

This meeting requires our team members and the client to meet at a given time based on the clients request to demonstrate the project work and documentation implemented till the last team meeting. The client reviews the project work and the documentation and gives the feedback, if any changes must be done.

IV. DATA FLOW DIAGRAM

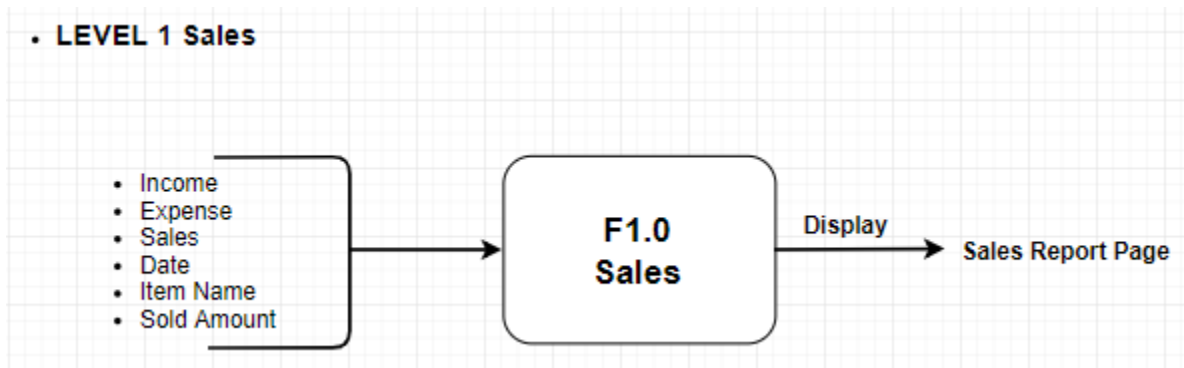
LEVEL 0

• LEVEL 0

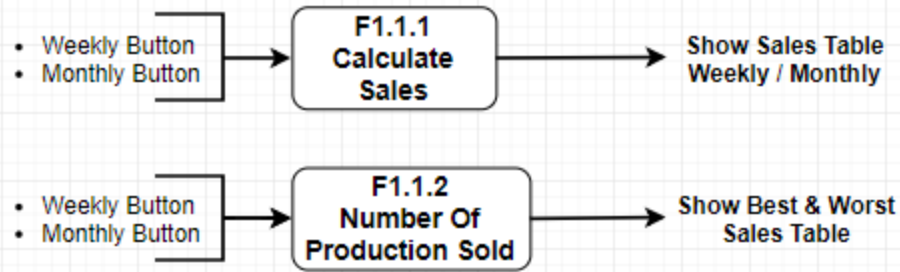


Data Flow Diagram - Sales

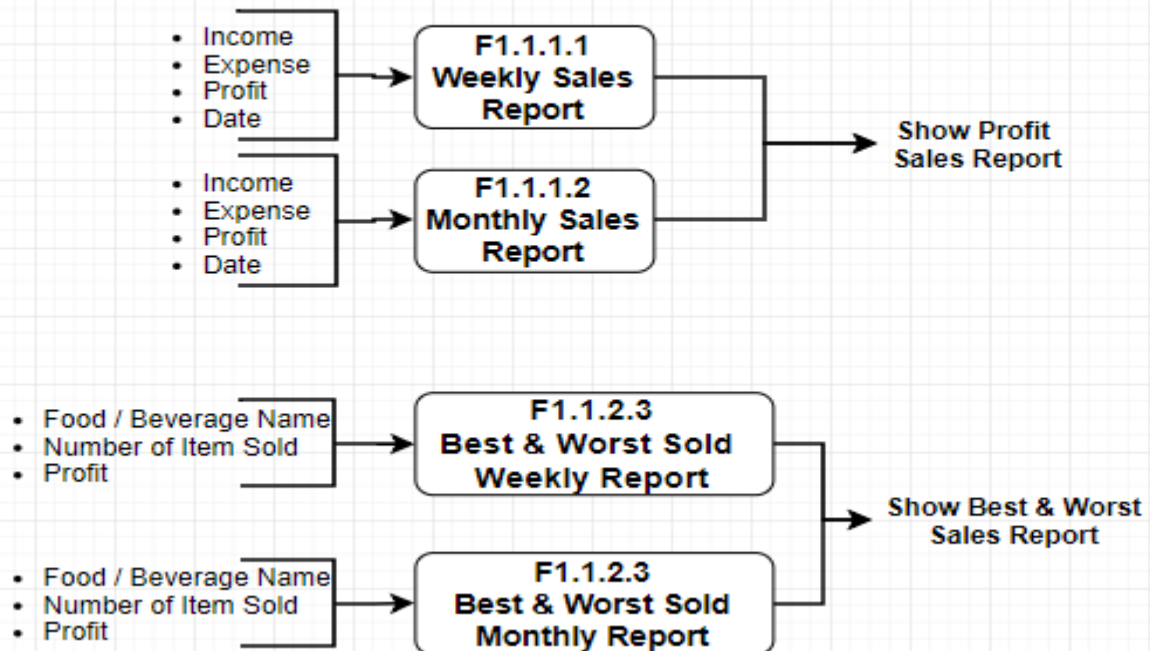
• LEVEL 1 Sales



• LEVEL 2 Sales

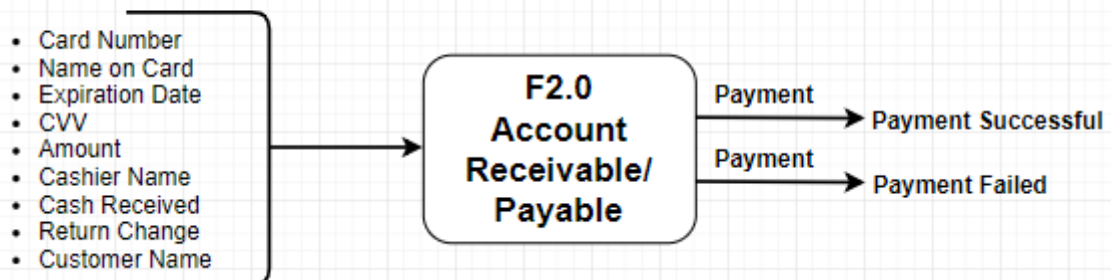


• LEVEL 3 Sales

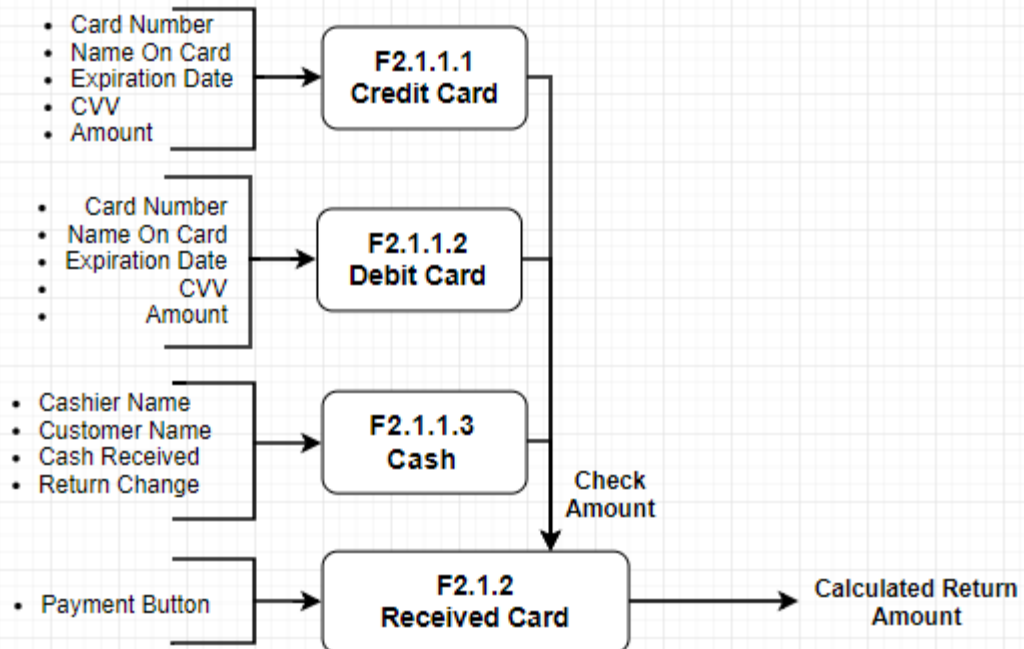


Data Flow Diagram – Account Receivable/ Payable

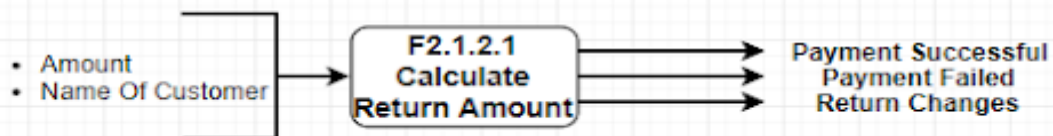
• LEVEL 1 Account Receivable / Payable



• LEVEL 2 Account Receivable / Payable

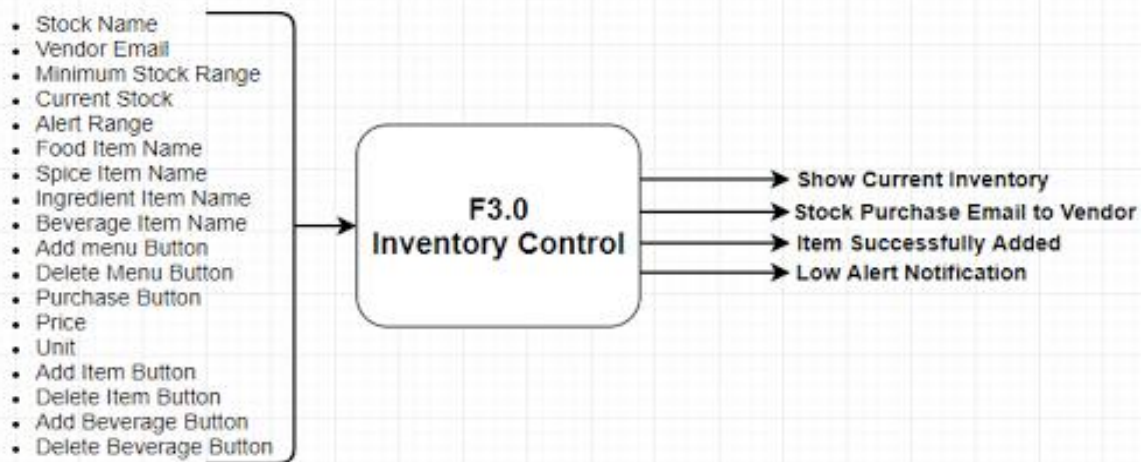


• LEVEL 3 Account Receivable / Payable

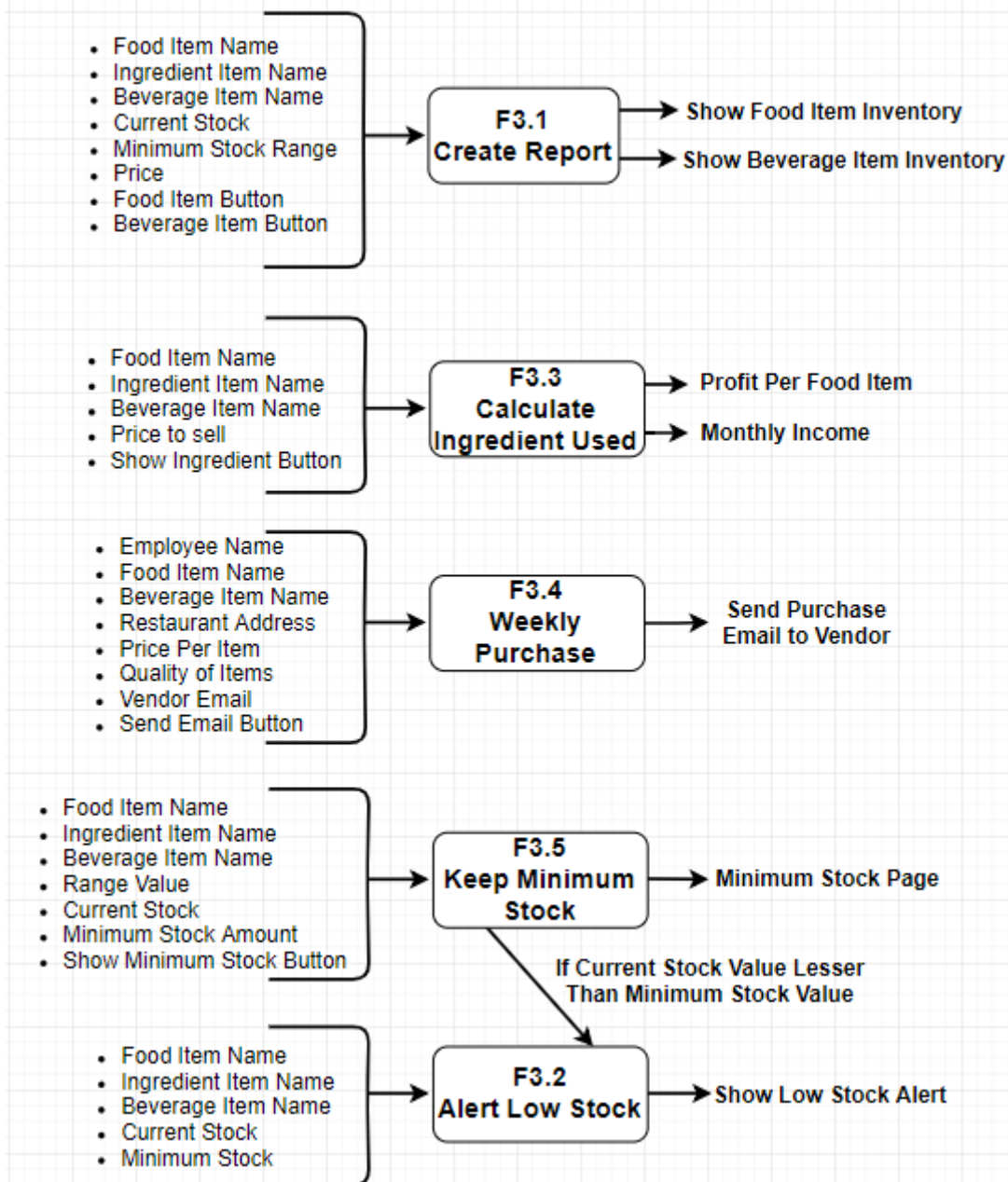


Data Flow Diagram – Inventory

• LEVEL 1 Inventory Control

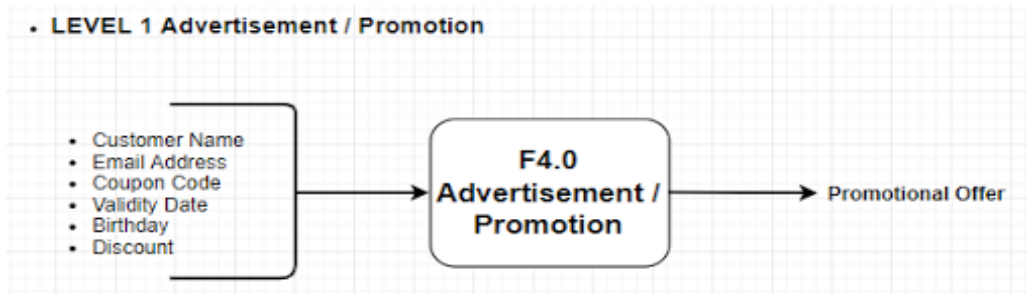


• LEVEL 2 Inventory Control

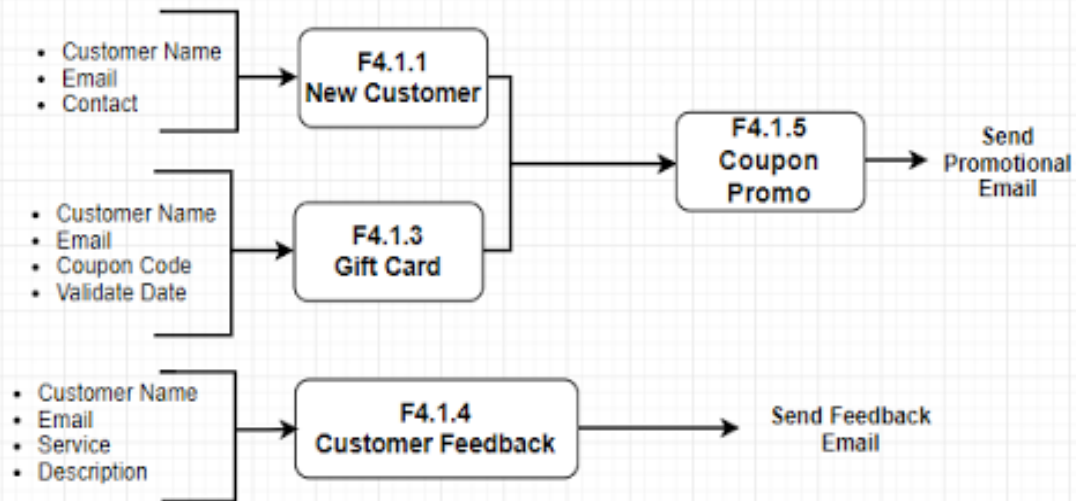


Data Flow Diagram – Advertisement/Promotion

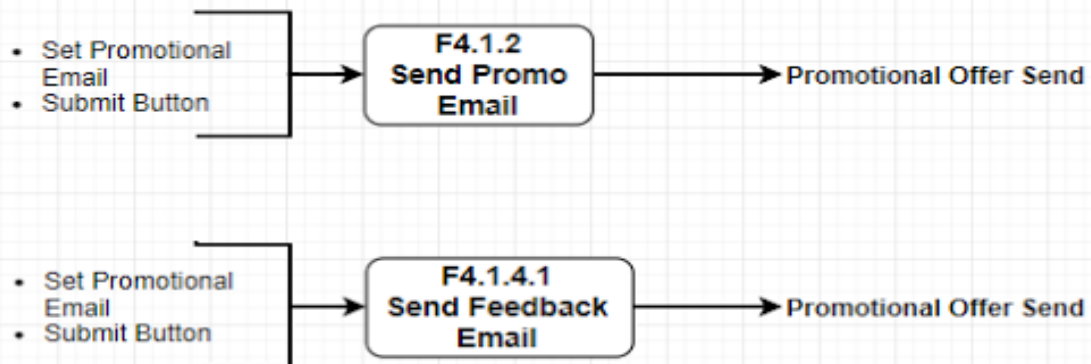
• LEVEL 1 Advertisement / Promotion



• LEVEL 2 Advertisement / Promotion

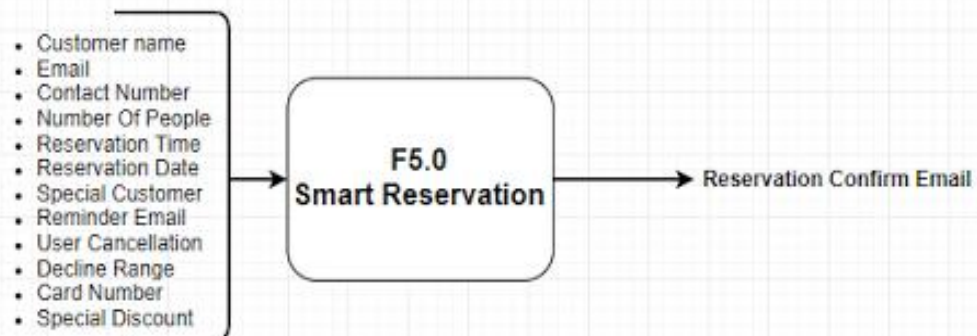


• LEVEL 3 Advertisement / Promotion

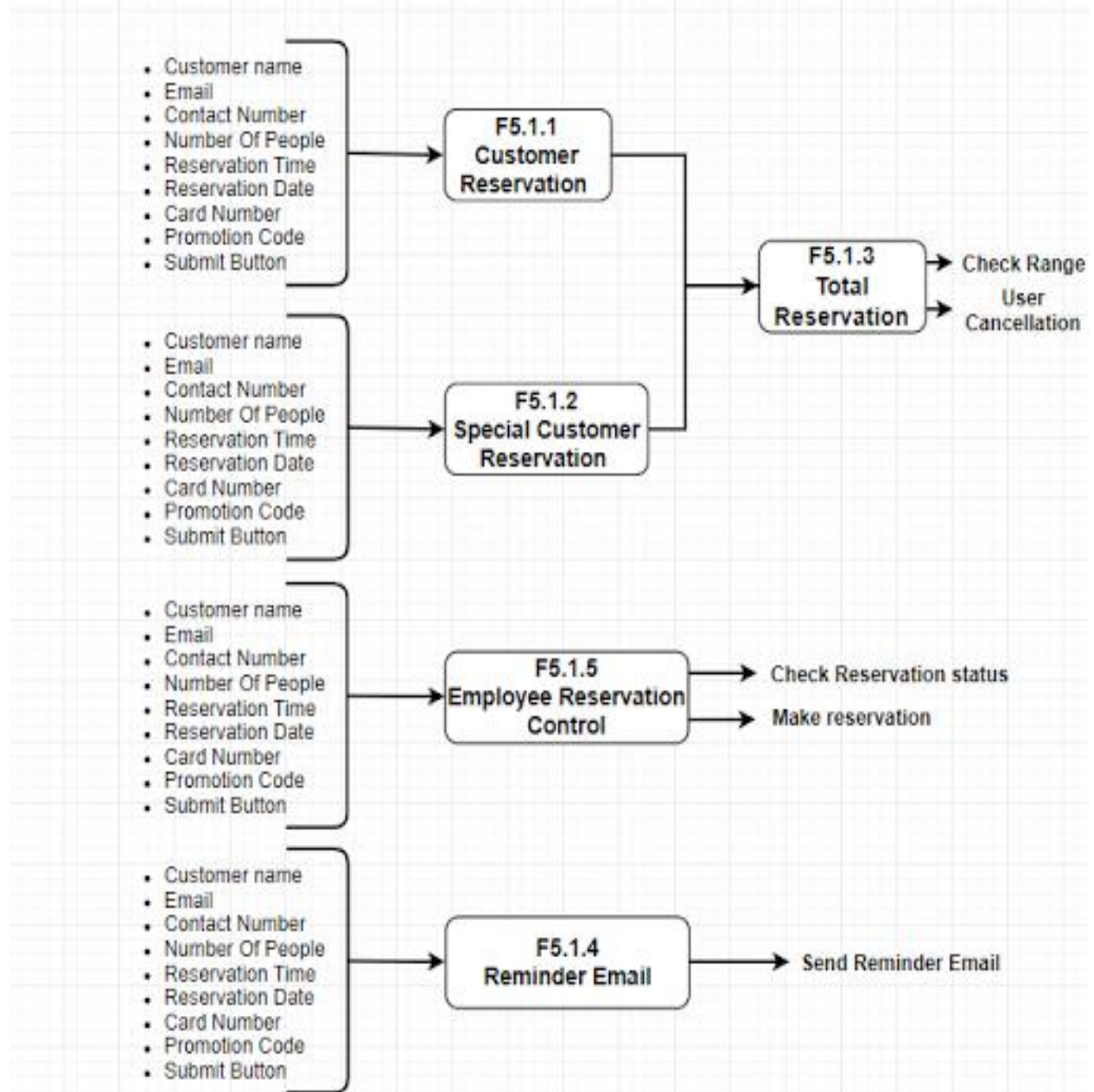


Data Flow Diagram – Reservation

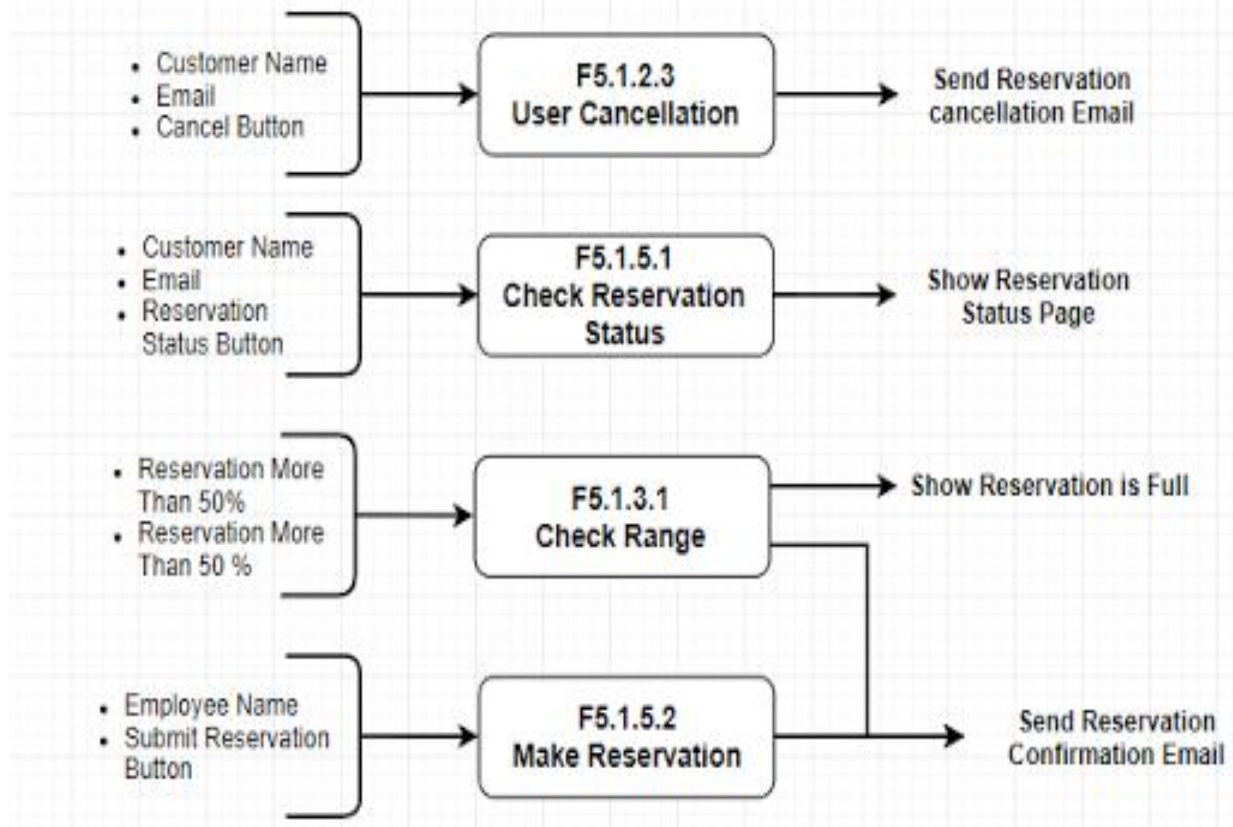
• LEVEL 1 Reservation



• LEVEL 2 Reservation

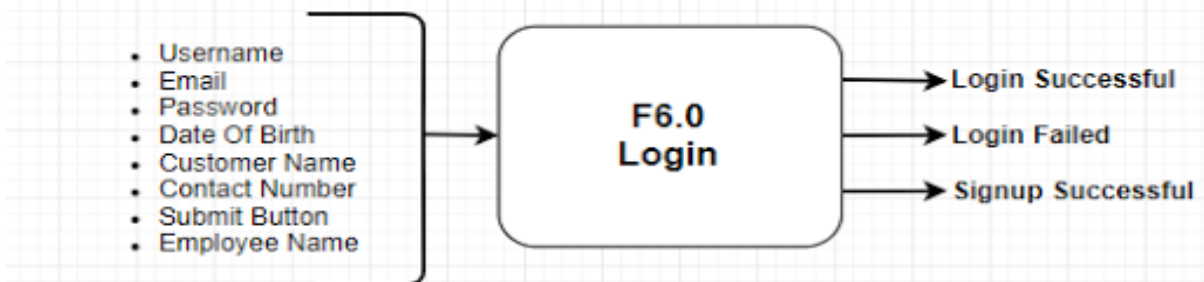


• LEVEL 3 Reservation

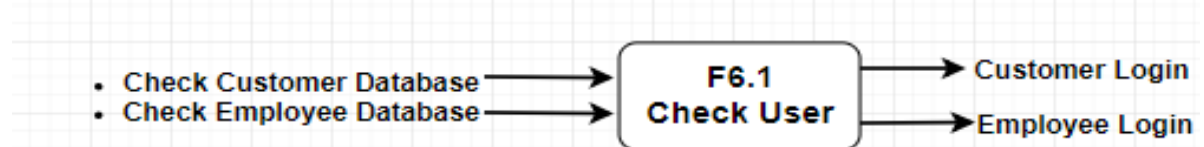


Data Flow Diagram - Login

