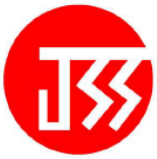


JSS Company

Division of Software Sales

Version 2.0.0

REQUIREMENT DOCUMENTATION

Sangwon Shin, Junmo Kim, Sanil Khamkar

JSS COMPANY

TABLE OF CONTENT

I. INTRODUCTION

I.I PURPOSE OF DOCUMENT

Requirements Document is written in order to describe how the product and services should function based on the agreement between the customer and developer.
The main purpose of requirement document is to help prevent software project failure.

The types of requirements are:

1. User requirements.
    - The software application must be developed according to the user requirements with all the operations constraints.

2. System requirements.
    - The application provides a simple user interface to perform an action.
    - The user is provided with facilities for a particular operation.

3. Functional requirements.
    - The application provides a login system for administrator and staff to have access to the system.
    - Providing weekly and monthly report on sales.
    - Keeping track of transactions done via credit and debit cards.
    - Keeping track of stock using the inventory control.
    - Sending promotional offers.
    - Making online reservations.
    - Providing a VIP system for regular customers.

4. Non-functional requirements.
    - Performance requirements to achieve work in a minimum time span.
    - Software reliability for failure free software operation.
    - Backup security when a fault or error in the software occurs.
    - Maintaining the software for modification, upgrading for performance improvements.
    - Providing portability of software on different platforms.

I.II MAJOR PROBLEMS AND PROJECT GOAL

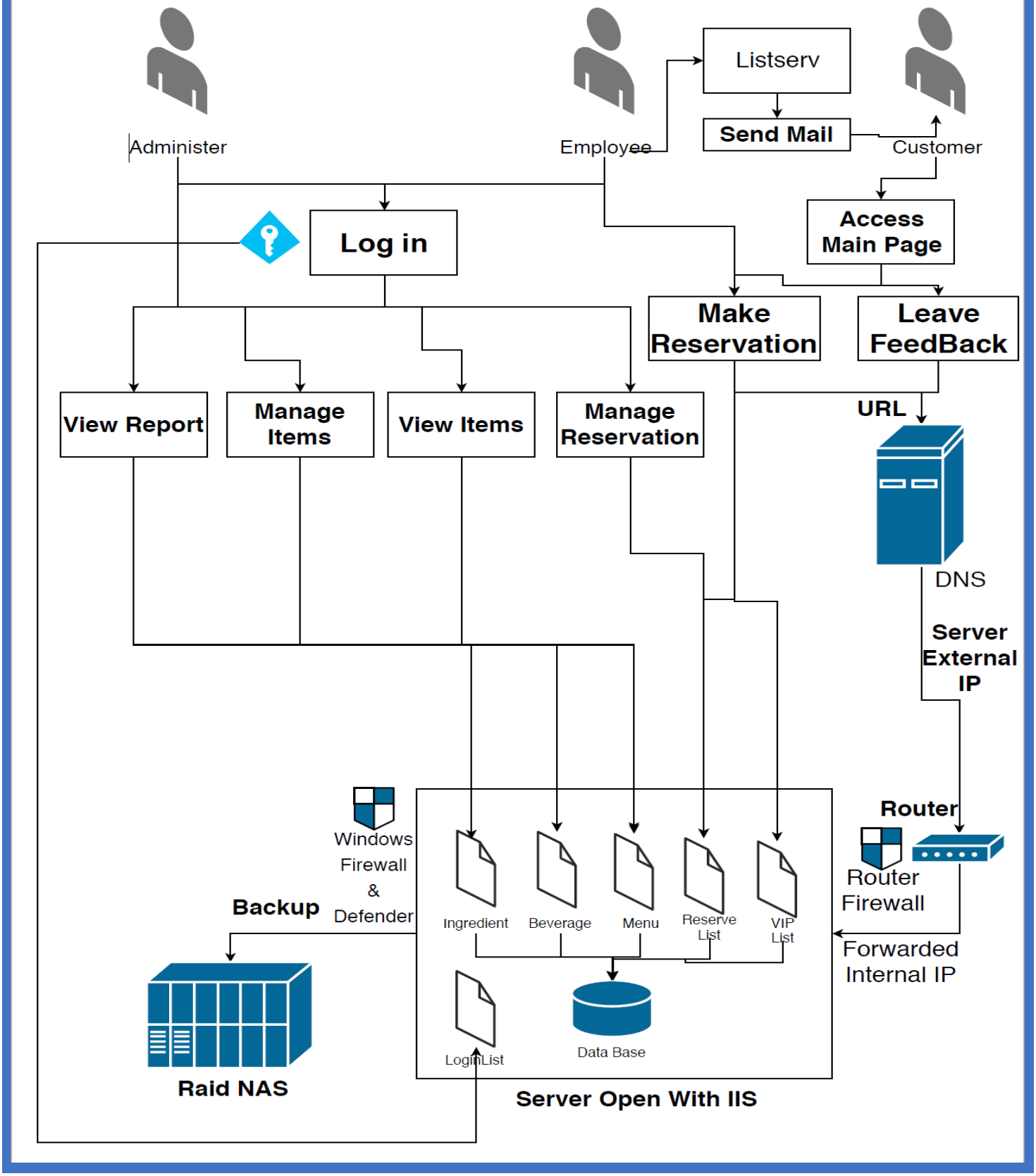
Problems:

- 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.

Project Goals:

The goal of the 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.III PROPOSED SYSTEM OVERVIEW AND CONFIGURATION CHART



I.IV DEFINITIONS, ACRONYMS AND ABBREVIATIONS.

1. Cloud Server: Cloud server refers to digital data storage on a virtual infrastructure.

2. Software: Software refers to collection of data or instructions that are executed on a computer to perform some work.

3. Flaw: Flaw means error.

4. RAID – 1: Raid-1 is a technique that stores the copy of a data on separate disks that is two or more copies of each block are stored on different disks.

5. Bug: Bug refers to error or flaw.

6. Portability: The ability to be easily moved.

7. Activity graph: An Activity Graph shows the flow of activities in a system.

8. Flow chart: A Flowchart is a type of diagram that represents an algorithm, workflow or process.

9. Source code: A list of instructions compiled together to execute a program.

10. Interactive: Having effect on each other.

11. State diagram: Used to describe behavior in diagrams.

12. Hardware: Collection of physical parts of a computer system.

13. Interface: A device or program enabling a user to communicate with a computer.

14. Administrator: A person responsible for running a business.

15. FWBS: Functional Work Breakdown Structure.

II. GENERAL DESCRIPTION

II.I PRODUCT PERSPECTIVE

The software package of managing restaurant will not only limit the use in only one restaurant but also can be use on most of food related business-like Café and fast food chains. However, our system will be targeted on use in International Restaurant of Brookings. Our product will support of view reports, managing inventory, Manages reservation and customer data. Our product will be able to access from any machine that have web browser allows wide accessibility and easy to use.

II.II MAJOR FUNCTIONS AND FEATURES

Our system provides many features and functions for make business more easy and smarter for more profit. Most functionalities are automated for easy ness however most automated function is also intercept able by administrator of system for manual overriding of jobs.

System performs 5 Main category jobs as following:

1. Create reports
   This functionality allows owner or administrator to see how much profit and how inventory is managed. With clear report provided with form of monthly and weekly helps owner what to do for next strategy of business.

2. Manage Items
   Restaurant will have many things to manage for their items. Inventory control is very sensitive and important that link directly into the profit restaurant make. System provides automated managing of items by calculating use of ingredient on ordered menu and order if stock hits minimum amount that owner had set in beginning. This will allow reducing the loss in ingredients, makes owner save more money.

3. Advertisement
   Owners of restaurant often do advertisement and promotion to get more customer. However, sending same promotion to new customer and existing VIP customer is not fair. Our VIP system allows more smarter way of advertising your store.

4. Reservation
   Customer don't like complicated reservation. All our system ask is their name, phone number, date to reserve and time to reserve. By doing this system allows track of customer and analyze individual customer to optimize status of customer to restaurants benefit.

5. Auto inventory order system
   Restaurants whom have contract with food supplier can have fully automated ordering method with auto completed mail sent to supplier. If some supply runs low than usual, system will automatically order ingredient for you. This process is also can be intercepted by administrator for special case for stocking.

6. Payment system
   If you have one system in store, why do you want more to do similar job, our system allows you to keep credit card payment profile and cash calculation.

II.III USER CHARACTERISTICS

Our software defines 3 following users for the system: Administrator, Employee, and Customer.

Administrator

- Administrator is personal who have access authority to manipulate inventories control, employee database, customer database, and VIP customer database
- Administrator has access authority to manage reservation system by customer and reservation by VIP customer.
- Administrator has authority to give promotions for customers
- Administrator has access authority to user database and manipulate user's VIP level
- Administrator has access authority to purchase more ingredients.
- Administrator has ID begins with "admin-" + restaurant number + administrator's freely chosen 5 letters of alphabets and digits.

Employee

- Employee is group of personal who have authority to handle user's purchase of menu
- Employee can access to database and send purchased menu data to database
- Employee can access to database and send purchased beverage data to database.
- Employee can access to reservation customer list and check.
- Employee has ID begins with "irb-" + restaurant number + initial of employee's name.

Customer

- Customer is group of personal who have authority to make reservation, control reservation date and time, cancel reservation
- Customer's purchase history will be recorded in database for determines customer's VIP level.
- Customer's VIP level will provide several upgraded services like grace time for late reservation, more promotion chances, and more discount amount of purchase.

II.IV GENERAL CONSTRAINTS

Our software has 3 following constraints categories: design, process, and implementation

Design constraints

- We expected to keep customers' data of purchase history without creating user accounts.
- The data of purchase by user will be stored if and only if the payment method related with credit
- The purchase history based on credit card will be encrypted and only administrator can decrypt the data and read only.

Process constraints

- In process of send promotion code to user, our software only can send the promotion email to email in the user database.
- Once smart reservation system will be applied, it takes time to store data to pick which user is in VIP list and black list.

Implementation constraints

- After our management system applied, then the previous system data cannot be compatible with some format of user information or employee database.

II.V ASSUMPTIONS AND DEPENDENCIES

Assumptions

- Assumed there are 30% of total seats reserved for customer reservation.

Dependencies

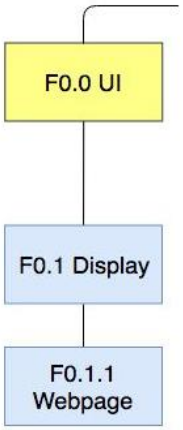
- The determination of VIP level upgrade is based on credit card payment by customers
- Sending promo code or advertisement email is depends on customer feedback after meal or when customer makes reservation. Both requires to input email address.

III. SOFTWARE REQUIREMENTS

III.I FUNCTIONAL REQUIREMENTS

III.I.I FUNCTIONAL REQUIREMENT DESCRIPTION (MOCK UPS)

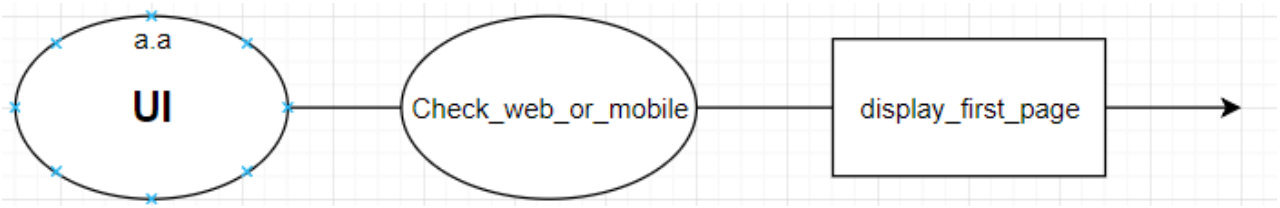
FWBS 1.0



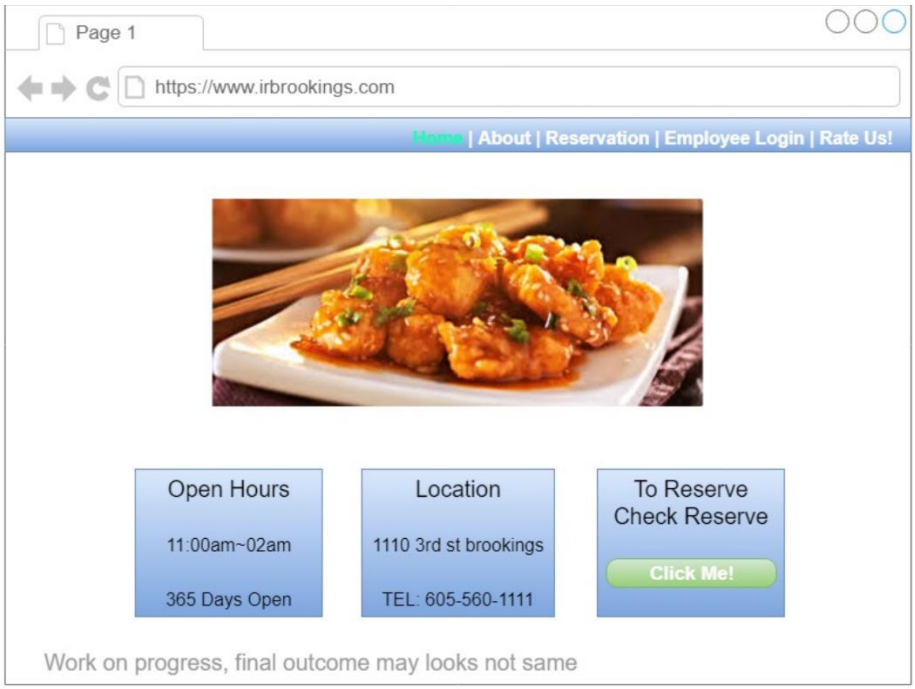
FWBS	Function Name	Input	Output
1.0	Display_first_page	None	UI on screen

Function Description: Display_first_page function is printout the first webpage of IRS management system.