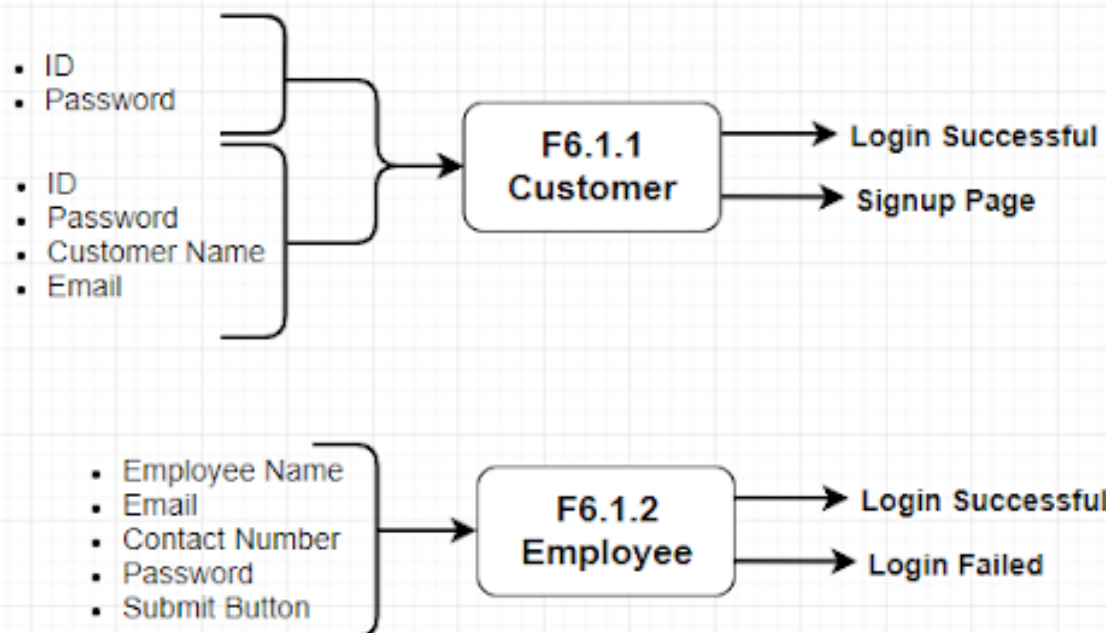
• LEVEL 1 Login



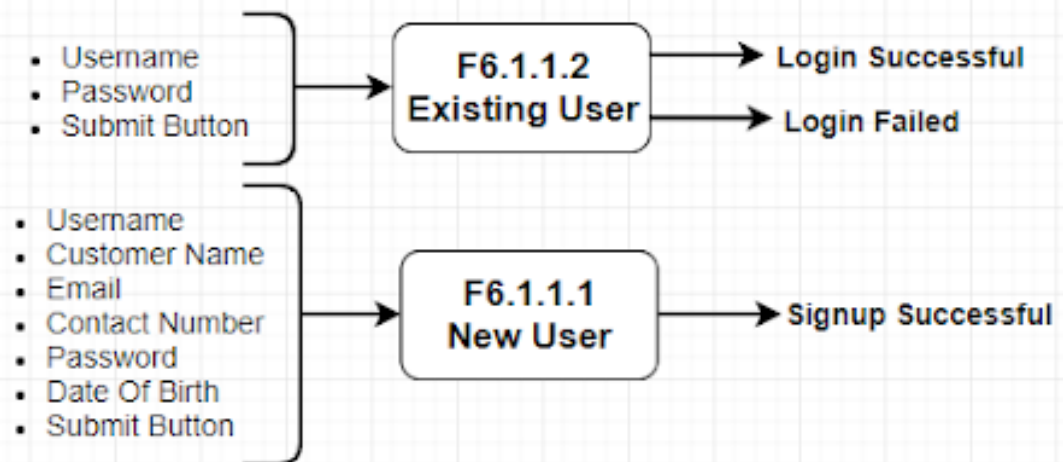
• LEVEL 2 Login



• LEVEL 3 Login



• LEVEL 4 Login



IV.I. DATA DICTIONARY

Name	Type of Data	Additional info	FWBS#
Admin_manage	Button	None	F5.1.2
Best_and_Worst	DropDownList 4	Click and select Best and Worst	F1.1.2.3
Beverage_report	DropDownList 3	or Beverage	F1.1.2.2
Calendar	Calendar	Select date	F5.1.1.1
Card_Holder	TextBox	None	F2.1.1.1.2, F2.1.1.2.2
Card_Holder	TextBox	None	F2.1.1.1.2, F2.1.1.2.2
Card_number	TextBox	None	F2.1.1.1.1, F2.1.1.2.1
Card_number	TextBox	None	F2.1.1.1.1, F2.1.1.2.1
Cash	button	None	F2.1.1.3
Cash	button	None	F2.1.1.3
CheckReservDetail	Button	None	F5.1.2.1
Credit_Card	button	None	F2.1.1.1
Customer_email	TextBox	None	F4.1.1.1, F4.1.2, F4.1.4.1
Customer_email	Textbox	None	F5.5.1.1
Customer_Name	TextBox	None	F4.1.1.1, F4.1.2, F4.1.4.1
CustomerEmail	TextBox	None	F5.1.2.4
CustomerName	TextBox	None	F5.1.2.3
Debit_Card	button	None	F2.1.1.2
Email_body	Textbox	Contains larger size	F5.5.1.3
Email_head	Textbox	None	F5.5.1.2
Employee_button	Button	None	F0.1.1.2
Food_report	DropDownList 3	Click and select Food	F1.1.2.1
Login_button	Button	None	F0.1.1
Low_Stock_Alert	message	None	F3.1.3
Low_Stock_Alert	message	None	F3.1.3
Make_Ad_Promo	Button	None	F4.1.1
MakeReservation	Button	Make reservation based on phone call	F5.1.2.2
Map_info	Label	None	F0.1.3
Monthly_report	DropDownList 2	or Month	F1.1.1.2
New_Signup_button	Button	None	F0.1.4
NumOfPeople	RadioButtonList	None	F5.1.1.5
Pay_button	button	None	F2.1.1.4
Pay_button	button	None	F2.1.1.4
Print_Error	Popup_message	Show Payment error	F2.1.1.6
Print_Error	Popup_message	Show Payment error	F2.1.1.6
Print_Success	Popup_message	Show Payment success	F2.1.1.5
Print_Success	Popup_message	Show Payment success	F2.1.1.5
Product_report	DropDownList 1	or Product	F1.1.2
Purchase	DropDownList6	Click and select Food, Beverage, Ingredient	F3.1.5.5

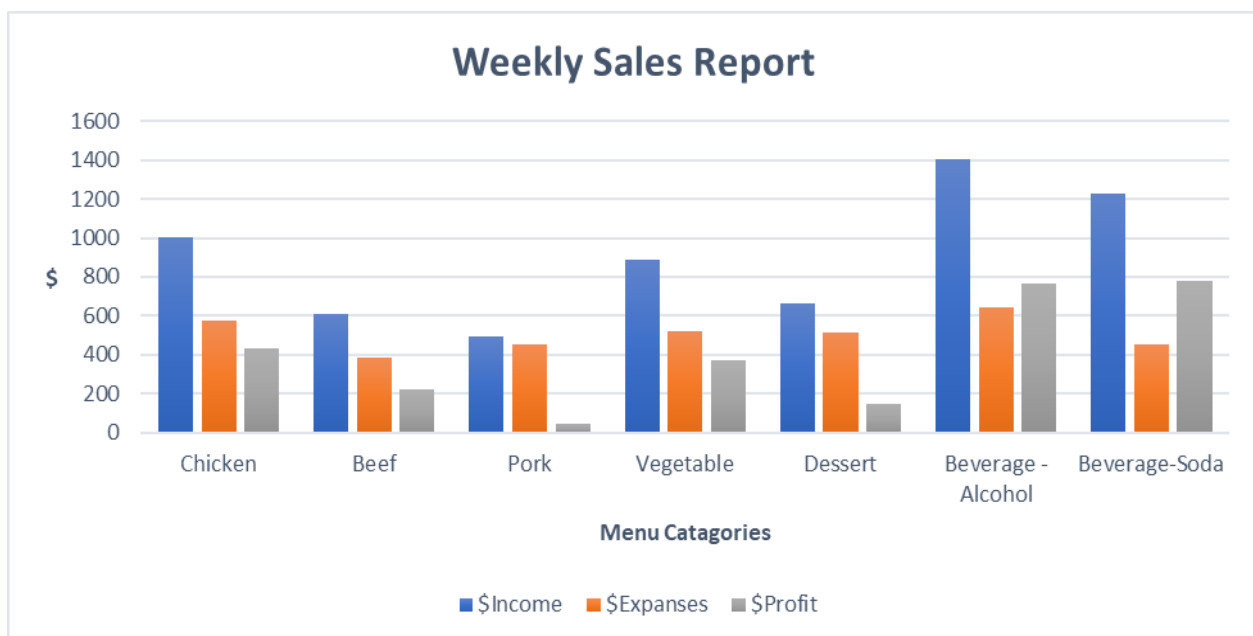
Reserve_button	Button	None	F0.1.1.1
Return_amount	Label	None	F2.1.2, F2.1.2.1
Return_amount	Label	None	F2.1.2, F2.1.2.1
Sales_report	DropDownList 1	Click and select Sales	F1.1.1
Spec_card_num	Integer	None	F5.2.1.1
Stock_report_generate	DropDownList 5	Click and select Food, Beverage, Ingredient	F3.1.1
Stock_report_generate	DropDownList 5	Click and select Food, Beverage, Ingredient	F3.1.1
Submit_button	Button	None	F5.1.1.6
Submit_button	Button	None	F6.1.3
TableSuggestion	ImageButtons	Select Table based on color	F5.1.1.4
Time_info	Label	None	F0.1.2
TimeSheet	Dropdownlist	Select Lunch or Dinner	F5.1.1.2
TimeTable	RadioButtonList	None	F5.1.1.3
User_cancel	Button	None	F5.1.3.2
User_ID	TextBox	None	F6.1.1
User_pasw	TextBox	None	F6.1.2
Weekly_report	DropDownList 2	Click and select Weekly	F1.1.1.1

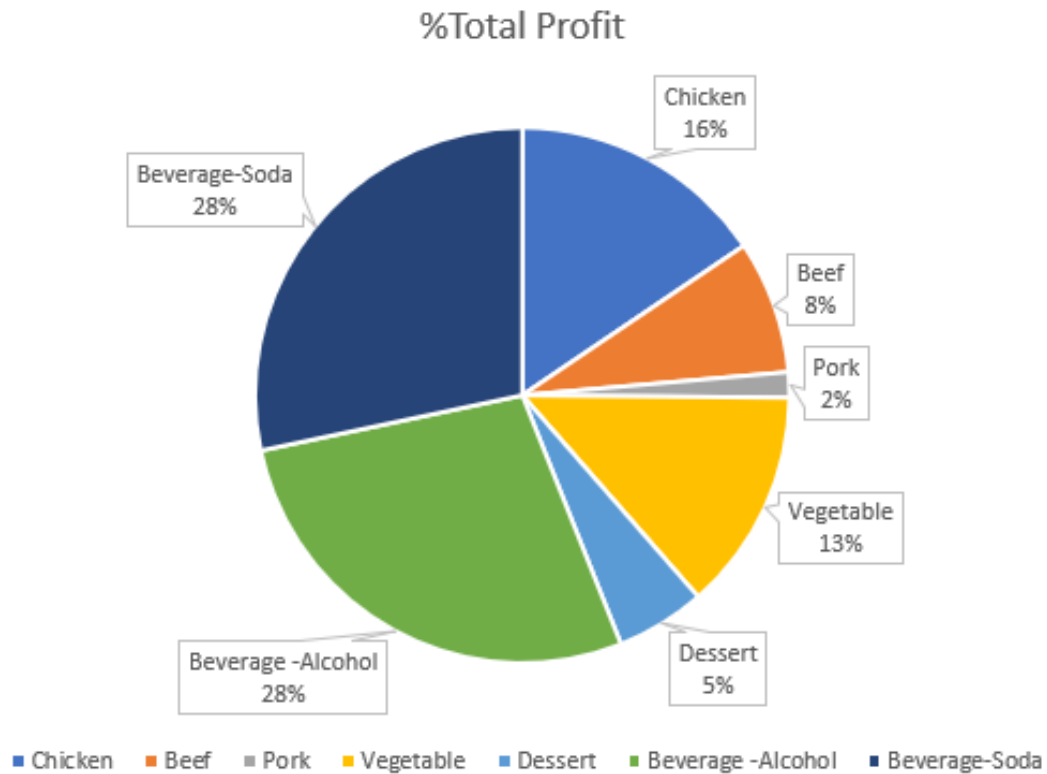
V. DESIGN LEGEND

V.I. Sales Report Format

V.I.I. Weekly Sales Report Format

International Restaurant Inc. Weekly Sales Report Format				
Menu categories	\$Income	\$Expenses	\$Profit	%Total Profit
Chicken	1002.87	573.32	429.55	16%
Beef	609.23	387.22	222.01	8%
Pork	492.44	450.21	42.23	2%
Vegetable	890.32	520.23	370.09	13%
Dessert	660.32	510.77	149.55	5%
Beverage-Alcohol	1405.89	641.32	764.57	28%
Beverage-Soda	1231.33	450.21	781.12	28%
Total	6292.4	3533.28	2759.12	100%