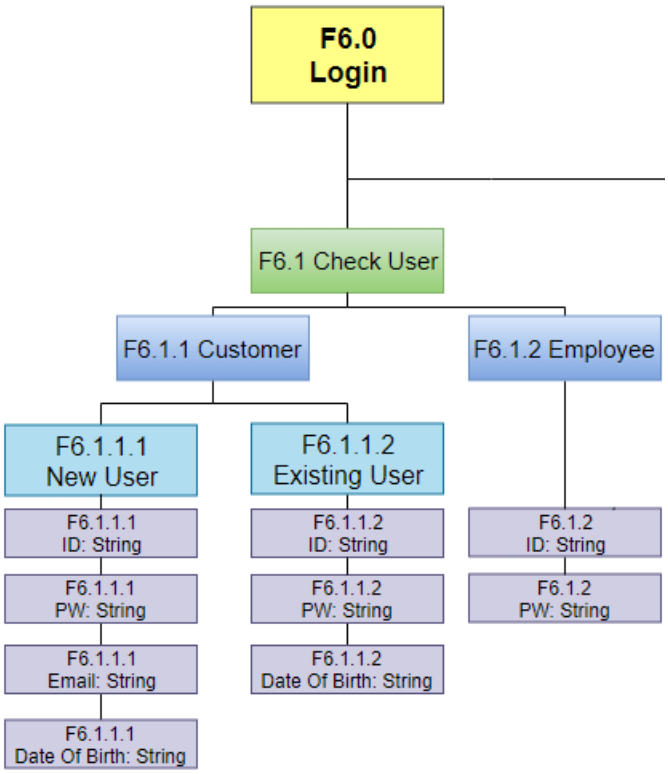
Function state diagram:



Function mock-up



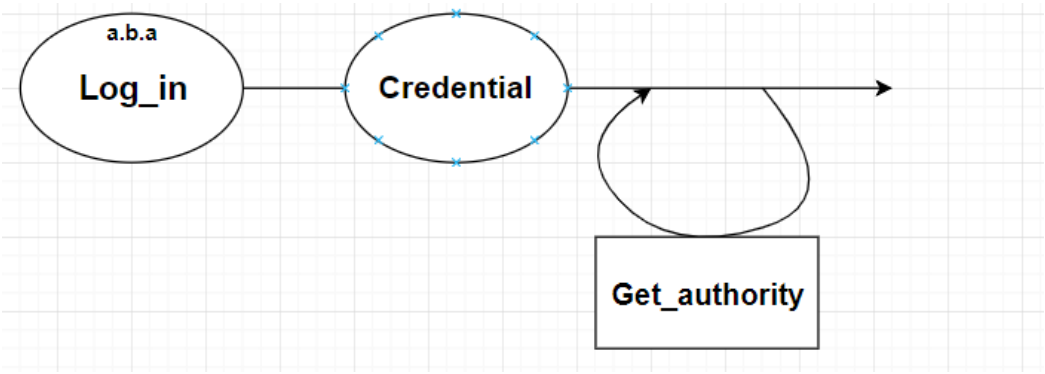
FWBS: 6.0



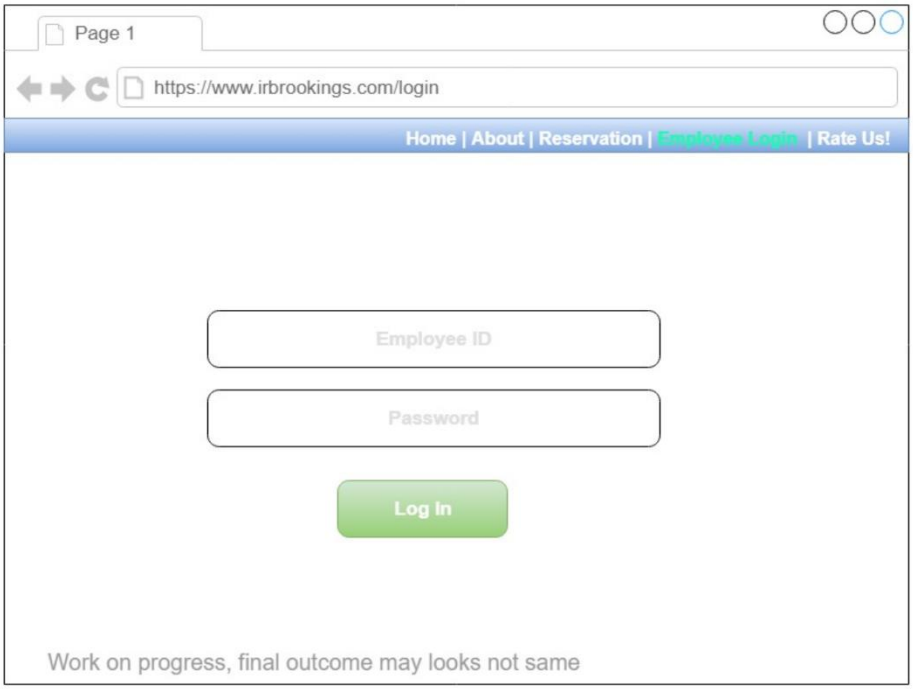
FWBS	Function Name	Input	Output
6.0	Credential	String: ID, PW	Enum Authority

Function Description: Credential is check the employee ID and password to the employee database and find the authority to match with ID and password.

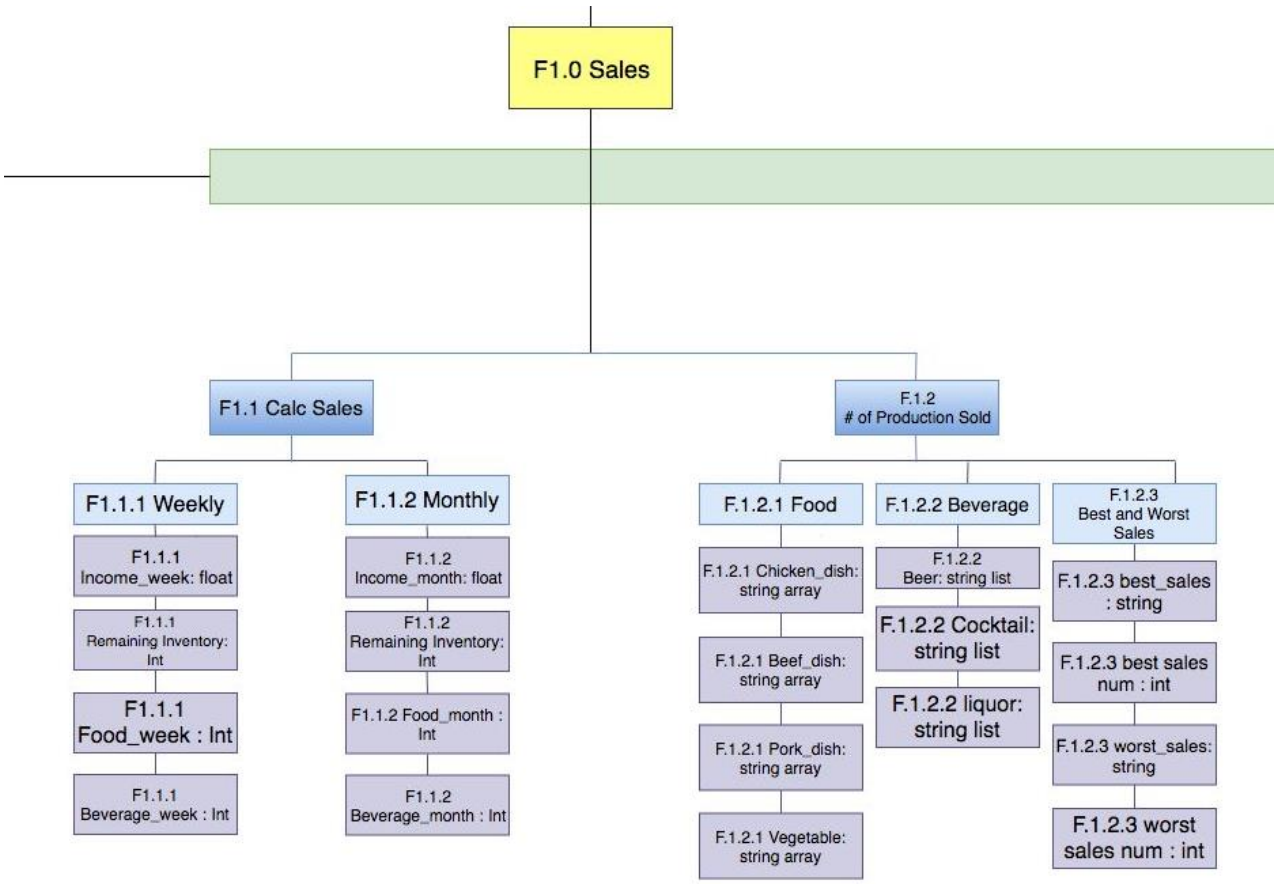
Functional state diagram:



Function mock-up



FWBS: 1.0



FWBS	Function Name	Input	Output
1.1	CalcSales	None	None

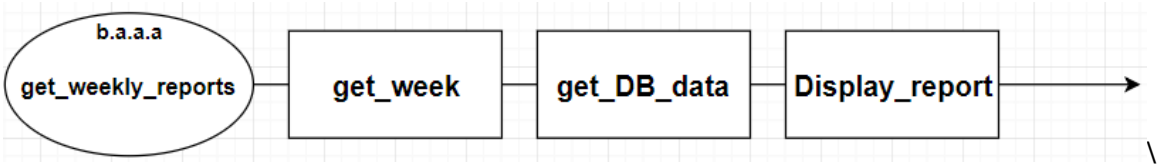
Function description: CalcSales is function to create the sales report based on weekly sales data and monthly sale data

FWBS: 1.1.1

FWBS	Function Name	Input	Output
1.1.1	Weekly	None	Float:Income, Outcome, Sales Date: Week

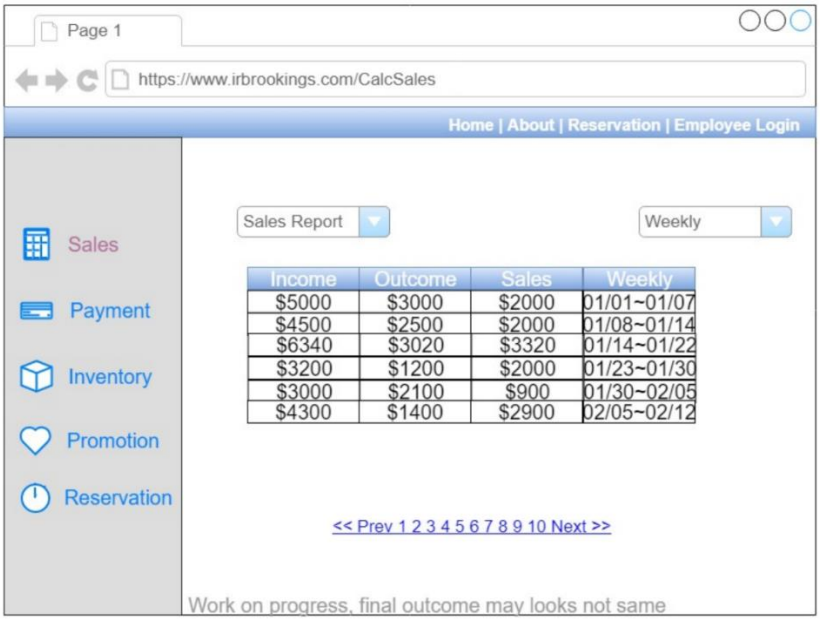
Function Description: Weekly is sub function of CalcSales function that create sales report based on weekly sales data

Function state diagram:



\

Function mock-up:

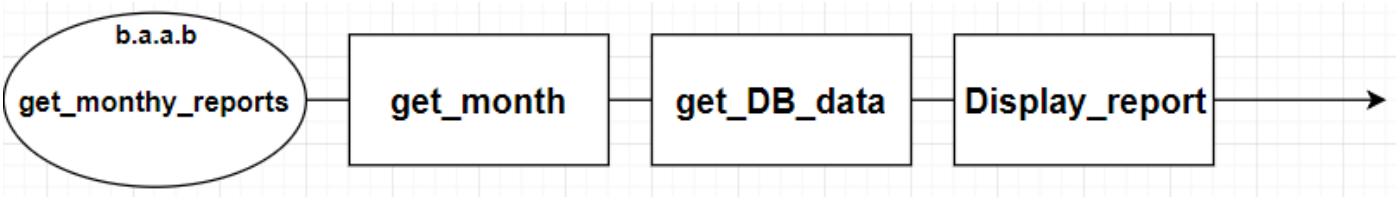


FWBS: 1.1.2

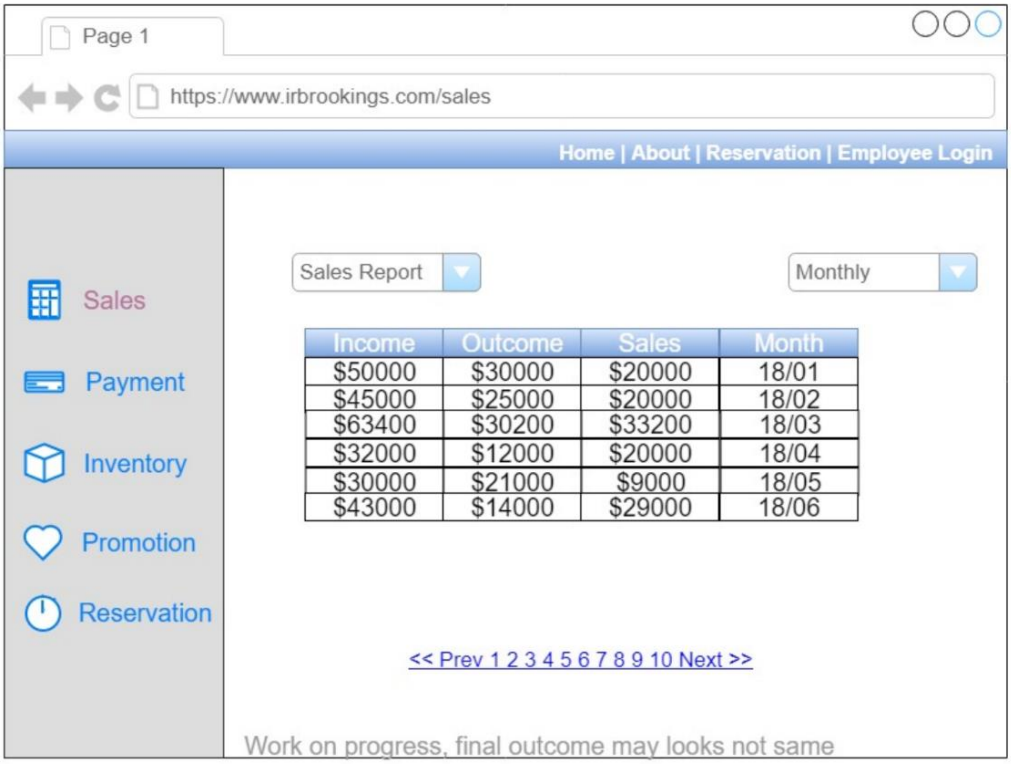
FWBS	Function Name	Input	Output
1.1.2	Monthly	None	Float:Income, Outcome, Sales Date: Month