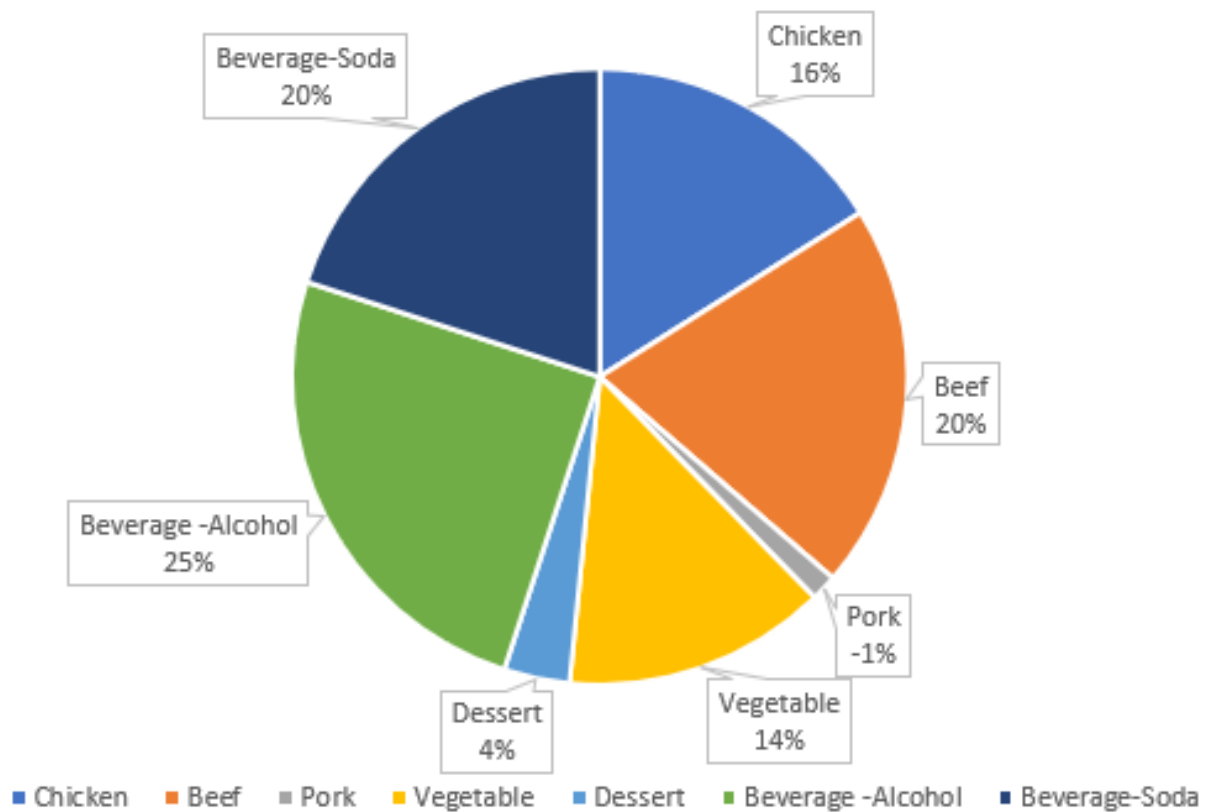
V.I.II. Monthly Sales Report Format

International Restaurant Inc. Monthly Sales Report Format				
Menu categories	\$Income	\$Expanses	\$Profit	%Total Profit
Chicken	3982.87	2431.32	429.55	16%
Beef	2609.23	1387.22	222.01	8%
Pork	2092.44	1150.21	42.23	2%
Vegetable	4090.32	1890.23	370.09	13%
Dessert	2260.32	1710.77	149.55	5%
Beverage-Alcohol	7405.89	3441.32	764.57	28%
Beverage-Soda	5131.33	1950.21	781.12	28%
Total	6292.4	3533.28	2759.12	100%

Monthly Sales Report

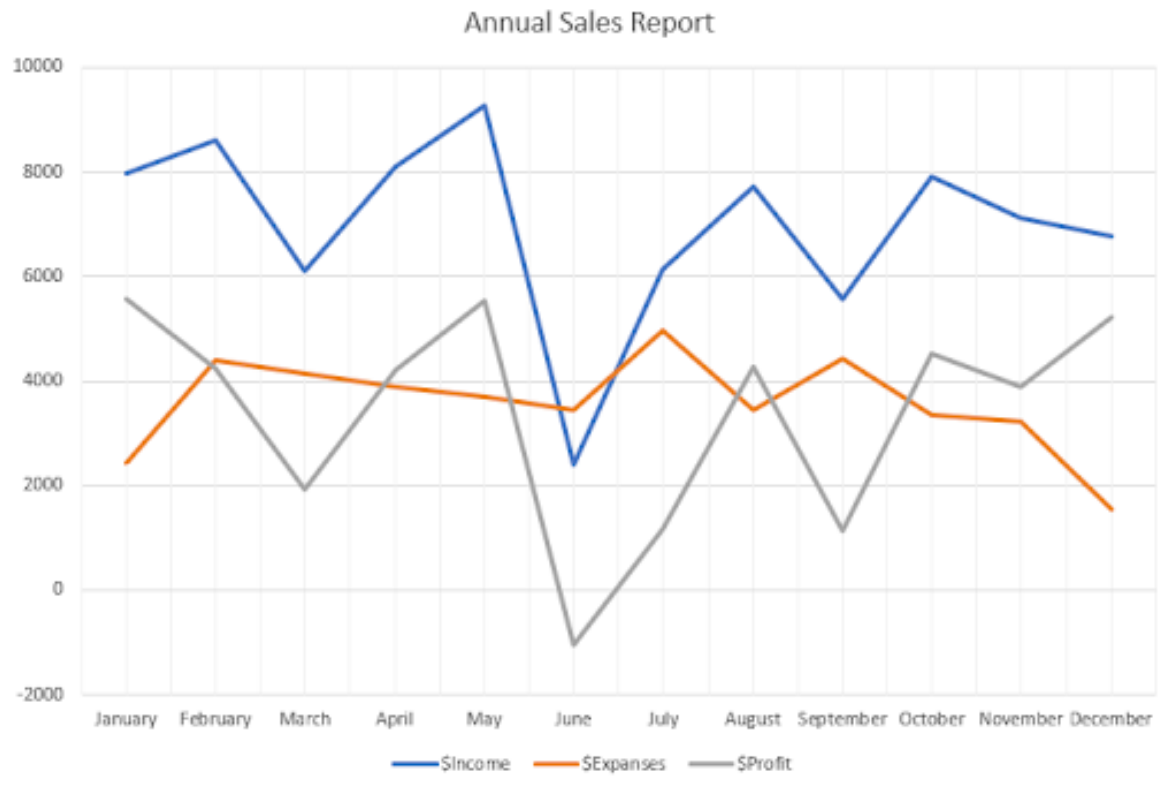


%Total Profit



V.I.III Annual Sales Report Format

International Restaurant Inc. Annual Sales Report Format				
Selected Menu Category = xxxxxx				
Selected Menu Category = Beverage-Alcohol				
Month	\$Income	\$Expenses	\$Profit	%Total Profit
January	7982.87	2431.32	5551.55	14%
February	8609.23	4387.22	4222.01	10%
March	6092.44	4150.21	1942.23	5%
April	8090.32	3890.23	4200.09	10%
May	9260.32	3710.77	5549.55	14%
June	2405.89	3441.32	-1035.4	-3%
July	6131.33	4950.21	1181.12	3%
August	7724.33	3445.34	4278.99	11%
September	5564.66	4432.43	1132.23	3%
October	7896.75	3365.43	4531.32	11%
November	7132.66	3232.22	3900.44	10%
December	6783.12	1562.63	5220.49	13%
Total	83673.9	42999.3	40674.6	100%



V.II. Non-Functional Requirements

1. Performance

Performance represents the response time of the functionalities within the system which should be as minimal as possible to improve performance and reduce the waiting time for the user.

- FWBS 6.0 Login
 1. FWBS 6.1.1.1 New User (SignUp): Creating an account for a new user and clicking on the SignUp button requires a response time of 2 seconds.
 2. FWBS 6.1.1.2 Existing User: Logging into the system after login button click requires a response time of 1 seconds.
- FWBS 1.0 Sales
 1. FWBS F1.1.1.1 Weekly Report: Clicking on the Weekly button to show the weekly sales report requires a response time of 1 second.

2. FWBS F1.1.1.2 Monthly Report: Clicking on the Monthly button to show the monthly sales report requires a response time of 1 second.

3. FWBS F1.2.1/FWBS F1.2.2 Food/Beverage Report: Clicking on the Food/Beverage button to show the weekly/monthly sales report for food/beverage requires a response time of 1 second.

4. FWBS F1.2.3 Best and Worst Sales: Clicking on the Best/Worst Sales button to show the best/worst sales on weekly/monthly basis requires a response time of 1 second.

- FWBS F2.0 Account Receivable/Payable

1. FWBS F2.1 Credit Card Payment: Clicking on Payment button shows payment successful or payment failed requires a response time of 4 seconds.

2. FWBS F2.2 Received Card (Calculate Amount): Printing the receipt after successful payment requires a response time of 2 seconds.

- FWBS F3.0 Inventory Control

1. FWBS F3.1 Create report: Clicking on the show report button for checking the current stock in the inventory requires a response time of 1 second.

2. FWBS F3.4 Weekly Purchase: Clicking on the send email button after filling the details for purchasing the items requires a response time of 1 seconds.

- FWBS F5.0 Smart Reservation

1. F5.1 Reservation: Clicking on make reservation button requires a response time of 2 seconds.

2. FWBS F5.5.1 Check Reservation Status: Clicking on check reservation status button requires a response time of 1 seconds.

2. Security

The Customer login information and credit card details are securely stored in the database which are in the encrypted form.

- Use of MD5 hash function for securely storing user's password.
- Payment Card Industry (PCI) Data Security Standard is used for storing the Credit Card information securely.

3. Backup

Application information is stored on the cloud server in case of system failure or power loss.

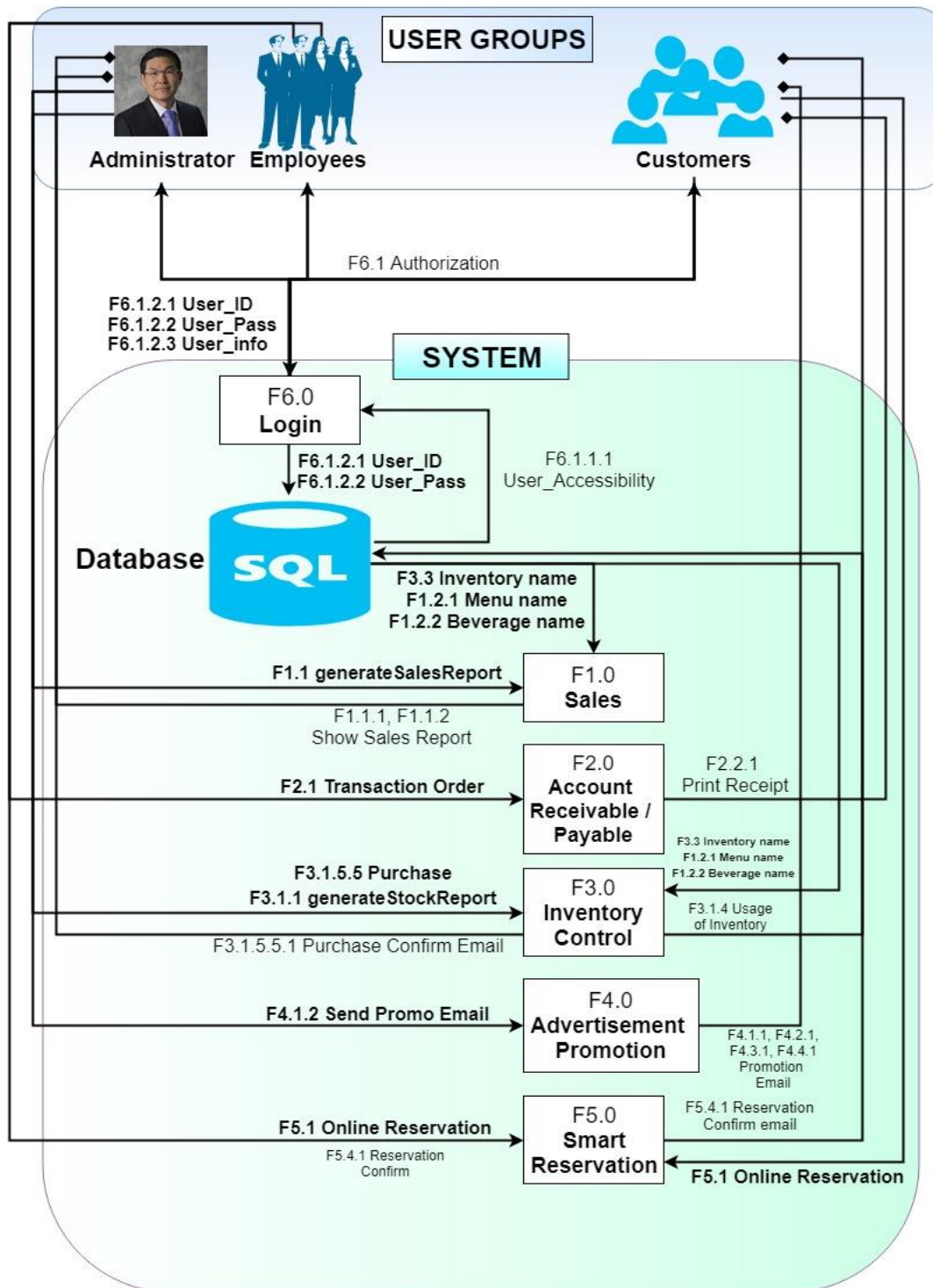
- Cloud Server is used for backup
Provided with a dedicated server for backing up the data.
- The application data is backed up on the server every day at 00:00 hours.

4. Maintainability

Maintainability of application refers to modifying, updating the existing application for improved performance.

- After completion of the product, functionalities can be added to the product or the product can be updated based on the needs on the client's request.

VI. TOP LEVEL DESIGN



VII. MEDIUM LEVEL DESIGN

1. Login

The Login resembles the FWBS 6.0. It allows the Customer, Admin, Employee to login and gain access to further functionalities.

I. Pseudocode: FWBS 6.0

```
Restaurant_Database DB;

protected void btnlogin( String User_ID, String User_Pass )
{
    int login_count = 0;
    while(login_count < 4)
    {
        if(User_ID == true && User_Pass == true)
        {
            GotoMainPage - logged in page
            return ;
        }
        else
        {
            Message pop up "Wrong ID or Password please input correctly"
            Add 1 to login_count
        }
    }
    if(login_count == 4)
    {
        Message pop up "Please try again few minutes later"
        GotoMainPage - login failed
        return;
    }
}
```

II. Table No. 1

FWBS No.	Name	Input	Description/Output
6.0	Login	User_ID and User_Pass in Textbox 1 and Textbox 2	User, staff or administrator login successful SC#1

6.1.1.1	New User Sign up	ID, Pass, Email, Date of Birth in Textbox 1, Textbox 2, Textbox 3, Textbox 4	Account created Successful message popup, redirect to main page
6.1.4.1	Login successful	Click the Login button	Redirect to main page

III. Screenshot SC #1

Page 1

https://www.irbrookings.com/login

Home | About | Reservation | **Employee Login** | Rate Us!

Employee ID

Password

Log In

Work on progress, final outcome may looks not same

1. Sales

The Sales resembles the FWBS 1.0 it shows the reports based on profits and what menu or beverages are sold most, which can be helpful to determine business operations.

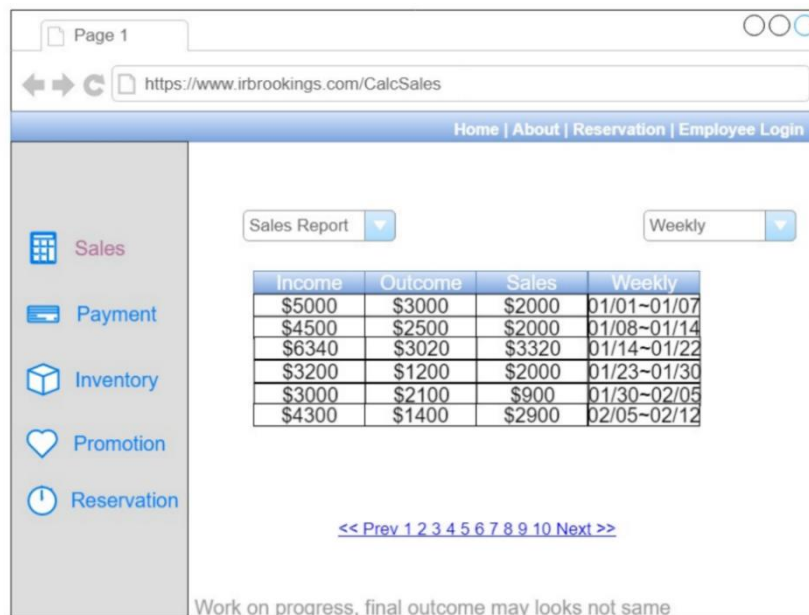
IV. Pseudocode: FWBS 1.0

```
protected void btngeneratesalesreport( Mouse_action_listener e )
{
    While User do not select different menu option
    {
        If user selected sales report
        {
            Select one of based on user selects
            {
                case weekly:
                    showWeeklyReport();
                    break;
                case monthly:
                    showMonthyReport();
                    break;
            }
        }
        else if(user selected product report)
        {
            Select one of based on user selects
            {
                Case Food:
                    showFoodReport();
                    break;
                Case Beverage:
                    showBeverageReport();
                    break;
                case Best&Worst:
                    showBestWorstReport();
                    break;
            }
        }
    }
    goToDifferentMenu( based on user selects );
}
```

V. Table No. 1

FWBS No.	Name	Input	Description/Output
1.1.1.1	Total Weekly Report	Select weekly in DropDownList 1	Display weekly sales report SC#2
1.1.1.2	Total Monthly Report	Select Monthly in DropDownList 1	Display monthly sales report SC#3
1.1.2.1	Weekly/Monthly Food Report	Select Food in DropDownList 2	Display sales report about Food SC#5
1.1.2.2	Weekly/Monthly Beverage Report	Select Beverage in DropDownList 2	Display sales report about Beverage SC#4

VI. Screenshots SC #2, SC #3, SC #4, SC #5



Page 1

<https://www.irbrookings.com/sales>

[Home](#) | [About](#) | [Reservation](#) | [Employee Login](#)

Sales

Payment

Inventory

Promotion

Reservation

Sales Report
Monthly

Income	Outcome	Sales	Month
\$50000	\$30000	\$20000	18/01
\$45000	\$25000	\$20000	18/02
\$63400	\$30200	\$33200	18/03
\$32000	\$12000	\$20000	18/04
\$30000	\$21000	\$9000	18/05
\$43000	\$14000	\$29000	18/06

[<< Prev](#)
[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[7](#)
[8](#)
[9](#)
[10](#)
[Next >>](#)

Work on progress, final outcome may looks not same

Page 1

<https://www.irbrookings.com/ProductionSold>

[Home](#) | [About](#) | [Reservation](#) | [Employee Login](#)

Sales

Payment

Inventory

Promotion

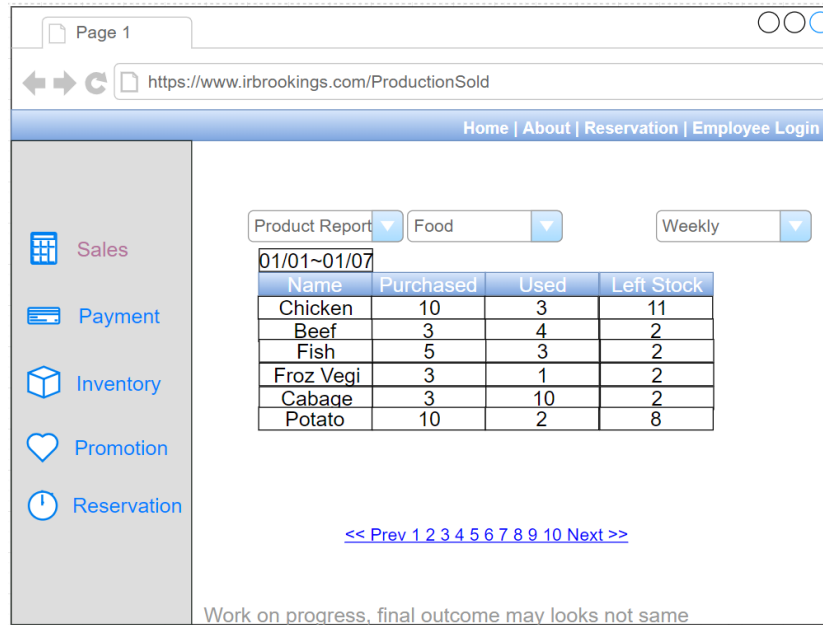
Reservation

Product Report
Beverage
Monthly

18/01			
Name	SoldAmount	Profit	Left Stock
Johnny-Blue	3	3	1
Soiu-Cham	40	30	10
Sake-Gek	30	20	20
Beer-Hien	400	300	200
Sprite	100	250	20
Coke	100	240	5

[<< Prev](#)
[1](#)
[2](#)
[3](#)
[4](#)
[5](#)
[6](#)
[7](#)
[8](#)
[9](#)
[10](#)
[Next >>](#)

Work on progress, final outcome may looks not same



2. Accounts Payable/ Receivable

The Accounts Payable/Receivable resembles the FWBS 2.0. It shows the credit or debit card payment profile and cash calculation.

I. Pseudocode: FWBS 2.0

```
protected void btnmakePayment ( Mouse_action_listener e)
{
    While user not selects different menu
    {
        if( user selects credit card or debit card)
        {
            int CardNum = get Value from ;
            string Name = Textbox2.getValue();
            int expiration_month = Textbox3.getValue();
            int expiration_year = Textbox4.getValue();
            int CVV = Textbox5.getValue();
            float charged_amount = Textbox6.getValue();
            float tip_amount = Textbox7.getValue();

            if( e.selected == Submit_button )
            {
                if( gotoPayment_card( CardNum, Name, expiration_month,
expiration_year, CVV, charged_amount, tip_amount ) == true )
                {
                    gotoPaymentSuccessPage_card();
                }
            }
        }
    }
}
```

```

    }
    else
    {
        gotoPaymentFailedPage_card();
    }
    return;
}
}
else if( e.selected == Cash )
{
    string Name = Textbox1.getValue();
    float amount_charged = Textbox2.getValue();
    float tip_amount = Textbox3.getValue();
    float amount_paid = Textbox4.getValue();

    if( e.selected == Submit_button )
    {
        int return_amount = gotoPayment_cash( Name, amount_charged,
tip_amount, amount_paid);

        gotoPaymentSuccess_cash( return_amount );
        return;
    }
}
}
goToDifferentMenu( e.selected );
}

```

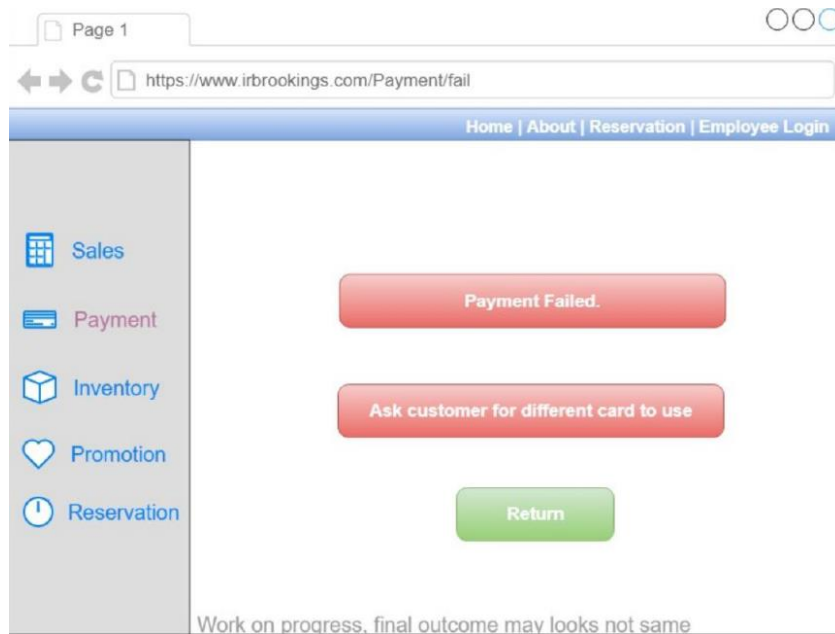
II. Table No. 1

FWBS No.	Name	Input	Description/Output
2.1.1.1	Credit Card	Click Credit button, write Card number, Expiration Date, Name, CVV, and click make payment button	SC#6 then SC#7 or SC#8
2.1.1.2	Debit Card	Click Debit button, write Card number, Expiration Date, Name, CVV, and click make payment button	SC#6 then SC#7 or SC#8
2.1.1.3	Cash	Click Cash button, write How much paid, Name, and click make payment button	SC#6 then SC#7 or SC#8

III. Screenshot SC #6, SC #7, SC #8

This screenshot shows the 'Credit' payment form in a web browser. The browser's address bar displays 'https://www.irbrookings.com/Payment/Credit'. The page has a blue header with navigation links: 'Home | About | Reservation | Employee Login'. On the left, a grey sidebar contains icons and labels for 'Sales', 'Payment', 'Inventory', 'Promotion', and 'Reservation'. The main content area features two tabs, 'Cash' and 'Credit', with 'Credit' being the active tab. Below the tabs are input fields for 'Card Number', 'Name on Card', 'Expiration' (split into 'Month' and 'Year'), 'CVV', 'Amount' (labeled 'Amount to charge'), and 'Tip Charge' (labeled 'Tip charge'). A green 'Submit Payment' button is positioned below these fields. At the bottom of the page, a disclaimer reads: 'Work on progress. final outcome may looks not same'.

This screenshot shows the 'Payment Successful' confirmation screen in the same web browser. The address bar now shows 'https://www.irbrookings.com/Payment/Successful'. The header and sidebar remain identical to the previous screenshot. The main content area displays a large blue button that says 'Payment Successful!'. Below it is another blue button that says 'Return to payment screen automatically!'. The same disclaimer, 'Work on progress. final outcome may looks not same', is visible at the bottom of the page.



3. Inventory Control

The Inventory Control resembles the FWBS 3.0. It shows the current stock in the restaurant and what stock will be needed for the next week.