Function description: Monthly is sub function of CalcSales function that create sales report based on monthly sales data

Function state-diagram:



Function mock-up:



FWBS: 1.2

FWBS	Function Name	Input	Output
1.2	Production_Sold	None	None

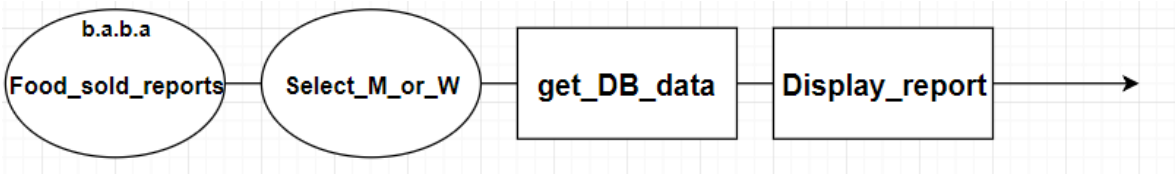
Function description: Production_Sold is a function that can show the report of weekly and monthly purchase data about food and beverage.

FWBS: 1.2.1

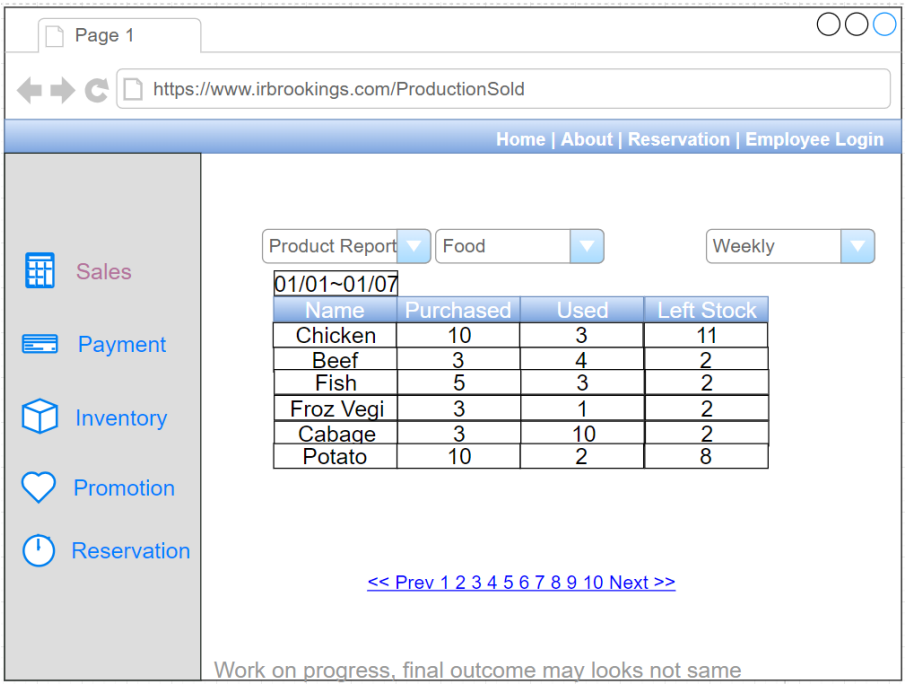
FWBS	Function Name	Input	Output
1.2.1	FoodWeekly	None	String: FoodName, int: SoldAmount Float: Profit

Function description: FoodWeekly is function to show sales report about purchased food data.

Function state-diagram:



Function mock-up

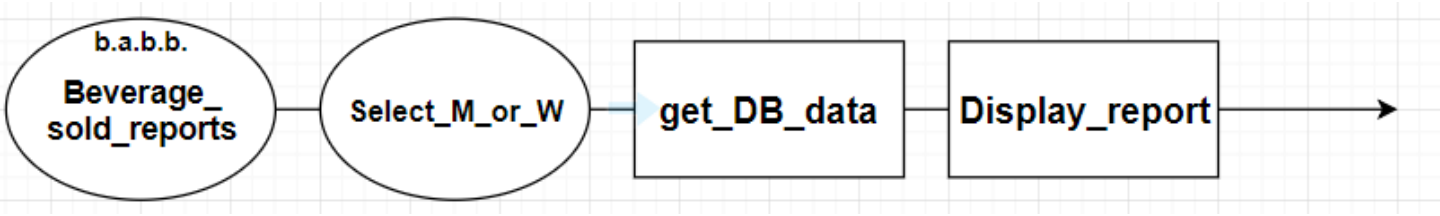


FWBS: 1.2.2

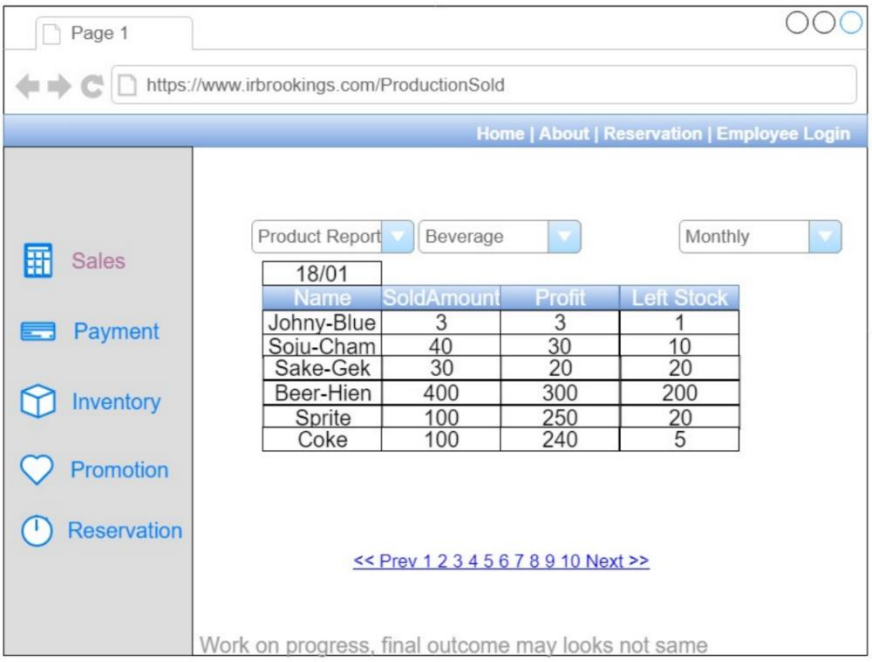
FWBS	Function Name	Input	Output
1.2.2	BeverageWeekly	None	String: BeverageName int: SoldAmount, LeftStock Float: Profit

Function description: BeverageWeekly is function to show the sales report about purchased beverage data.

Function state-diagram:



Function mock-up:

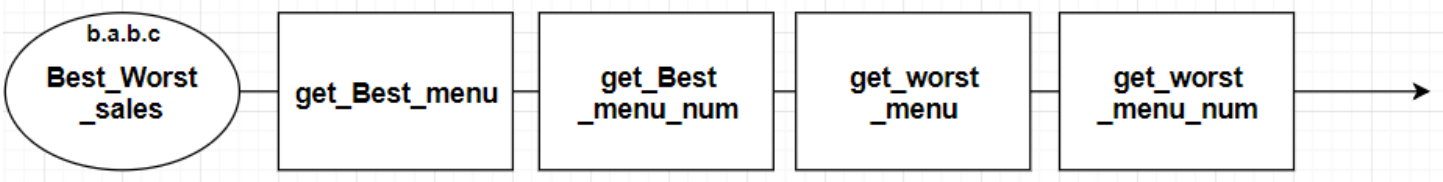


FWBS: 1.2.3

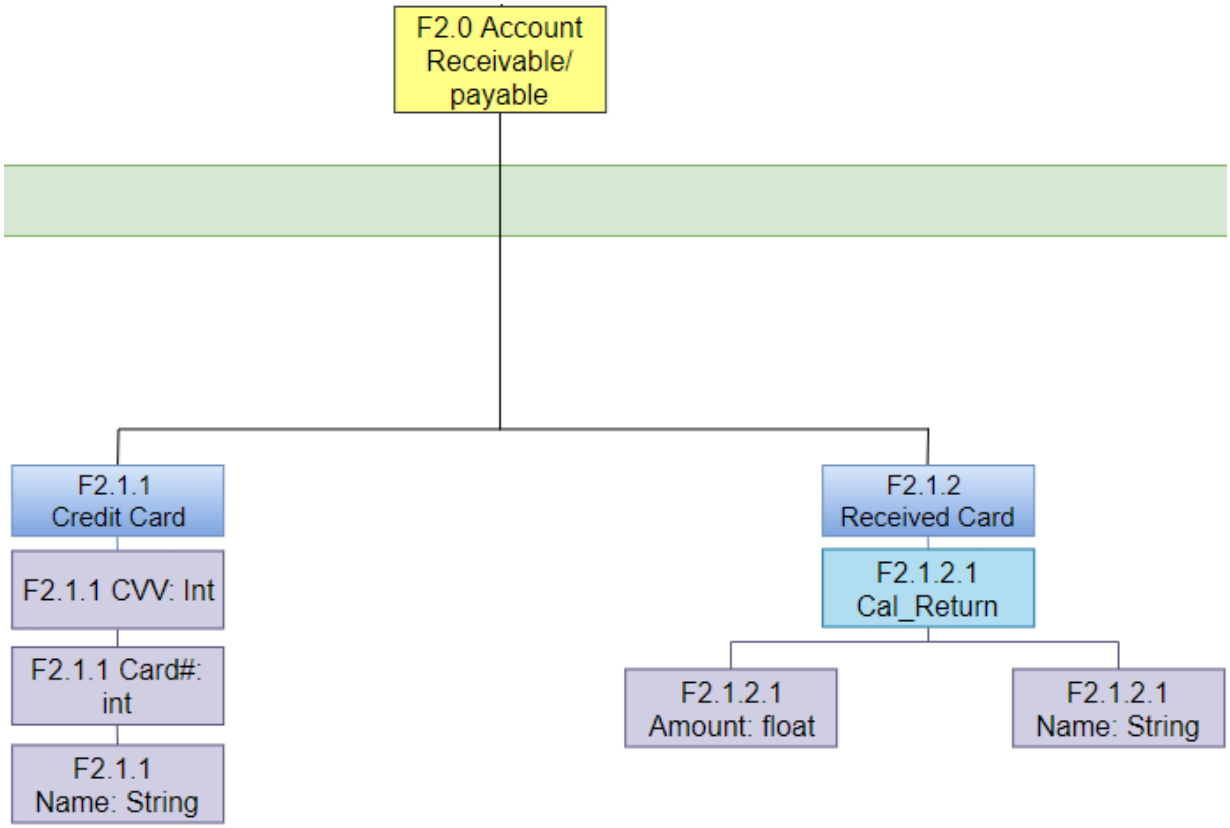
FWBS	Function Name	Input	Output
1.2.3	BestnWorstSales	None	String:BestSales Int: BSoldAmount String:WorstSales Int: WSoldAmount

Function description: BestnWorstSales is function to show food or beverage that sold most or least in weekly sales report.

Function state-diagram:



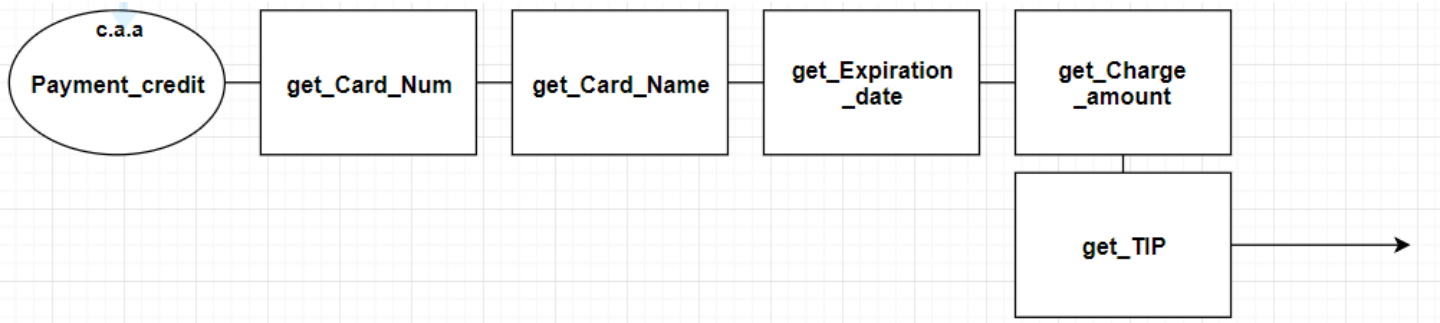
FWBS: 2.0



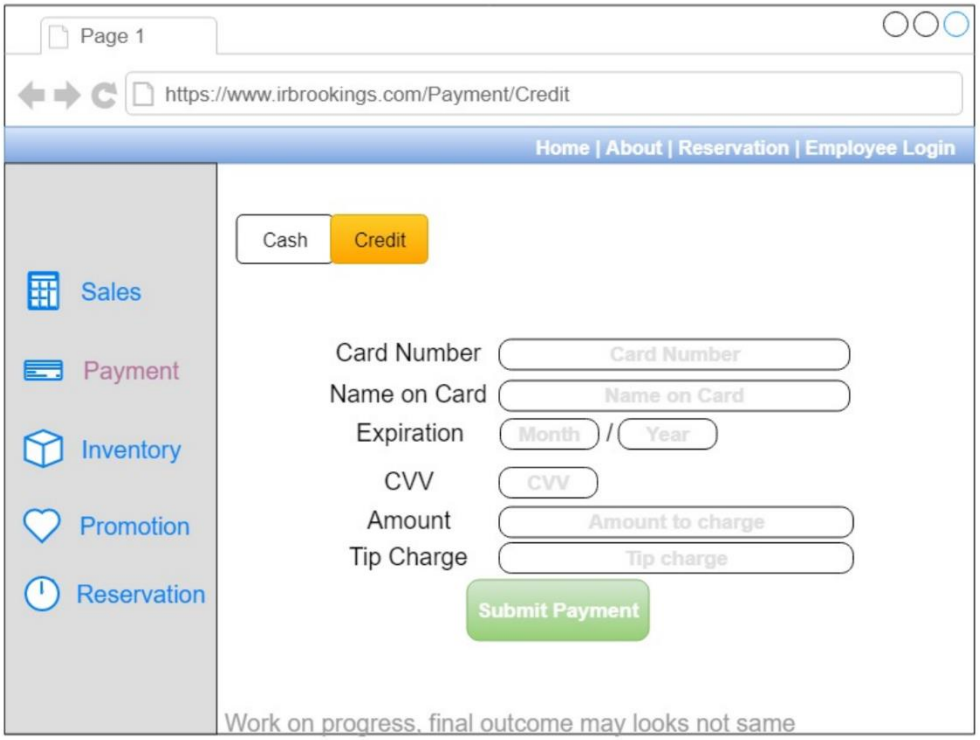
FWBS	Function Name	Input	Output
2.1	CreditCardPay	Int:CardNum, CVV, Expiration Float:Amount, Tip String: PayerName	None

Function description: CreditCardPay is a function to receive customer's card data, verify that is correct data, and purchase the money that customer needs to pay.

Function state-diagram:



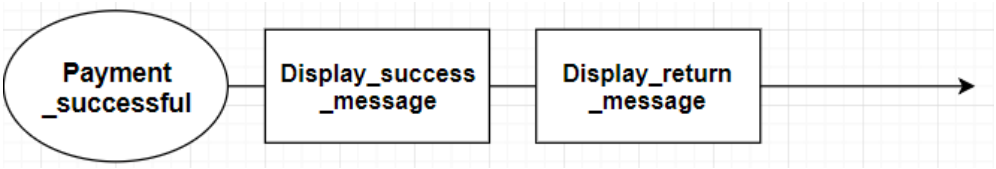
Function mock-up:



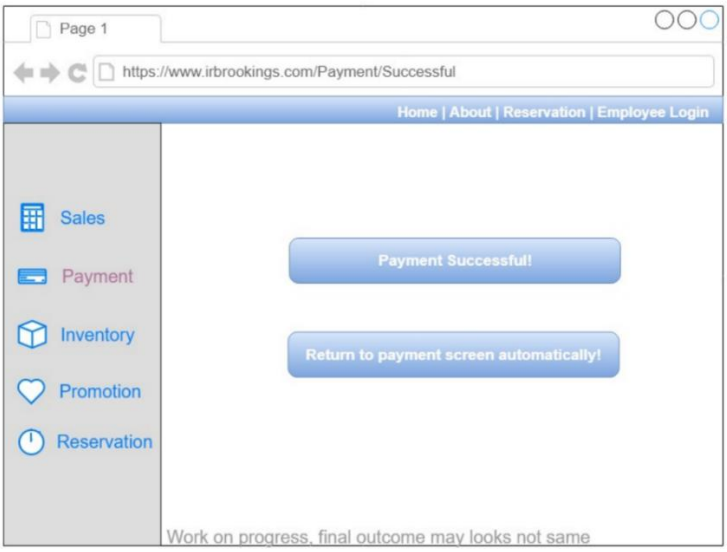
FWBS: 2.1.1

Function description: Payment_Successful is a function that when customer's purchase is done successfully.

Function state-diagram:



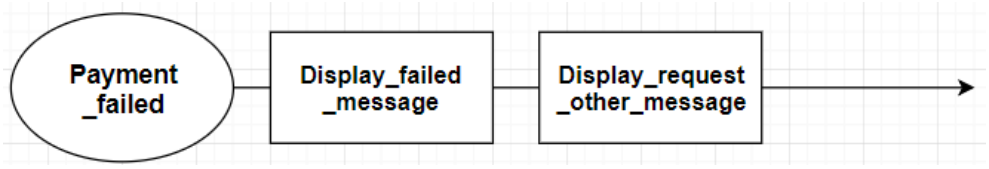
Function mock-up:



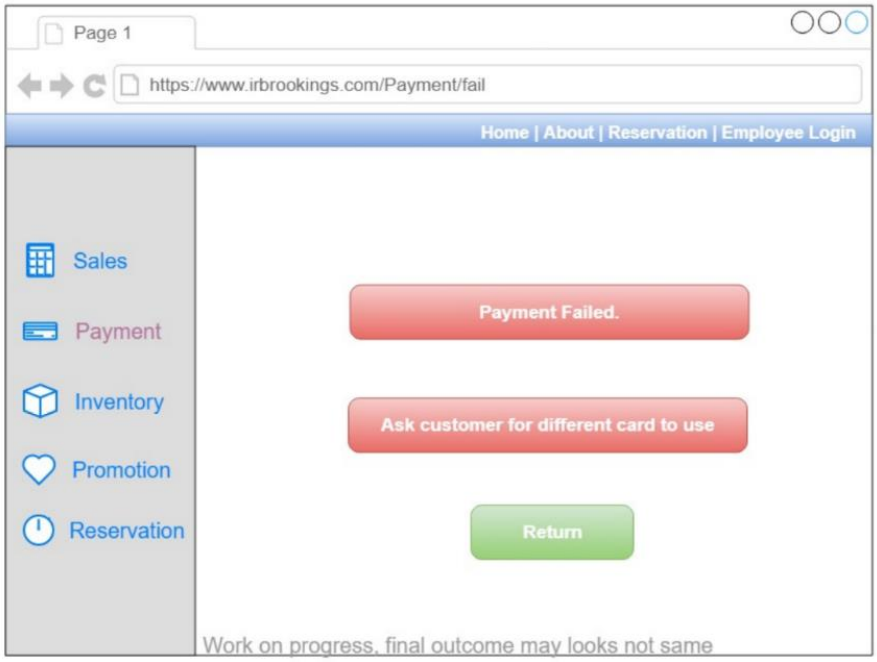
FWBS: 2.1.2

Function description: Payment_failed is a function that when customer's purchase is fail.

Function state-diagram:



Function mock-up:

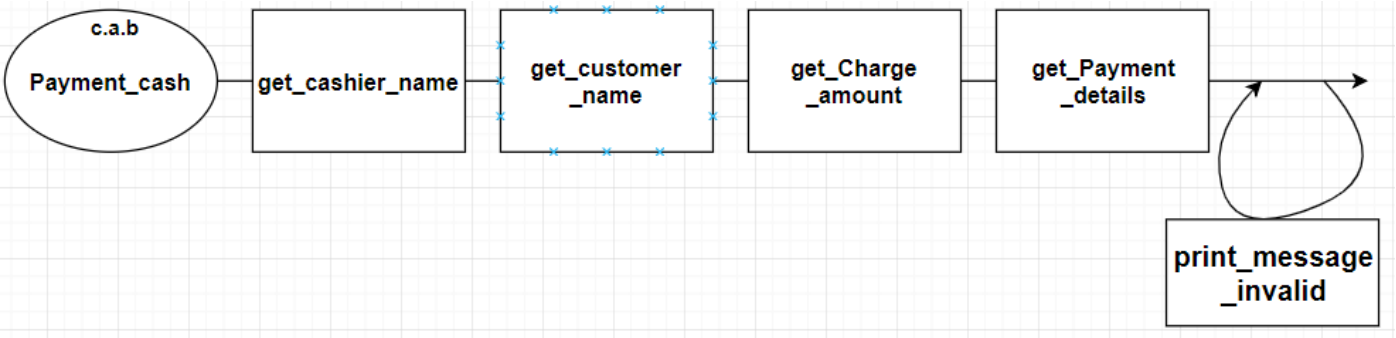


FWBS: **2.2**

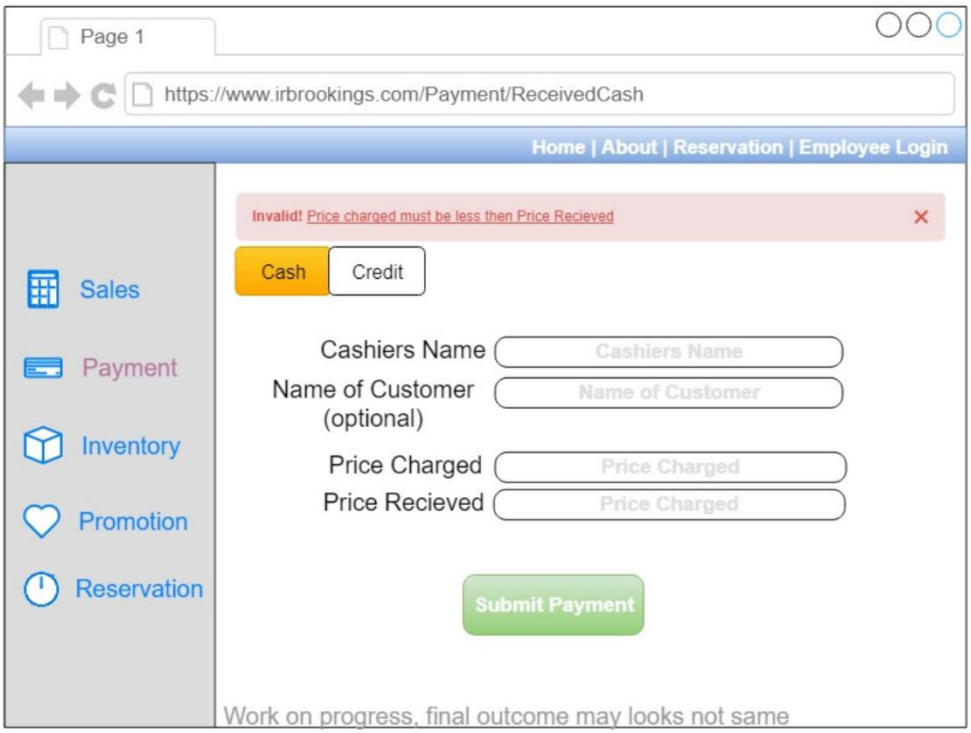
FWBS	Function Name	Input	Output
2.2	CashPay	Float: Amount, String: Name	String: CName, CashierName Float:Charged, Recieved

Function description: CashPay is function when user decided to pay by cash.

Function state diagram:



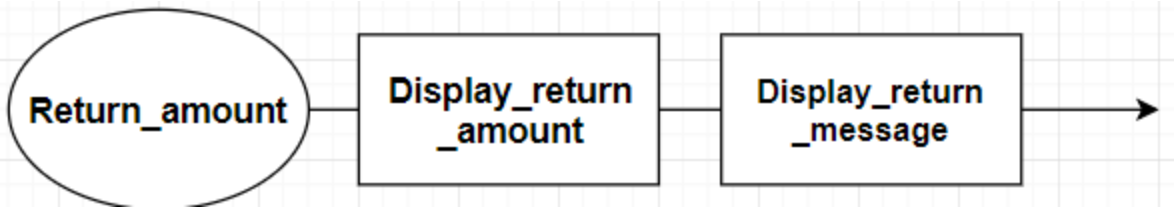
Function mock-up:



FWBS: 2.2.1

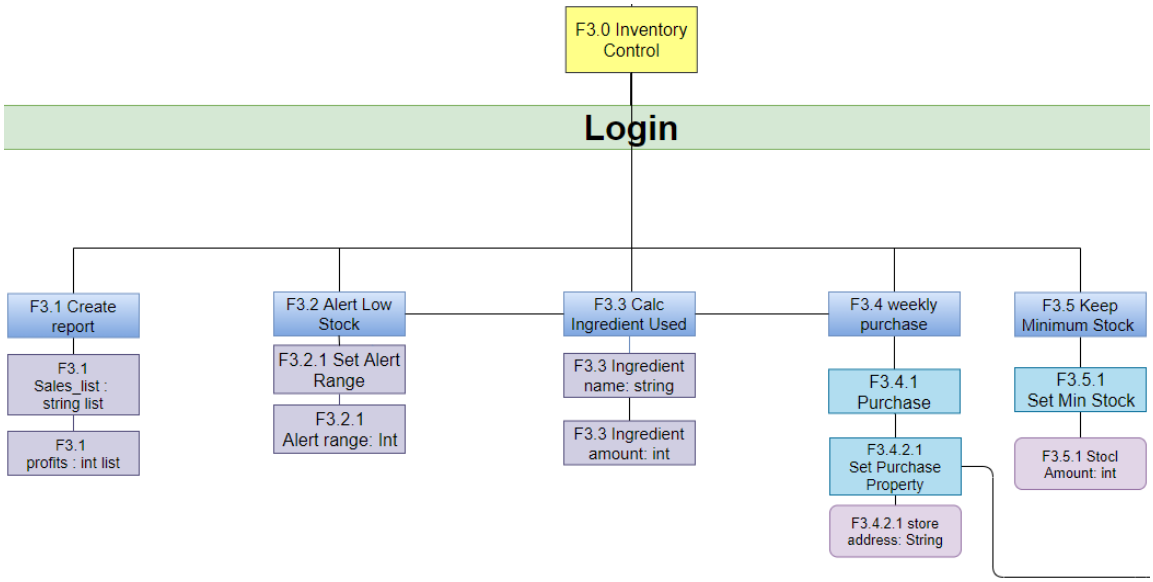
Function description: ReturnSuccess is a function run after user finished cash pay and print the amount of money return by register.