I. Pseudocode: FWBS 3.0

```
protected void btninventorycontrol( Mouse_action_listener e ){
    while( e.selected != differentMenu )
    {
        switch( e.selected )
        {
            case Ingredient:
                btnIngredientManage();
                break;
            case Food:
                btnFoodManage();
                break;
            case Menu:
                btnMenuManage();
                break;
            case Beverage:
                btnBeverageManage();
                break;
        }
    }
    goToDifferentMenu( e.selected );
}
```

II. Pseudocode2: FWBS 3.0

```
protected void btningredientmanage( Mouse_action_listener e )
{
    bringIngredientReport();
    System.pop_up(Low_alert_message);

    while( e.selected != differentMenu )
    {
        switch( e.selected )
        {
            case setAutoPurchase:
                System.pop_up("Auto purchasement started");
                btnpurchase(Auto, ingredient);
                break;
            case forcePurchase:
                btnpurchase(Force, ingredient);
                break;
            case deleteIngredient:
                btndelete(ingredient);
                break;
            case addIngredient:
                btnadd(ingredient);
                break;
        }
    }
    goToDifferentMenu( e.selected );
}

protected void btnfoodmanage( Mouse_action_listener e )
{
    bringFoodReport();
    System.pop_up(Low_alert_message);

    while( e.selected != differentMenu )
    {
        switch( e.selected )
        {
            case setAutoPurchase:
                System.pop_up("Auto purchasement started");
                btnpurchase(Auto, food);
                break;
            case forcePurchase:
                btnpurchase(Force, food);
                break;
        }
    }
}
```

```

        case deleteIngredient:
            btndelete(food);
        break;
        case addIngredient:
            btnadd(food);
        break;
    }
}
goToDifferentMenu( e.selected );
}

protected void btnmenumanage( Mouse_action_listener e )
{
    while( e.selected != differentMenu )
    {
        switch( e.selected )
        {
            case deletemenu:
                btndelete(menu);
            break;
            case addmenu:
                btnadd(menu);
            break;
        }
    }
    goToDifferentMenu( e.selected );
}

protected void btnbeveragemanage( Mouse_action_listener e )
{
    bringBeverageReport();
    System.pop_up(Low_alert_message);

    while( e.selected != differentMenu )
    {
        switch( e.selected )
        {
            case setAutoPurchase:
                System.pop_up("Auto purchasement started");
                btnpurchase(Auto, beverage);
            break;
            case forcePurchase:
                btnpurchase(Force, beverage);
            break;
            case deleteBeverage:

```

```

        btndelete(beverage);
        break;
        case addBeverage:
            btnadd(beverage);
            break;
    }
}
goToDifferentMenu( e.selected );
}

protected void btndelete( Mouse_action_listener e, Item d )
{
    while( e.selected != differentMenu )
    {
        switch( d )
        {
            case ingredient:
                Ingredient_delete();
                break;
            case food:
                food_delete();
                break;
            case menu:
                menu_delete();
                break;
            case beverage
                beverage_delete();
                break;
        }
    }
    goToDifferentMenu( e.selected );
}

protected void btnadd( Mouse_action_listener e, Item a )
{
    while( e.selected != differentMenu )
    {
        switch( a )
        {
            case ingredient:
                add_ingredient();
                break;
            case food:
                add_food();
                break;

```

```

        case menu:
            add_menu();
            break;
        case beverage:
            add_beverage();
            break;
    }
}
goToDifferentMenu( e.selected );
}

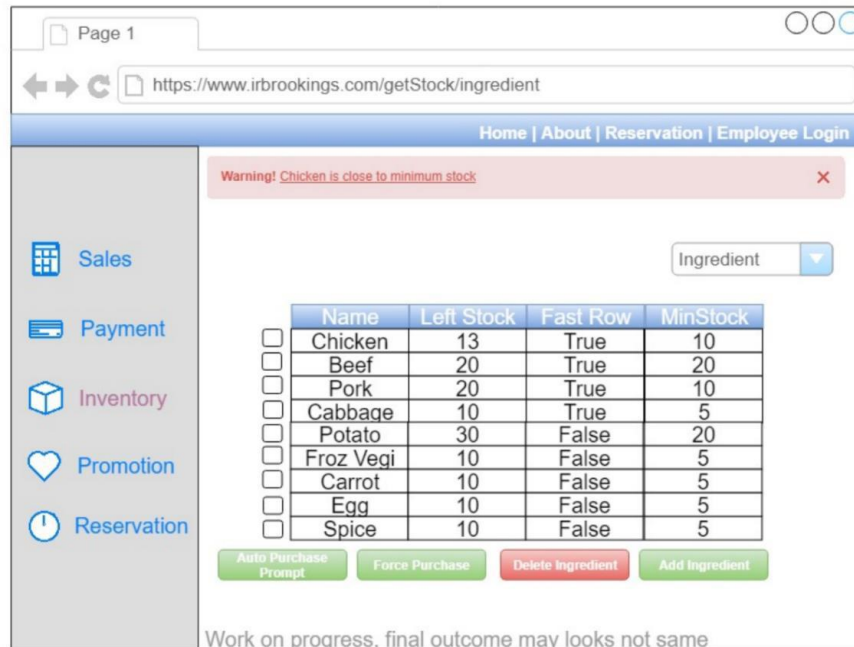
protected void btnpurchase(Mouse_action_listener e, Status s, Purchasing_Item
i )
{
    //S is either Auto or Immediate
    while( e.selected != differentMenu )
    {
        switch( i )
        {
            case ingredient:
                send_email(purchase, s, ingredient);
                break;
            case food:
                send_email(purchase, s, food);
                break;
            case beverage:
                send_email(purchase, s, beverage);
                break;
        }
    }
    goToDifferentMenu( e.selected );
}
}

```

III. Table No. 1

FWBS No.	Name	Input	Description/Output
3.1.1	Show report	Click on the show inventory button	Displays the current stock in the inventory SC#9
3.1.2	Low stock alert	An item in the inventory is near or goes below the minimum value set for a particular stock item	Displays a warning, alerting the admin SC#9 includes alert on top

IV. Screenshot SC #9



V. Table No. 2

FWBS No.	Name	Input	Description/Output
3.1.5.3	Weekly Purchase	Click on purchase button	Input the item name and amount to be purchased SC#10
3.1.5.4	Immediate Purchase	Alert is not manually cancelled	Sends an auto generated email to vendor for purchasing items SC#11

VI. Screenshot SC #10

The screenshot shows a web browser window with the address bar displaying <https://www.irbrookings.com/fpurchase>. The page has a blue header with navigation links: Home | About | Reservation | Employee Login. On the left, there is a sidebar menu with icons and labels: Sales (grid icon), Payment (credit card icon), Inventory (box icon), Promotion (heart icon), and Reservation (clock icon). The main content area contains a form for purchasing ingredients. The form fields are: Item to purchase (a dropdown menu showing 'Ingredient'), Buy Amount (a text input field with 'Buying Amount' as a placeholder), Unit (a text input field with 'Unit' as a placeholder), Price to buy (a text input field with 'Price to buy' as a placeholder), and Mail address to order (a text input field with 'Autofill with modifiable' as a placeholder). Below the form is a red button labeled 'Force Purchase'. At the bottom of the page, there is a note: 'Work on progress, final outcome may looks not same'.

VIII. Pseudocode

FWBS No.	Name	Input	Description/Output
3.1.1	Purchase	Click on purchase button	Sends an email to vendor for purchasing items SC#11

X. Screenshot SC #11

From	<input type="text" value="IRB@IRbrookings.com"/>
Subject	<input type="text" value="Order Inventory (IRB)"/>
To	<input type="text" value="Retailer@Retail.com"/>
CC	<input type="text"/>
BCC	<input type="text"/>

Thank you for being prescious partner of
International Restaurant Brookings!

This is Weekly regular automatic purchase ordering mail
from Brookings International Restaurant

Order for inventory items from Retailer
Name of Order: Chicken
Amount: 10 Full Chicken
Price want to buy: \$2 per chicken

If you want to renegotiate with price please reply this mail!

Thank you.

4. Advertisement/Promotion

The Advertisement/Promotion system resembles the FWBS 4.0. For sending promotional emails to the customer.

I. Pseudocode: FWBS 4.0

```
Restaurant_Database DB;
Enum ad_promo_feedback = {advertisement, promotion, feedback};

protected void btnadvertisepromotion(Mouse_action_listener e, ad_promo_feedback s,
String User_ID)
{
    while( e.selected != differentMenu )
    {
        if ( s == advertisement )
        {
            send_email(advertisement, User_ID);
        }
        else if ( s == promotion )
        {
            if( DB.User.search(User_ID).status == New )
            {
                send_email( promotion, New, User_ID );
            }
            else if ( DB.User.search(User_ID).status != New )
            {
                send_email( promotion, DB.User.search(User_ID).status, User_ID );
            }
        }
        else if ( s == feedback )
        {
            if( DB.User.search(User_ID).status == New )
            {
                //feedback from new customer will create 5% of promotion
                string promo_code = generatePromotionCode( feedback, new);
                send_email( feedback, New, User_ID, promo_code);
            }
            else if ( DB.User.search(User_ID).status != New )
            {
                //otherwise send thank you email for feedback
                send_email( feedback, DB.User.search(User_ID).status,
                    User_ID, "Thank you");
            }
        }
    }
    gotoDifferentMenu( e.selected );
}
```

II. Table No. 1

FWBS No.	Name	Input	Description/Output
4.1.1	New Customer	Check if new customer SC#12	Send Coupon through Email to customer
4.1.2	Existing Customer	Check if existing customer SC#12	Send Thank you email to customer
4.1.4	Customer Feedback	Enter Customer name, email, description SC#12	Fill out the feedback form

III. Screenshot SC #12

Page 1

https://www.irbrookings.com/feedback/write

Home | About | Reservation | Employee Login | [Rate Us!](#)

Please Rate us and get 10% off for next visit!

Name

E-mail

Taste

Service

Comment

[Rate Us!](#)

Work on progress, final outcome may looks not same

IV. Table No. 2

FWBS No.	Name	Input	Description/Output
4.1.3	Send Promotion Email	Click Submit button	Send Coupon through Email SC#14
4.1.4.1	Send Thankyou Email	Click submit button	Send Thankyou feedback Email SC#13

V. Pseudocode

```
protected void btnfeedback(Mouse_action_listener e, string User_ID)
{
    while( e.selected != differentMenu )
    {
        System.print("Your feedback is our great pleasure");

        string taste = RadioButtonList1.getListValue();
        string service = RadioButtonList2.getListValue();
        string comments = Textbox1.getValue();

        if( e.selected == submit_button1 )
        {
            btnadvertisepromotion( feedback, User_ID );
            return ;
        }
    }
    goToDifferentMenu( e.selected );
}
```

VI. Screenshot SC #13, SC #14, SC #15

From

Subject

To

CC

BCC

Thanks for visiting our restaurant very often.

With our gracious thanks to you for often visit of us,

here we send you the promo code that discount 10%
for Heinekin beer you like to order!

Next time you visit please show this mail with promo code on it to get discount!

Code: JMHEIN011
Valid threw: 2018/nov/24~2018/dec/24

From

Subject

To

CC

BCC

Thanks for visiting our restaurant very often.

With our gracious thanks to you for often visit of us,

here we send you the promo code that provide 1 entree for free to use on next visit.

Next time you visit please show this mail with promo code on it to get discount!

Code: JSENTR101
Valid threw: 2018/Sep/24~2018/dec/24

From

Subject

To

CC

BCC

Thank you for leaving the rate about our restaurant
for time you took to rate us, we greatly thanks to you.

Here is the coupon for discount 10% of any order next time visit.

Thank you for your time and See you next time.

5. Reservation System

The Smart Reservation resembles the FWBS 5.0. It helps the customer to reserve a table at the restaurant.

I. Pseudocode: FWBS 5.0

```

Restaurant_Database DB;

protected void btnreservation( Mouse_action_listener e, User_status u )
{
    while( e.selected != differentMenu )
    {
        if( u == guest )
        {
            btnguestreservation();
        }
        else
        {
            btnuserreservation( u );
        }
    }
    gotoDifferentMenu( e.selected );
}

```

II. Table No. 1

FWBS No.	Name	Input	Description/Output
5.1	Online Reservation	Click on Reservation button in the main page. Check if user is logged in. If not direct to login page	Redirect to Reservation webpage SC#16
5.1.1	Enter User Data	Enter username, email, contact, no. of people, time and date	Fill out the Reservation form SC#16

III. Screenshot SC #16

Page 1

https://www.irbrookings.com/reservation

Home | About | [Reservation](#) | Employee Login | Rate Us!

Please Note! If you dont show up by reserved time, we may cancel your reservation

Name:

E-mail:

Phone #:

of People:

Time and Date:

October 2018

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

☐ 10:00am
☐ 11:00am
☒ 12:00 pm
☐ 1:00 pm
☐ 2:00 pm

Submit Reservation

Work on progress, final outcome may looks not same

IV. Table No. 2

FWBS No.	Name	Input	Description/Output
5.1.4.2	Send reservation mail	Click on submit button	Send Email to customer SC#17, SC#18

V. Pseudocode

```
protected void btnguestreservation( Mouse_action_listener e )
{
    while( e.selected != differentMenu )
    {
        string name = Textbox1.getValue();
        string email = Textbox2.getValue();
        string date = Calendar1.getDates.toString();
        string time = Radiobuttonlist1.getList.getList();
        int num_people = Radiobuttonlist2.getList.getList();

        if( e.selected == submit_button1 )
        {
            DB.reservation.new_reservation(name, email, date, time, num_people);
            return ;
        }
    }
    goToDifferentMenu( e.selected );
}

protected void btnuserreservation( Mouse_action_listener e, User_ID )
{
    while( e.selected != differentMenu )
    {
        string name = DB.User.search(User_ID).name;
        string email = DB.User.search(User_ID).email;
        string date = Calendar1.getDates.toString();
        string time = Radiobuttonlist1.getList.getList();
        int num_people = Radiobuttonlist2.getList.getList();

        if( e.selected == submit_button1 )
        {
            DB.reservation.new_reservation(name, email, date, time, num_people);
            return ;
        }
    }
    goToDifferentMenu( e.selected );
}
```

VI. Screenshot SC #17, SC #18

From

Subject

To

CC

BCC

Thank you for making reservation
to dine at International Restaurant of Brookings!
your reservation is following:
Date:2018/oct/04
Time: 01:00 PM
We hope to see you that day!
Thank you!

From

Subject

To

CC

BCC

Thank you for using International Restaurant of Brookings!
We are deeply sorry but your reservation has been canceled
due to not showing up on reserved time.
We deeply appologize for your inconvenience,
we understand your busy schedule to miss the appointment.
We hope to see you again soon!
Thank you.