Function state diagram:



Function mock-up:

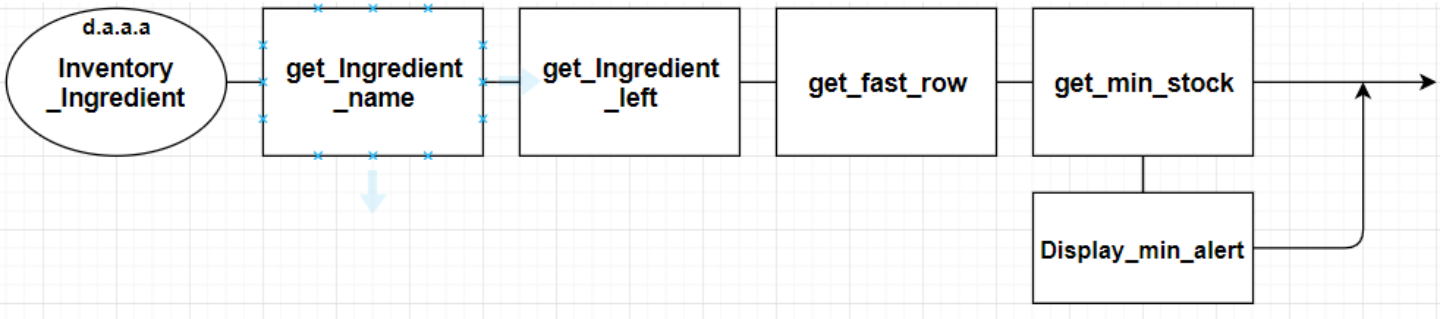
FWBS: 3.0



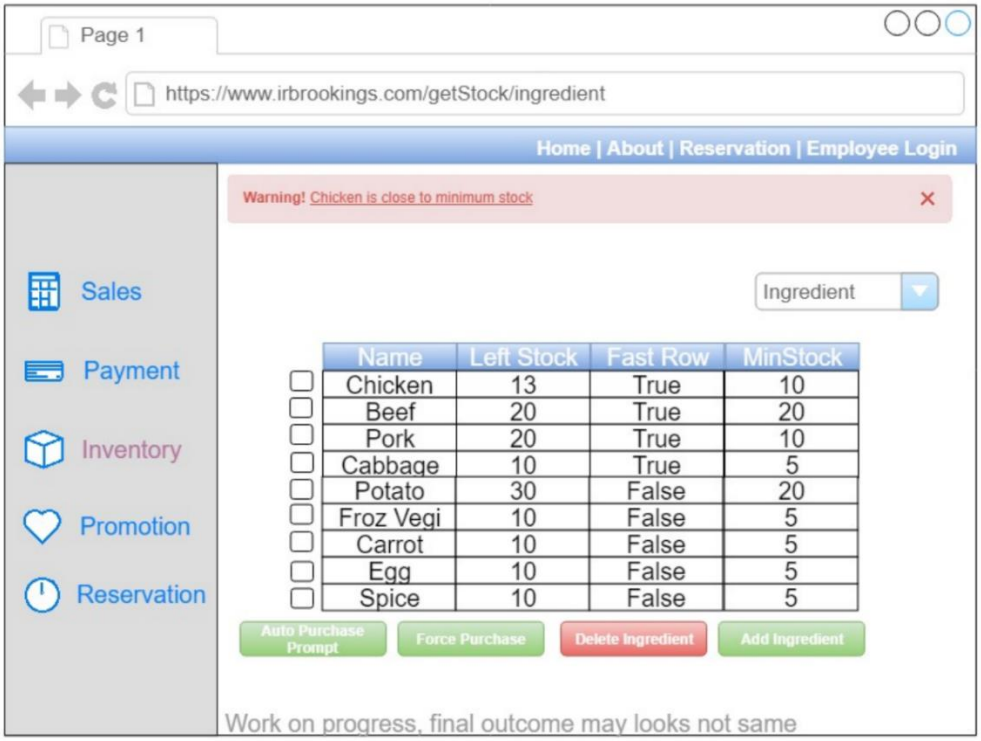
FWBS	Function Name	Input	Output
3.0	InventoryIngerd	None	String: InName, Int: leftStock, MinStock Bool: FastRow

Function description: InventoryIngred is function that print out the stock of ingredient from database.

Function state-diagram:



Function mock-up:

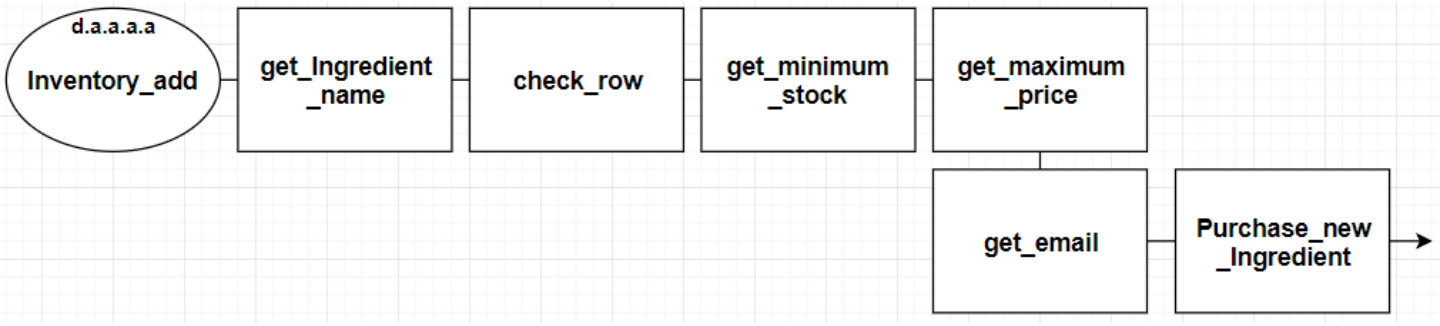


FWBS: 3.0

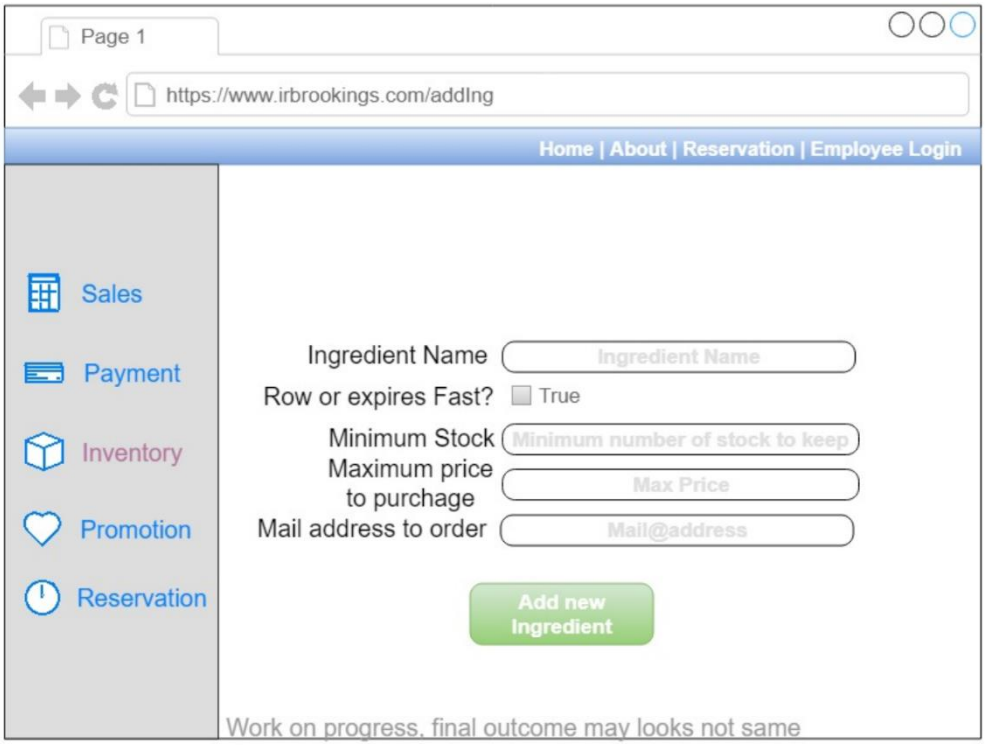
FWBS	Function Name	Input	Output
3.0	addIng	String:IngName, IngOrderMailAdd, Int: minStock Float:MaxPrice Bool: RowItem	None

Function description: addIng is function to purchase input amount of ingredient based on input name by user.

Function state-diagram:



Function mock-up:

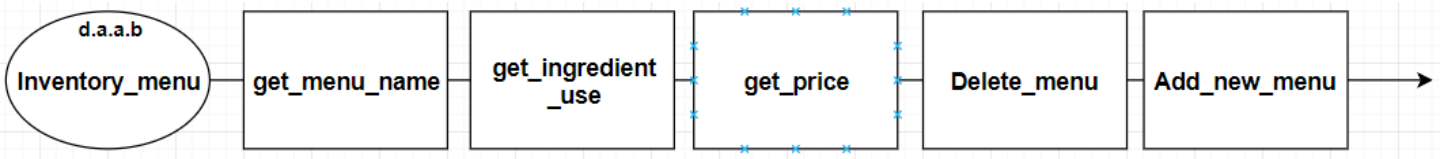


FWBS: 3.0

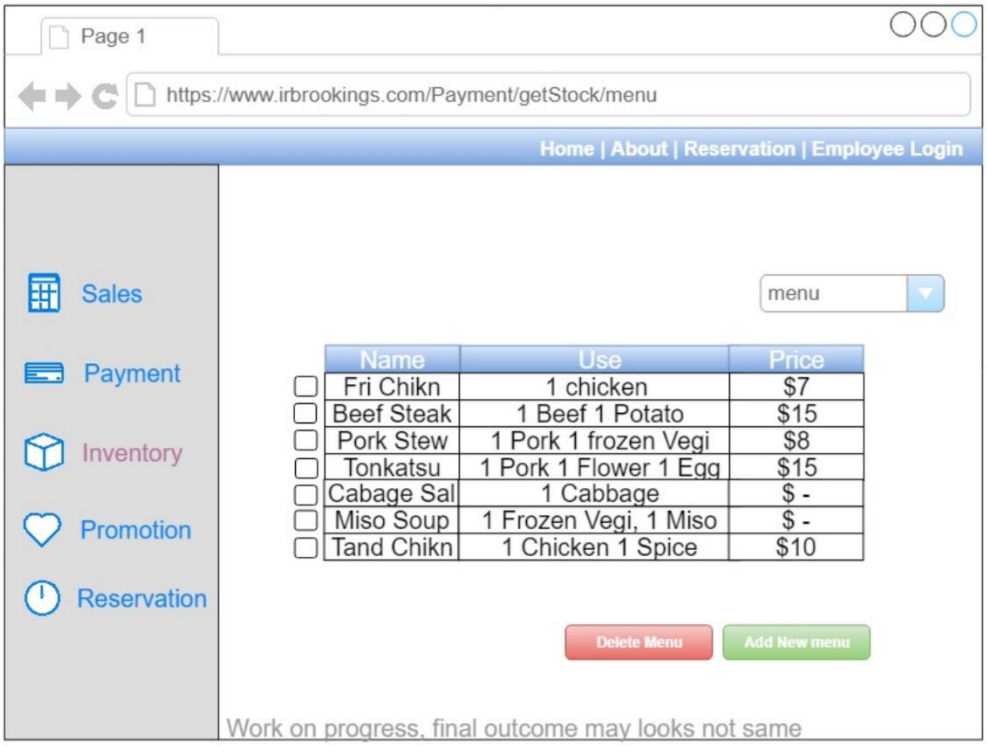
FWBS	Function Name	Input	Output
3.0	InventoryMenu	None	String: MenuName, IngName Int: UseAmount Float:Price

Function Description: InventoryMenu is function displays menu's name, ingredient uses, and price. Also it has sub function as delete menu and add menu.

Function state diagram:



Function mock-up:

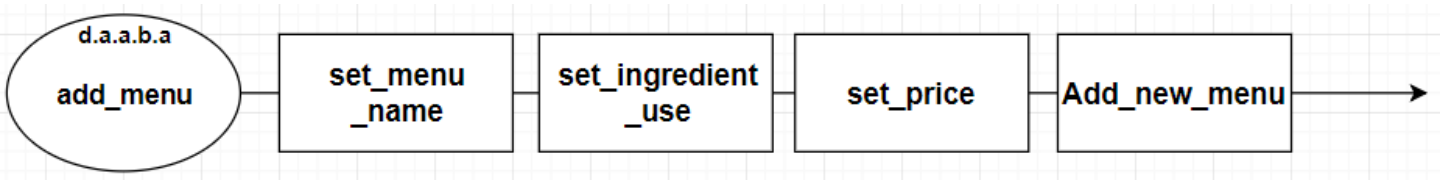


FWBS: 3.0

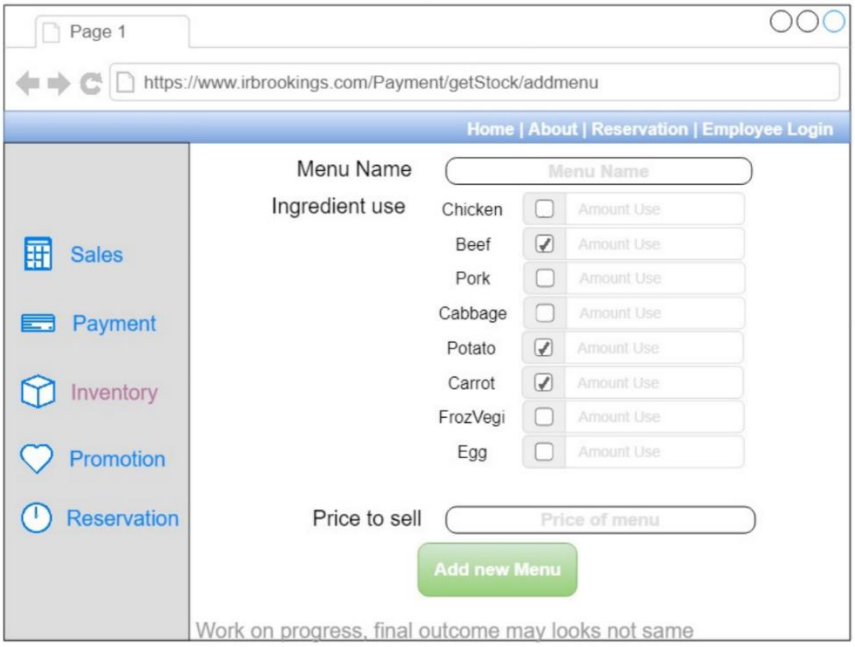
FWBS	Function Name	Input	Output
3.0	addMenu	String: MenuName, IngName Int: UseAmount Float:Price	None

Function description: addMenu is function that add new menu into menu database by manager. It requires menu name, ingredients names, ingredient amount of use, and price.

Function state diagram:



Function mock-up

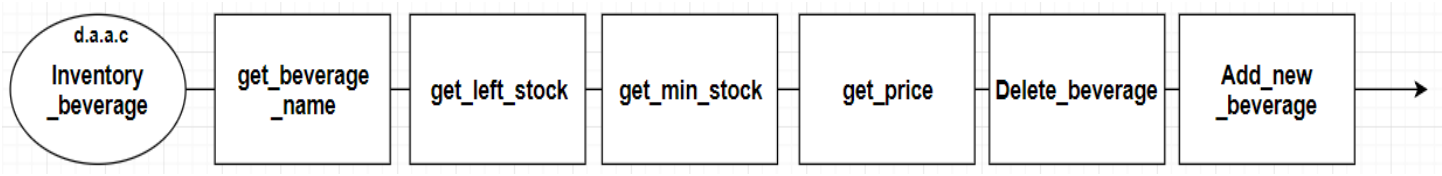


FWBS: 3.0

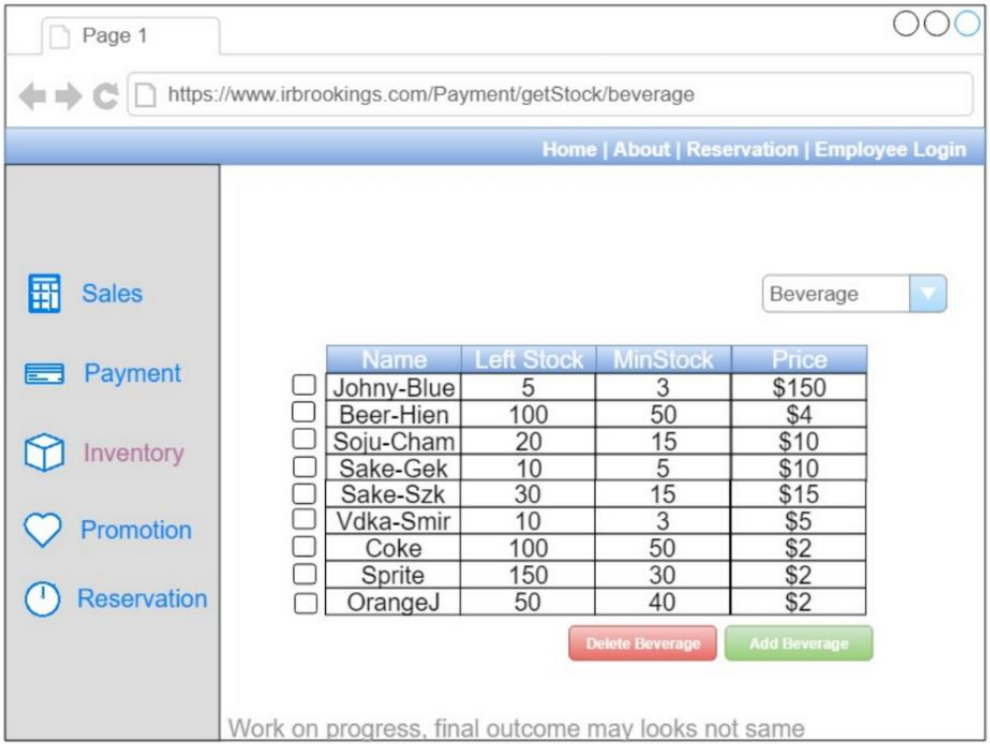
FWBS	Function Name	Input	Output
3.0	InventoryBeverage	None	String: BevName, int: LeftStock, MinStock, Float: Price

Function description: InventoryBeverage is function to display beverage name, stock left, minimum stock, and price of each beverages. It has sub functions that delete beverage and add beverage.

Function state-diagram:



Function mock up:

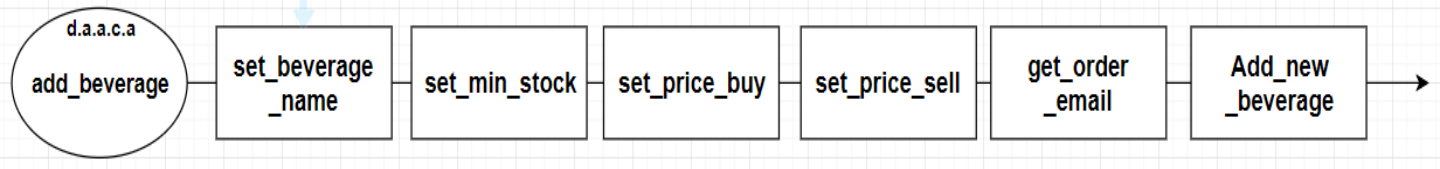


FWBS: 3.0

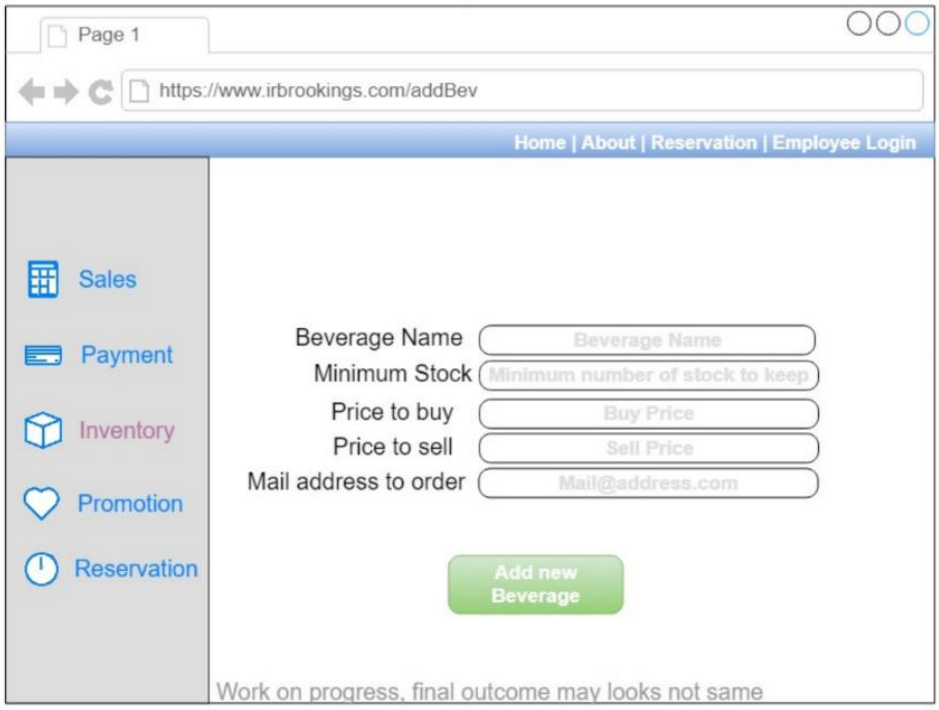
FWBS	Function Name	Input	Output
3.0	addBev	String: BevName, BevOrderMailAdd int: BevMinStock, float: BuyPrice, SellPrice	None

Function description: addBev is function that add new beverage into beverage database by manager. It requires beverage name, mail address that can order online, minimum stock, buying price, and sell price.

Function state diagram:



Function mock up:

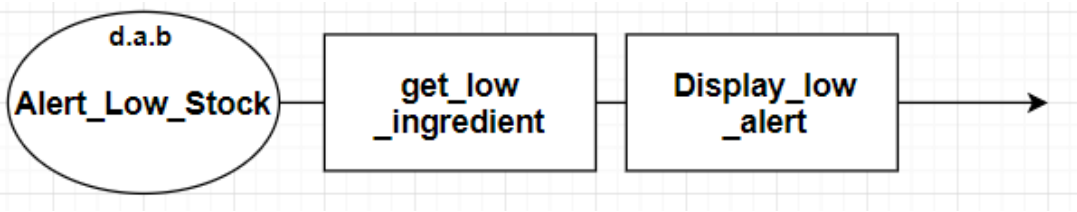


FWBS: 3.2

FWBS	Function Name	Input	Output
3.2	AlertLowStock	None	String: Name

Function description: It is automatic function that if any ingredients or beverage amount is going lower than minimum stock for each of them, then print out the alert message.

Function state diagram:

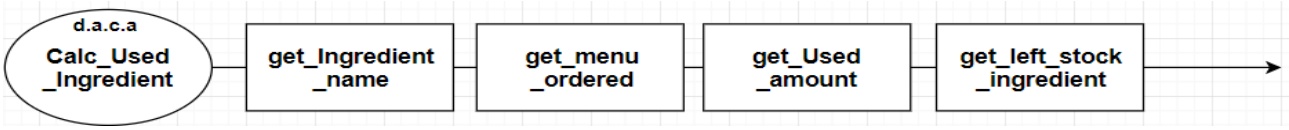


FWBS: 3.3

FWBS	Function Name	Input	Output
3.3	CalcUsed	String: IngName, MenuName, Int: UseAmount, leftStock	Int: leftStock

Function description: CalcUsed is function to calculate how much amount of stock left for after purchase a menu.

Function state-diagram:

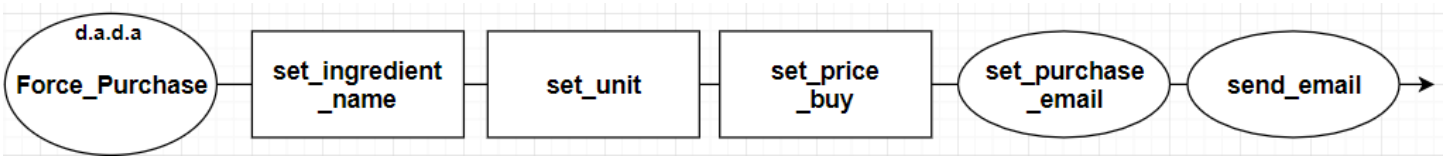


FWBS: 3.4

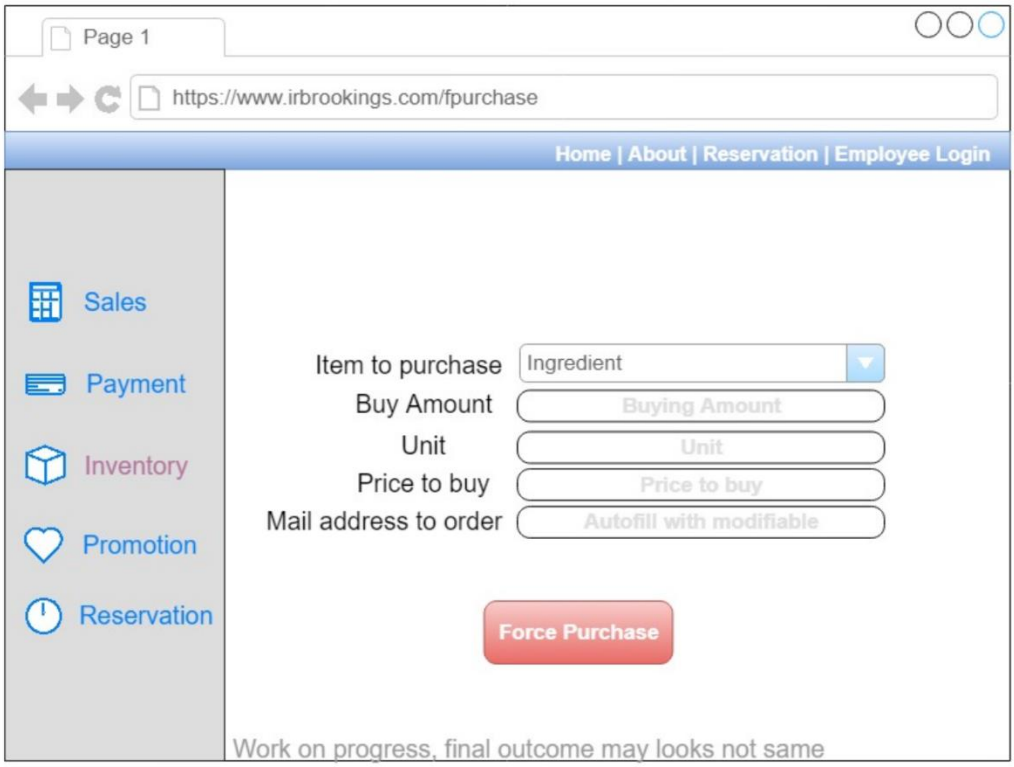
FWBS	Function Name	Input	Output
3.4	ForcePurchase	String: IngName, Unit, IngOrderMaillAdd Int: BuyAmount Float:PriceBuy	None

Function description: ForcePurchase is function to order amount of ingredient by manager. It requires ingredient's name, unit number, ordering mail, amount to buy, and price to buy.