VIII. FILES AND TABLES

VIII. I Database Tables

1. Customer Details Table

```
create table custlogin  
(id int PRIMARY KEY,  
Name varchar(255),  
Email varchar(255),  
Contact varchar(20),  
Password varchar(255));
```

SANIL\MSSQLSERVER...t - dbo.custlogin						SQLQuery1.sql - SAN...(SANIL\sanil (55))*				SANIL\MSSQLSERVER...t - dbo.custlogin			
	id	Name	Email	Contact	Password					Column Name	Data Type	Allow Nulls	
▶	1	Junmo	jun@gmail.com	6053456123	jun1234					id	int	<input type="checkbox"/>	
	2	Minsup	min@gmail.com	6056267045	min1234					Name	varchar(255)	<input checked="" type="checkbox"/>	
	3	Sanil	sanil@hotmail.	6056912324	san1234					Email	varchar(255)	<input checked="" type="checkbox"/>	
*	NULL	NULL	NULL	NULL	NULL					Contact	varchar(20)	<input checked="" type="checkbox"/>	
										Password	varchar(255)	<input checked="" type="checkbox"/>	
												<input type="checkbox"/>	

2. Employee Details Table

```
create table emplogin  
(id int PRIMARY KEY,  
Name varchar(255),  
Email varchar(255),  
Contact varchar(20),  
Password varchar(255));
```

SANIL\MSSQLSERVE...nt - dbo.emplogin						SQLQuery1.sql - SAN...(SANIL\sanil (55))*				SANIL\MSSQLSERVE...nt - dbo.emplogin			
	id	Name	Email	Contact	Password					Column Name	Data Type	Allow Nulls	
▶	1	Sam	sam@gmail.com	6054321234	sam1234					id	int	<input type="checkbox"/>	
	2	Cherryl	cherry@gmail.c	605908456	cherry1234					Name	varchar(255)	<input checked="" type="checkbox"/>	
	3	Penny	pen@hotmail.c	6057549087	pen1234					Email	varchar(255)	<input checked="" type="checkbox"/>	
*	NULL	NULL	NULL	NULL	NULL					Contact	varchar(20)	<input checked="" type="checkbox"/>	
										Password	varchar(255)	<input checked="" type="checkbox"/>	
												<input type="checkbox"/>	

3. Sales Table

```
create table sales  
(id int PRIMARY KEY,  
Income varchar(255),  
Outcome varchar(255),  
Sales varchar(20),  
Month varchar(255));
```

Column Name	Data Type	Allow Nulls	id	Income	Outcome	Sales	Month
id	int	<input type="checkbox"/>	1	\$50000	\$30000	\$20000	18/01
Income	varchar(255)	<input checked="" type="checkbox"/>	2	\$45000	\$25000	\$20000	18/02
Outcome	varchar(255)	<input checked="" type="checkbox"/>	3	\$63400	\$30200	\$33200	18/03
Sales	varchar(20)	<input checked="" type="checkbox"/>	4	\$30000	\$12000	\$9000	18/04
Month	varchar(255)	<input checked="" type="checkbox"/>	5	\$43000	\$21000	\$29000	18/05
		<input type="checkbox"/>	* NULL	NULL	NULL	NULL	NULL

4. Inventory Table

```
create table inventory
(id int PRIMARY KEY,
Name varchar(255),
LeftStock varchar(255),
MinStock varchar(255));
```

Column Name	Data Type	Allow Nulls	id	Name	LeftStock	MinStock
id	int	<input type="checkbox"/>	1	Chicken	13	10
Name	varchar(255)	<input checked="" type="checkbox"/>	2	Beef	20	20
LeftStock	varchar(255)	<input checked="" type="checkbox"/>	3	Pork	20	10
MinStock	varchar(255)	<input checked="" type="checkbox"/>	4	Potato	30	20
		<input type="checkbox"/>	5	Spices	10	5
			* NULL	NULL	NULL	NULL

IX.I Definition

Definitions and Abbreviations and Acronyms

Accounting

This section describes business rules that apply to handling of accounts payable, accounts receivable, payroll, taxes, utilities, and other expenses.

Accounting Reports

All accounting areas shall have reports that break down and summarize all the data. These reports will run mainly from the sales records by daily, weekly, monthly, annually.

Maintain Accounts Payable

The accounts payable shall be broken out and grouped into sections by vendor.

Maintain Accounts Receivable

The accounts receivables shall be maintained and grouped into sections.

Taxes

A sales tax of 6.5% shall be added to each order.

Credit/Debit card payment

When an order is placed, a credit/debit card should be available to confirm the order. All major credit and debit cards are accepted.

Local phone number

All online orders must supply a local telephone number to facilitate confirmation of the reservation.

Order time frame

Orders may be placed no more than five days in advance.

Employee Schedule

All employees cannot work more than 40 hours per week. In addition, extra work gives more premium bonus that will be charged by payment system.

Distribute Schedule

Employees will be given access to work schedule no less than 7 days in advance.

Maintain Schedule

Only the Top-Manager/Assistant Manager/Owner shall be given the ability to edit or change the work schedule and update or revises the system.

ERD (Entity–Relationship diagram): describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between instances of those entity types

Exception Handling: is the process of responding to the occurrence, during computation, of exceptions – anomalous or exceptional conditions requiring special processing– often changing the normal flow of program execution.

Net Profit (Income): is a definition that is calculated by

$$NP = \text{Money Earned} - \text{Expenditures (Cost of Goods Sold + Expense + Taxes)}$$

Integrated Testing: A software testing where individual units are combined and tested as a group. This test is done after Unit Testing is finished

Enterprise resource planning (ERP): is the integrated management of core business processes, often in real-time and mediated by software and technology.

Customer relationship management (CRM): is an approach to manage a company's interaction with current and potential customers.

POS (Point of Sale): The point of sale (POS) is the time and place where a retail transaction is completed.

CTR (Click Through Rate): is the ratio of users who click on a specific link to the number of total users who view a page, email, or advertisement.

Unit Testing: A software development process where parts of an application is tested in details

Data Dictionary: This is a designing tool that describes the structure of a database and the relationship between its elements

Data Flow Diagram: A graphical representation of data through an information system.

FWBS: Functional Work Breakdown Structure – Overall works are breakdown as per its function. It's a top down modularity; divides overall the task into small pieces.

Critical Path Analysis: A method to analyze the time (maximum) taken by project which helps in cost estimation.

Encryption: A security mechanism that encrypts plain text input cipher text (encrypted text) which makes impossible to understand it without decryption.

Architecture Design: Internal design of our system which show how our system works.

Gantt Chart: It's a table showing all the details of activities and time taken by each activity.

Remote cloud service: It's an online service that provides data storage and backup remotely.

Miles: Accumulate the customer's score and give some exciting discounts and / or prizes based on score.

Gift Card: It's a virtual money; customer can use in our system.

External Device: High storage, high speed device used to back up our data for security and portability.

Firewall: A system to prevent unamortized access to a network.

Database: Well organized collection of data to facilitate storage, retrieval, modification, and deletion of data.

User: Someone who interacts with our software.

Administrator: System administrator who is given specific permission for managing and controlling the system.

Backup system: Backup refers to copying records or information into some media so that in case of loss of records, we can recover easily. It enhanced information security.

RD: Requirement Documentation

H/W: Hardware

S/W: Software

I/O: Input and Output data

Authentication: Check if username and or password is valid or not.

POS: Point of Sale

System overview: Internal design of our system which show how our system works.

IX.II. Meeting Log

· Group Meeting Log

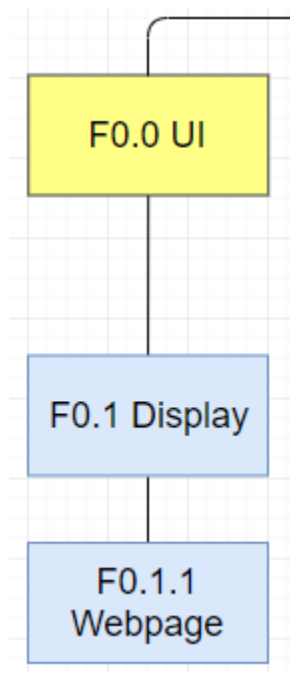
Date	Participants	Agenda
Oct 4, 2018	<ul style="list-style-type: none">• MinSup Kim• Junmo Kim• Sanil Khamkar	<ul style="list-style-type: none">• Revise Proposal• Revise Requirement Documentation• Task allocation
Oct 5, 2018	<ul style="list-style-type: none">• MinSup Kim• Junmo Kim• Sanil Khamkar	<ul style="list-style-type: none">• Discussion for design documentation specification• Select the language and how to make program code for webpage• Review individual work• Task allocation
Oct 6, 2018	<ul style="list-style-type: none">• MinSup Kim• Junmo Kim• Sanil Khamkar	<ul style="list-style-type: none">• Discussion about presentation contents• Review individual work• Task allocation
Oct 10, 2018	<ul style="list-style-type: none">• MinSup Kim• Junmo Kim• Sanil Khamkar	<ul style="list-style-type: none">• Update Proposal for presentation• Update Requirement Documentation for presentation• Make the functionality of five main deliverables• Review individual work• Task allocation
Oct 11, 2018	<ul style="list-style-type: none">• MinSup Kim• Junmo Kim• Sanil Khamkar	<ul style="list-style-type: none">• Revise and finalize the functionality of five main deliverables• Make the presentation template• Discuss about files and tables• Review individual work• Task allocation
Oct 12, 2018	<ul style="list-style-type: none">• MinSup Kim• Junmo Kim• Sanil Khamkar	<ul style="list-style-type: none">• Discuss about design priority• Discuss about data flow diagram and data dictionary• Update the presentation template• Working on Design legend• Working on Top level design

		<ul style="list-style-type: none"> • Working on Medium level design • Review individual work • Task allocation
Oct 13, 2018	<ul style="list-style-type: none"> • MinSup Kim • Junmo Kim • Sanil Khamkar 	<ul style="list-style-type: none"> • Finalize the presentation template • Rehearsal the presentation • Working on Design legend • Working on Top level design • Working on Medium level design • Review individual work • Task allocation
Oct 14, 2018	<ul style="list-style-type: none"> • MinSup Kim • Junmo Kim • Sanil Khamkar 	<ul style="list-style-type: none"> • Rehearsal the presentation • Working on Design legend • Working on Top level design • Working on Medium level design • Review individual work • Task allocation
Oct 16, 2018	<ul style="list-style-type: none"> • MinSup Kim • Junmo Kim • Sanil Khamkar 	<ul style="list-style-type: none"> • Rehearsal the presentation • Working on Design legend and Implementation • Working on Top level design and Implementation • Working on Medium level design and Implementation • Review individual work • Task allocation
Oct 17, 2018	<ul style="list-style-type: none"> • MinSup Kim • Junmo Kim • Sanil Khamkar 	<ul style="list-style-type: none"> • Rehearsal the presentation • Design legend Implementation and revise • Top level design Implementation and revise • Medium level design Implementation and revise • Review individual work • Task allocation
Oct 18, 2018	<ul style="list-style-type: none"> • MinSup Kim • Junmo Kim • Sanil Khamkar 	<ul style="list-style-type: none"> • Write Design legend document • Write Top level design document • Medium level design document • Update FWBS • Update files and tables • Update meeting logs • Review individual work • Task allocation

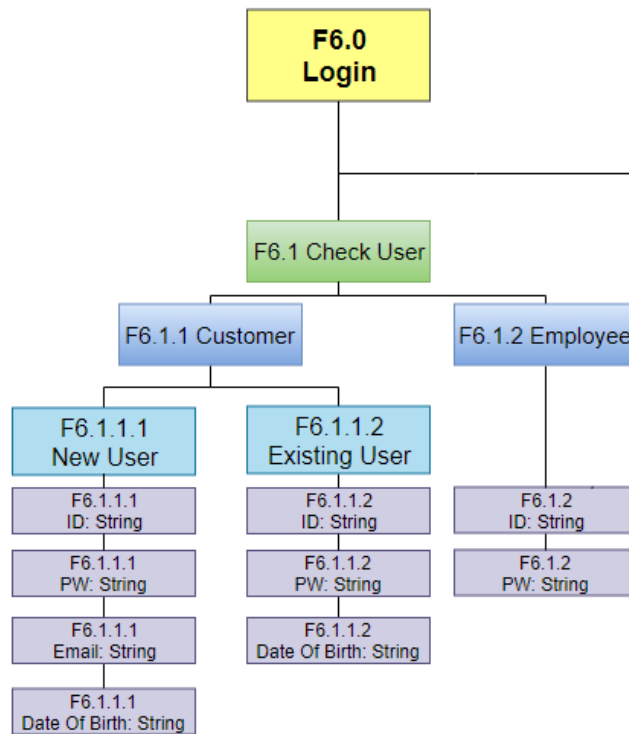
Oct 19, 2018	<ul style="list-style-type: none"> • MinSup Kim • Junmo Kim • Sanil Khamkar 	<ul style="list-style-type: none"> • Revise and Write Design legend document • Revise and Write Top level design document • Revise and Write Medium level design document • Update Proposal • Update Requirement Documentation • Review individual work • Task allocation
Oct 20, 2018	<ul style="list-style-type: none"> • MinSup Kim • Junmo Kim • Sanil Khamkar 	<ul style="list-style-type: none"> • Revise and Write Design legend document • Revise and Write Top level design document • Revise and Write Medium level design document • Update Proposal • Update Requirement Documentation • Review individual work • Task allocation

IX.III. [FWBS](#)