Function state diagram:



Function mock up:



Mail template for auto weekly order

From IRB@IRbrookings.com

Subject Order Inventory (IRB)

To Retailer@Retail.com

CC

BCC

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 renegociate with price please reply this mail!

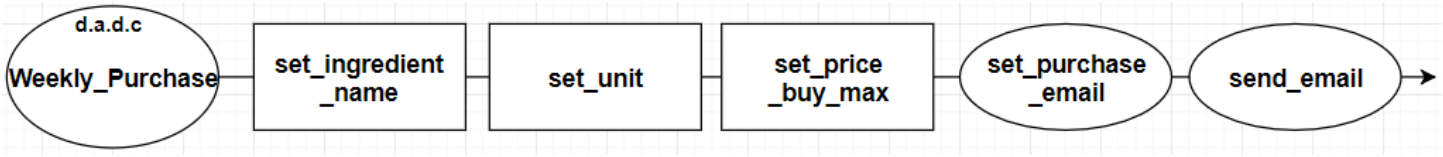
Thank you.

FWBS **3.4**

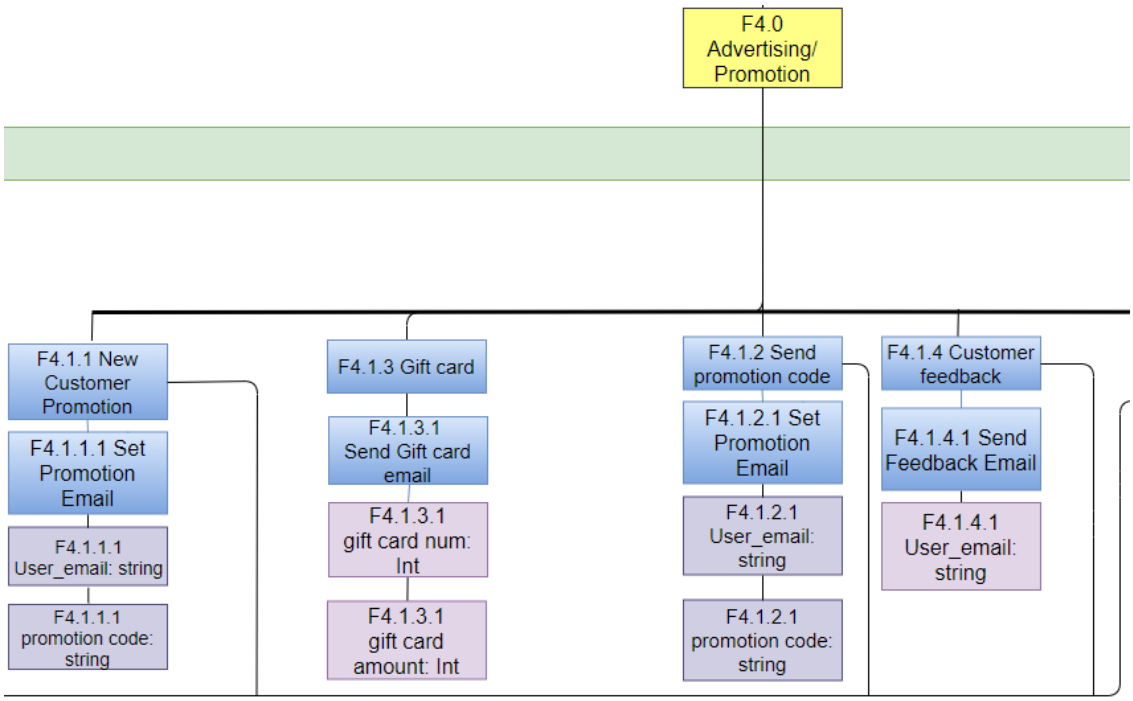
FWBS	Function Name	Input	Output
3.4	WeeklyPurchase	String: IngName, Unit, IngOrderMaillAdd Int: BuyAmount Float: IngMaxPrice	None

Function description: WeeklyPurchase is automatic function that run once a week, this function is ordering ingredients based on calculated amount of ingredient.

Function state-diagram:



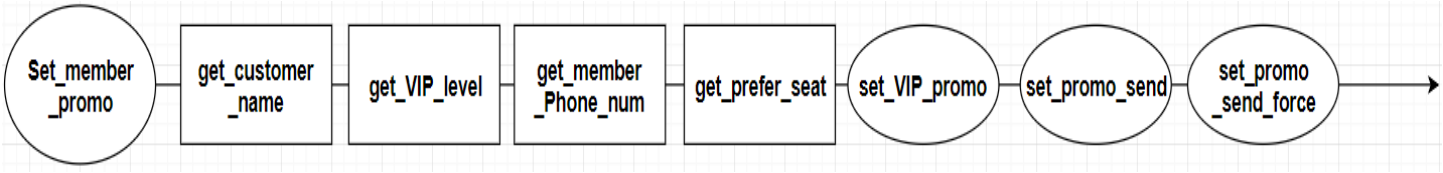
FWBS: 4.0



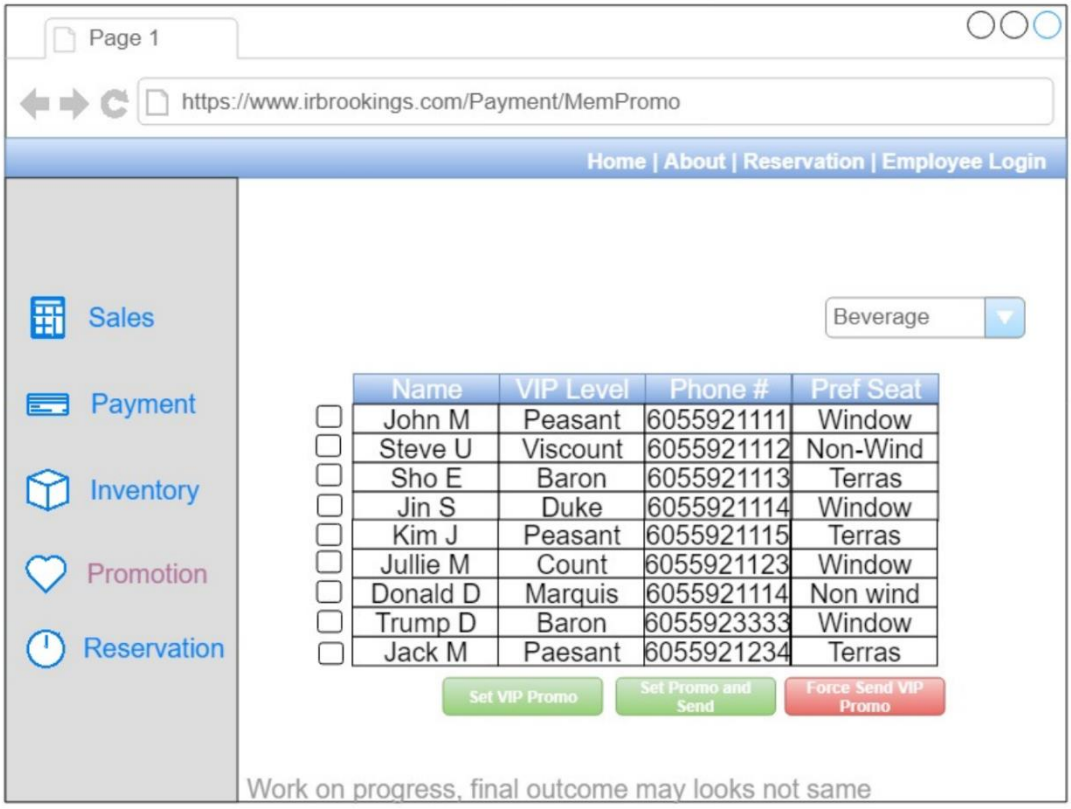
FWBS	Function Name	Input	Output
4.1	SetMemberPromo	None	None

Function description: function to printout the customers contact, VIP level, Phone number, and preferred seat. It also have sub functions to set VIP promo, Set Promo and send, and Force send VIP promo.

Function state diagram:



Function mock-up



Mail template for Coupon generation

From IRB@IRbrookings.com

Subject Coupon Just for you!

To JohnM@gmail.com

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 IRB@IRbrookings.com

Subject Coupon Just for you!

To JinS@gmail.com

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!

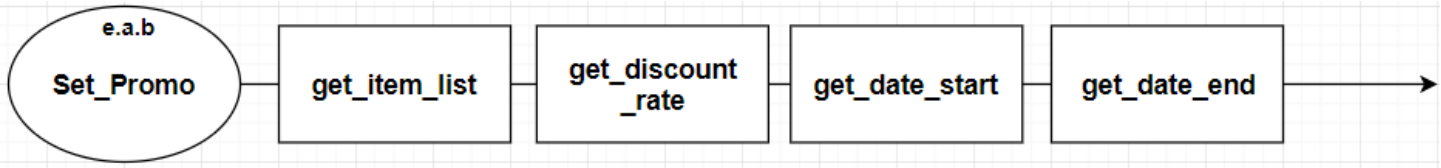
Code: JSENTR101
Valid threw: 2018/Sep/24~2018/dec/24

FWBS: 4.1.1

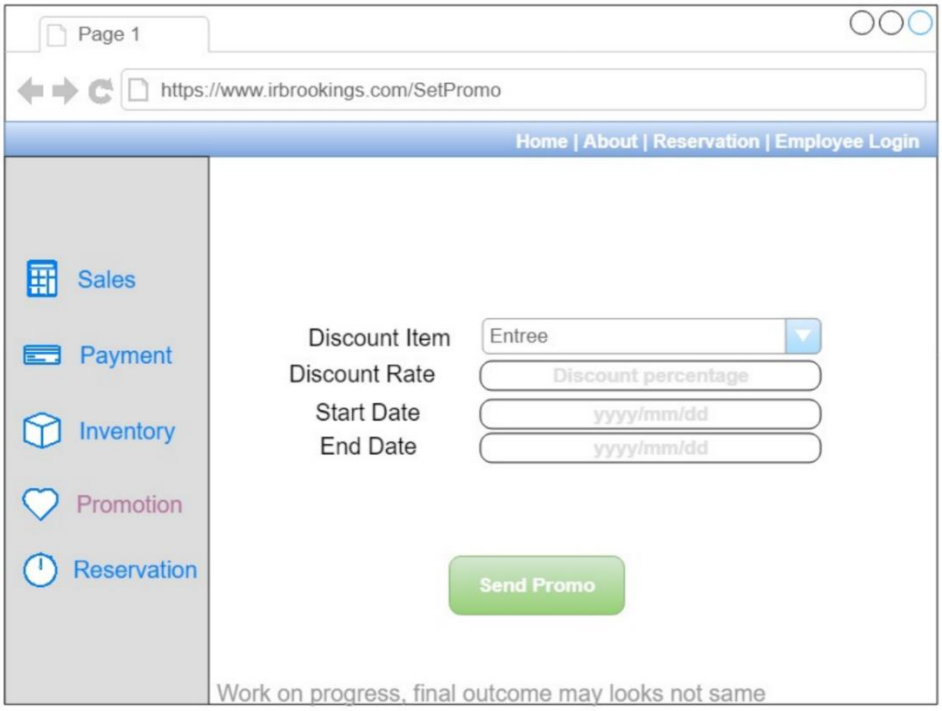
FWBS	Function Name	Input	Output
4.1.1	SetSendPromo	String: ItemList, int rate, date: Start, End	None

Function description: SetSendPromo is function to set the promo item with starting date and end date.

Function state-diagram:



Function mock-up:



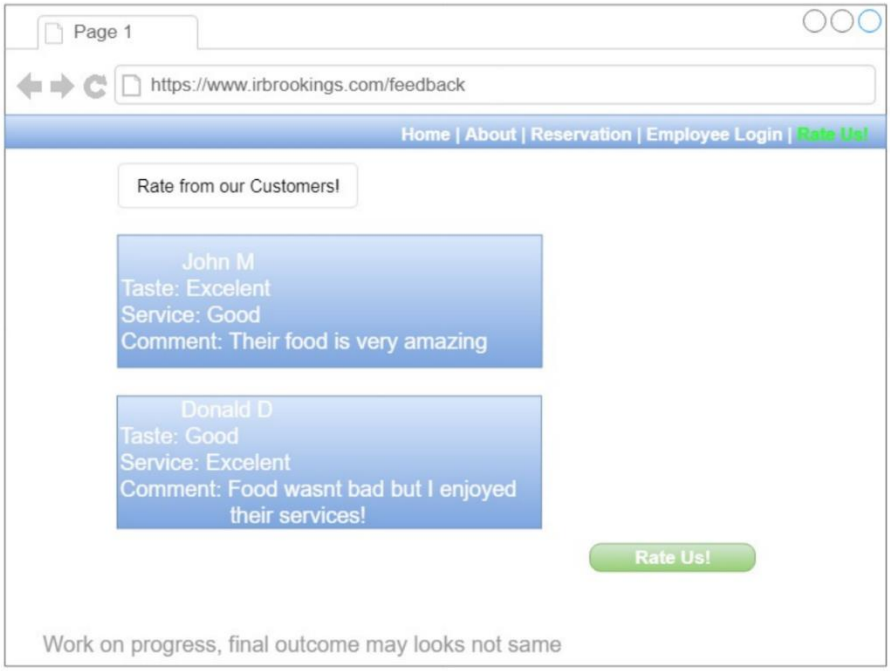
FWBS: 4.1.2

FWBS	Function Name	Input	Output
4.1.2	PromoCodeGen	String: CustName, VIPStatus, int: VIPDiscountRate, Date: DStart, DEnd	None

Function description: PromoCodeGen is function to generate the promo code based on VIP status from customer.