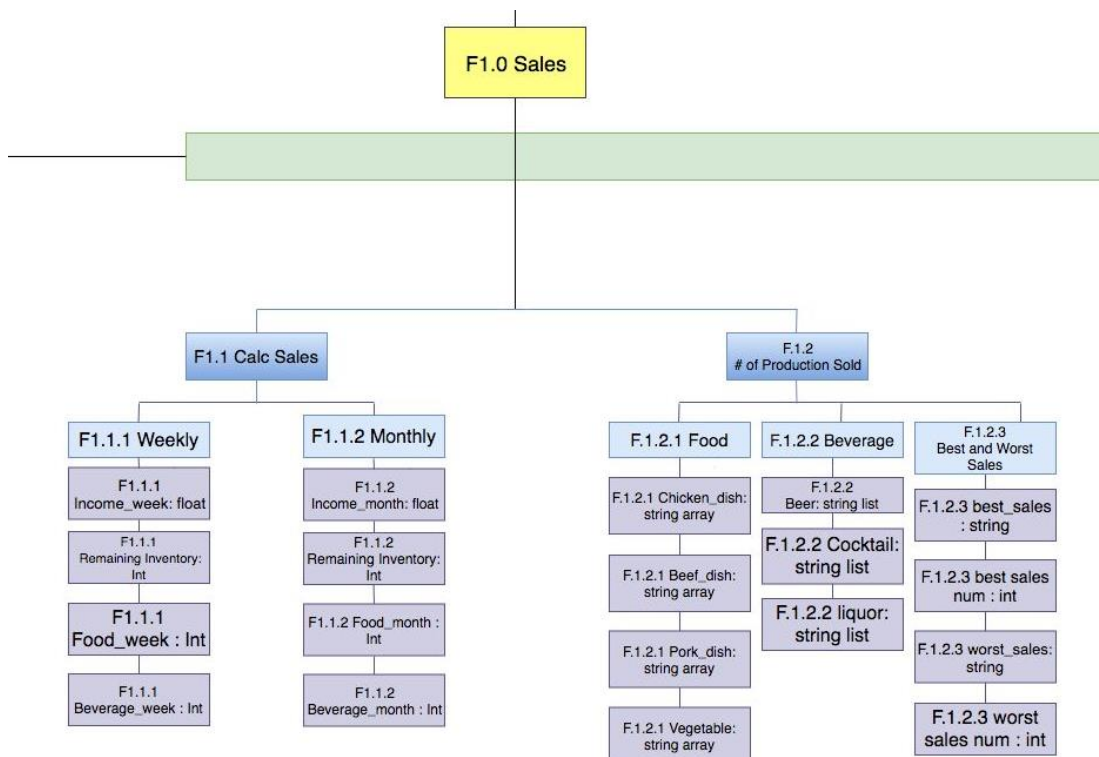
1. User Interface



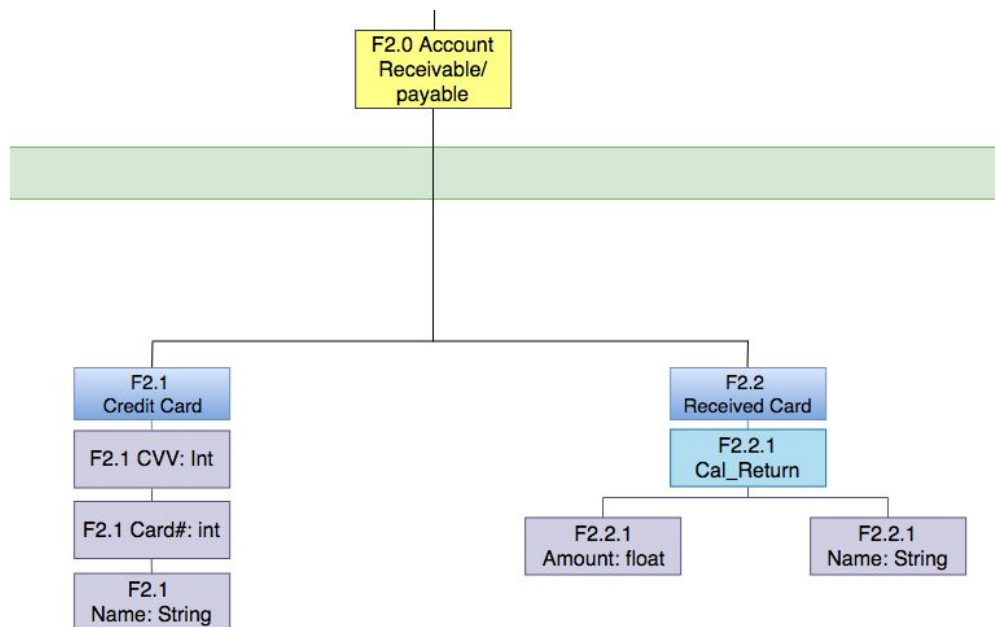
2. Login



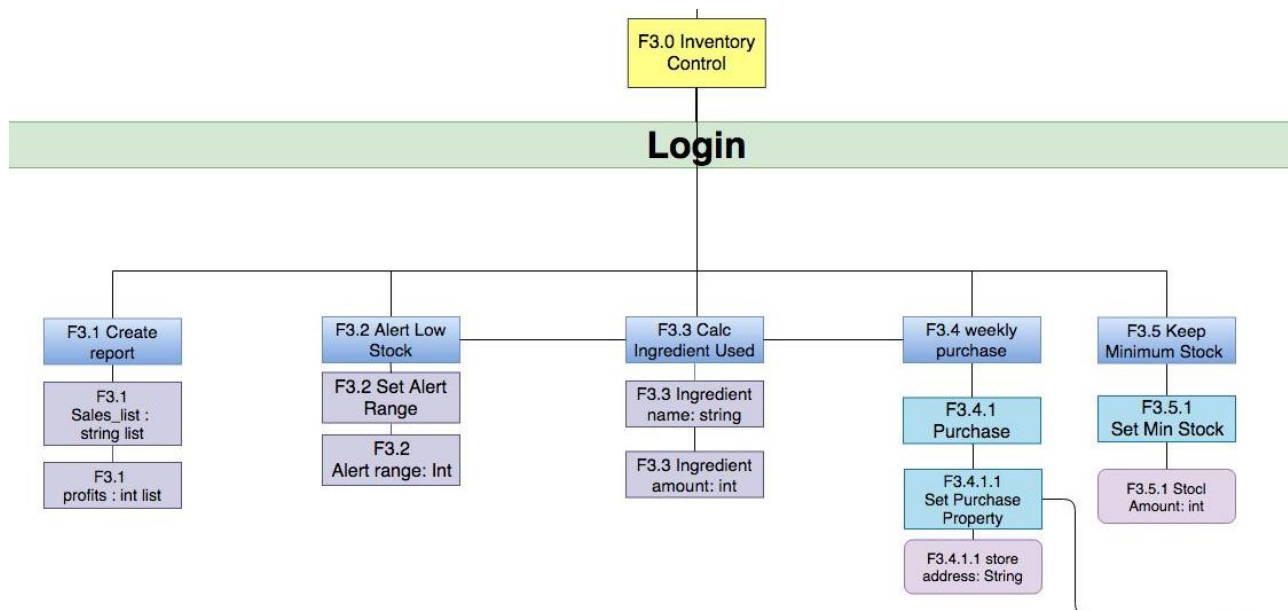
3. Sales



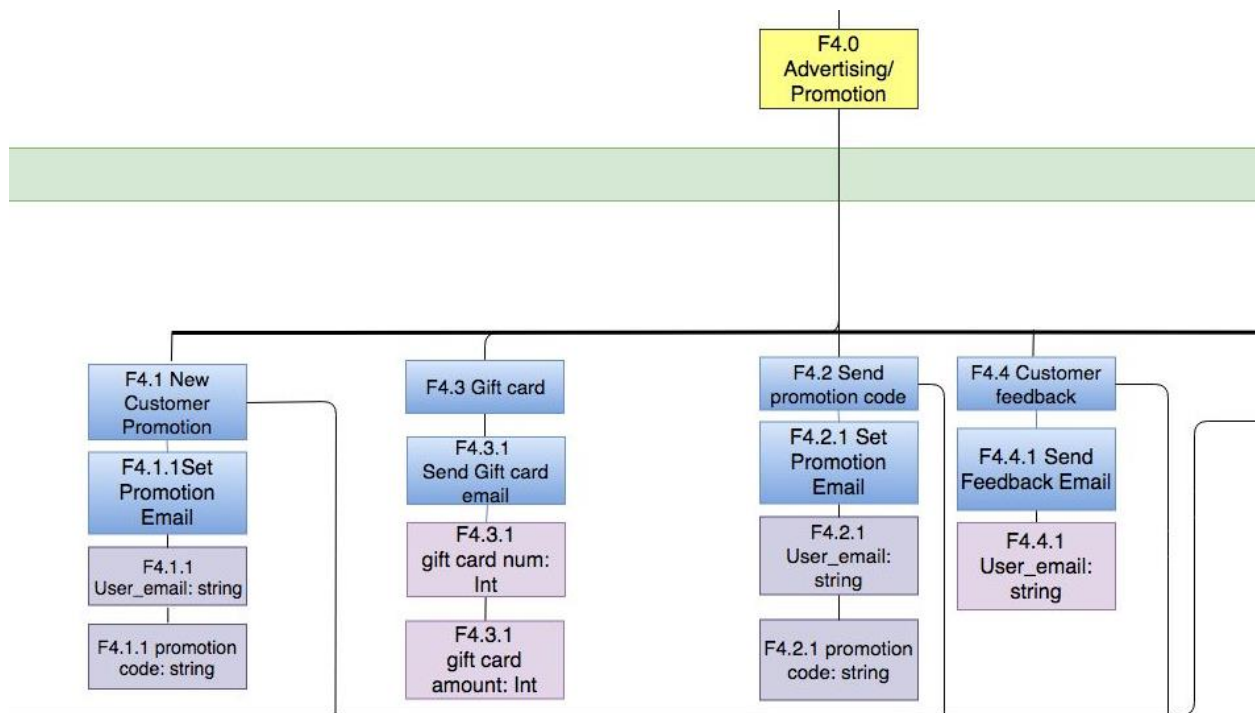
4. Account Receivable/Payable



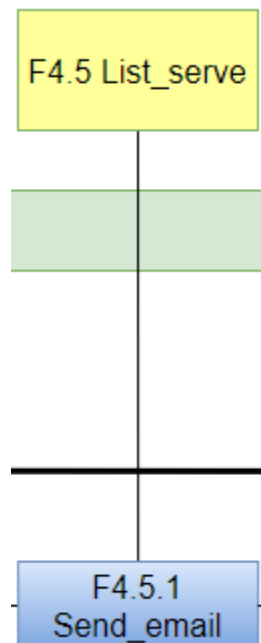
5. Inventory Control



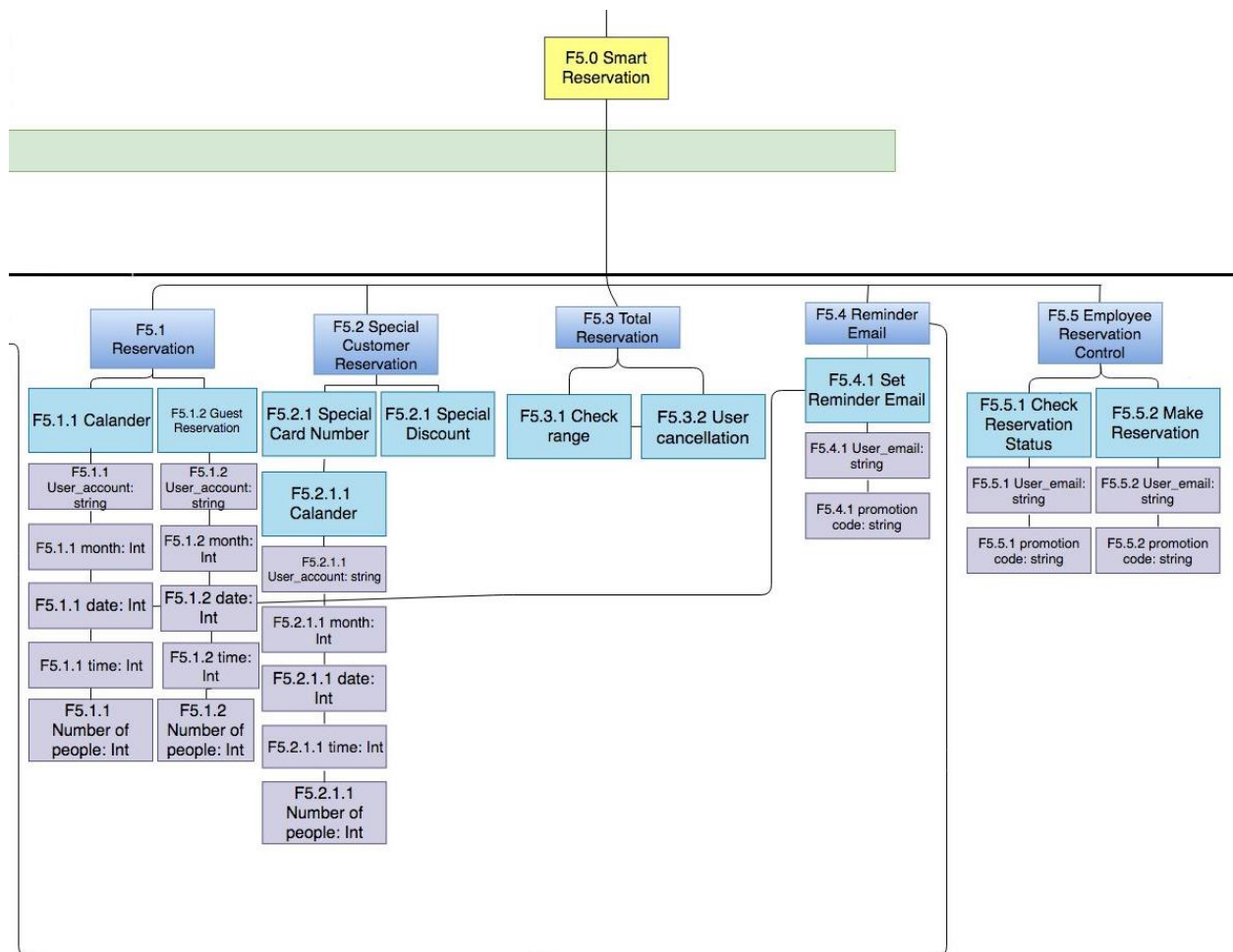
6. Advertisement/Promotion



7. Email



8. Smart Reservation



IX.IV References

1. <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>
2. <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/exceptions/exception-handling>
3. <https://plan.io/blog/ultimate-guide-to-implementing-agile-project-management-and-scrum/>