Function state diagram:

Function mock up:

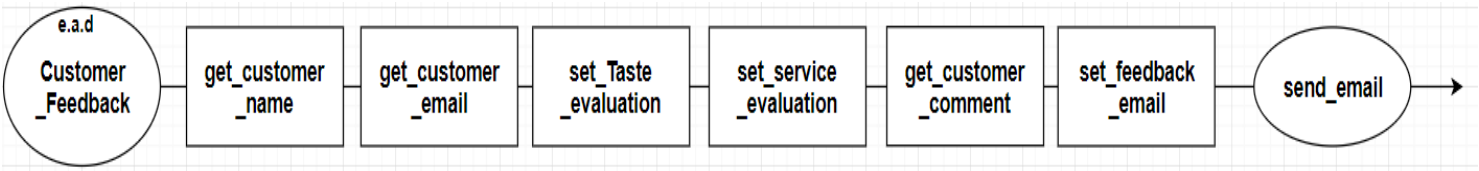


FWBS: **4.4**

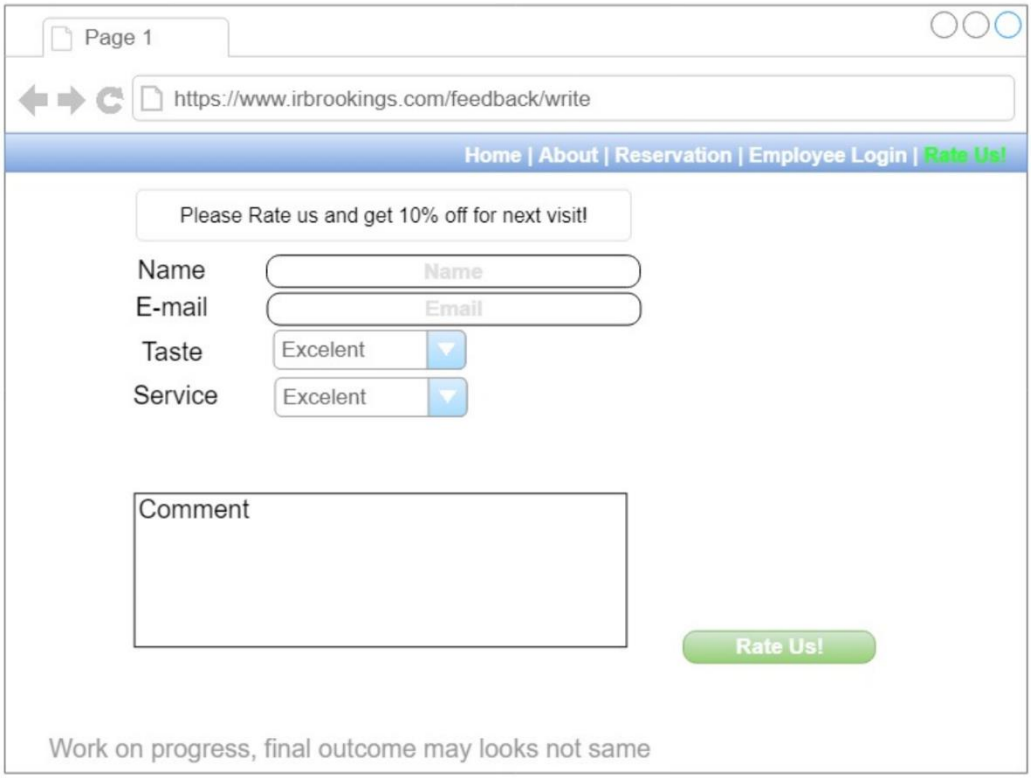
FWBS	Function Name	Input	Output
4.4	ViewCustFeed	String: CustName, Email Enum Taste, Service	String: GenCouponCode

Function description: ViewCustFeed is function to give customer feedback to user through mail. If feedback is done, it will gives promo code

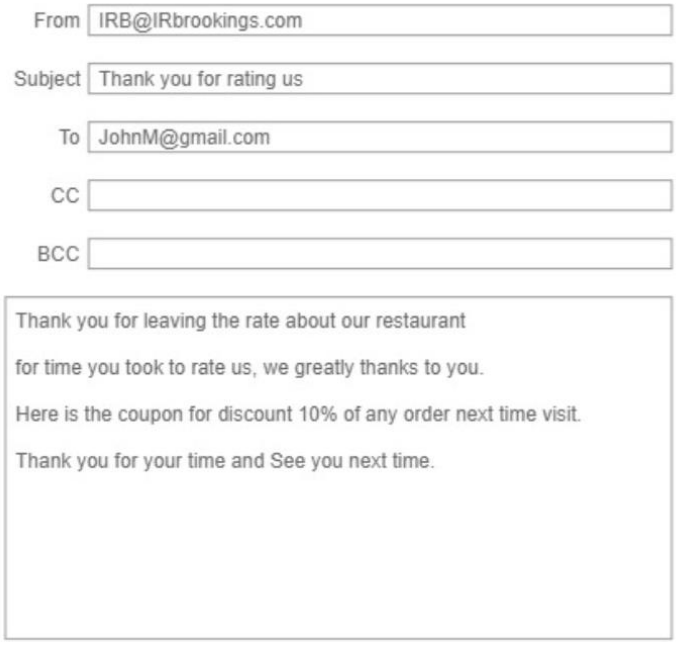
Function state diagram:



Function mock up



Feedback discount code mail

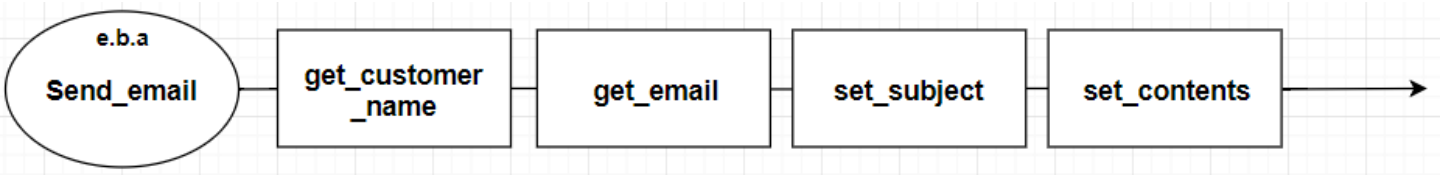


FWBS: 4.5

FWBS	Function Name	Input	Output
4.5	Send_email	String: CustName, Email, MailContent, MailSubject	None

Function description: Send_email is function to send email automatically to the customer and set the mail content and subject automatically based on purpose.

Function state diagram:

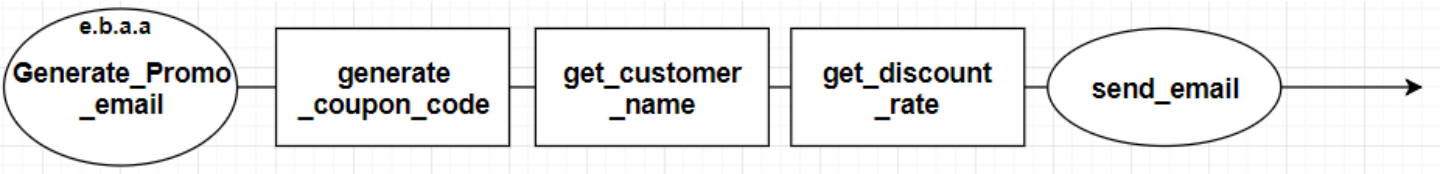


FWBS: 4.2

FWBS	Function Name	Input	Output
4.2	GenPromoMail	String: GenCoupCode, CustName Int: DiscountRate	String: MainContent, MailSubject

Function description: sub function of send email, set the mail content and subject for promo code to customer.

Function state diagram:

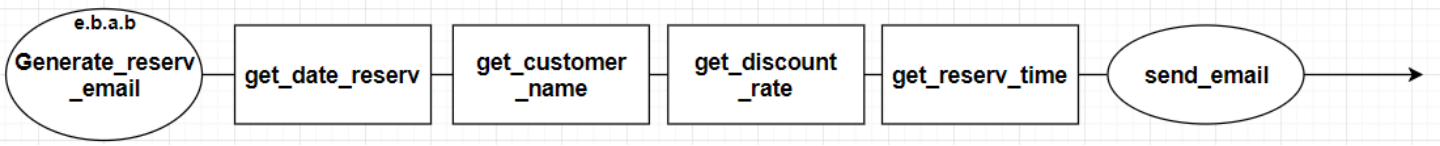


FWBS: 5.3.1

FWBS	Function Name	Input	Output
5.3.1	GenResvMail	String: Custname, Date: ResvDate, int: ResvTime	String: MainContent, MailSubject

Function description: sub function of send email, set the mail content and subject for reservation by customer.

Function diagram:



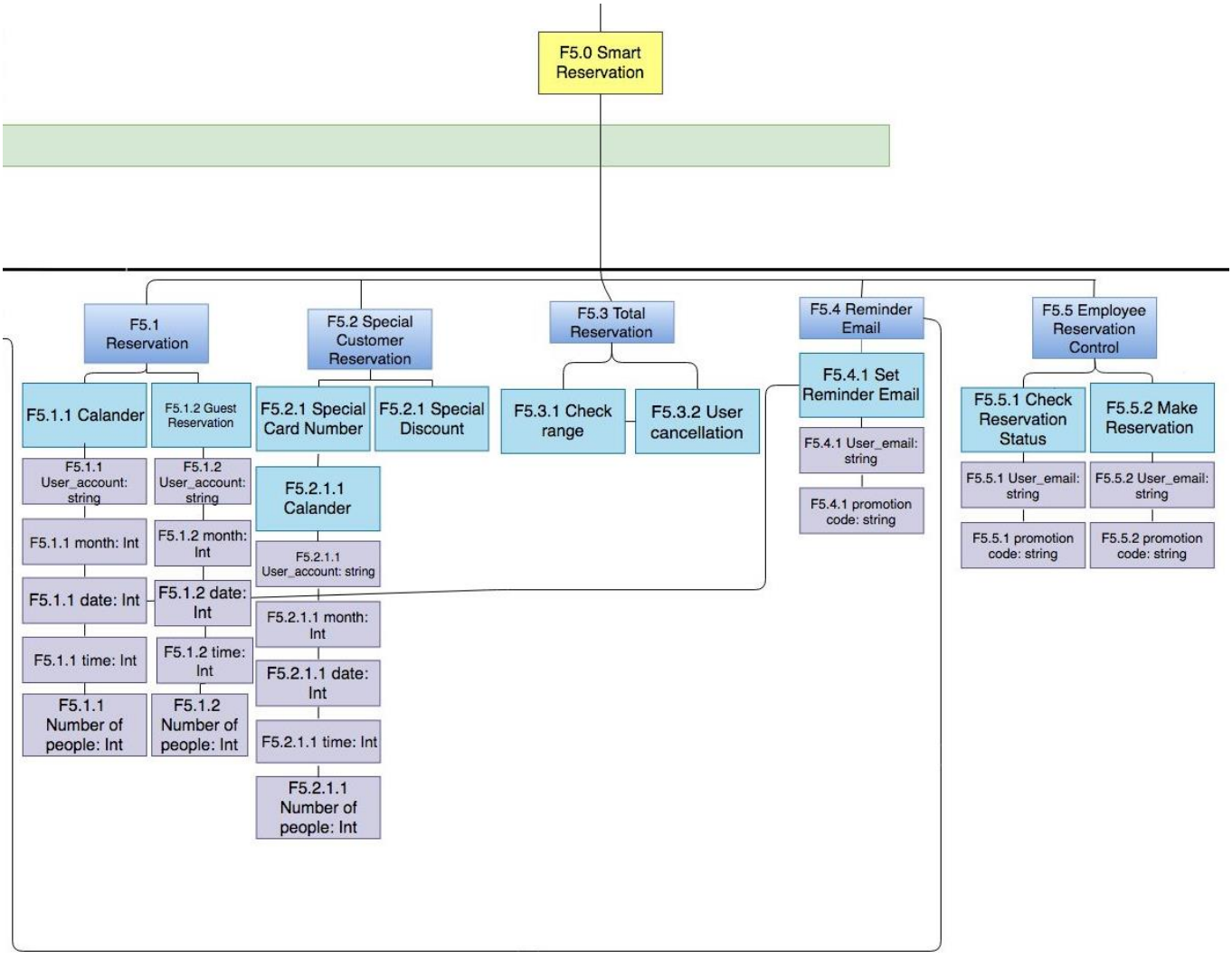
FWBS: 5.3.2

FWBS	Function Name	Input	Output
5.3.2	GenCancMail	String:Custname, Date: ResvDate, Int: ResvTime	String: MainContent, MailSubject

Function description: sub function of send email, set the mail content and subject for cancellation of reservation by customer or late.

Function diagram:

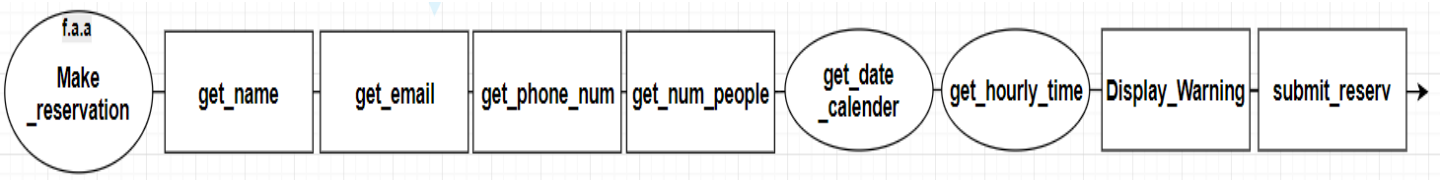
FWBS: 5.0



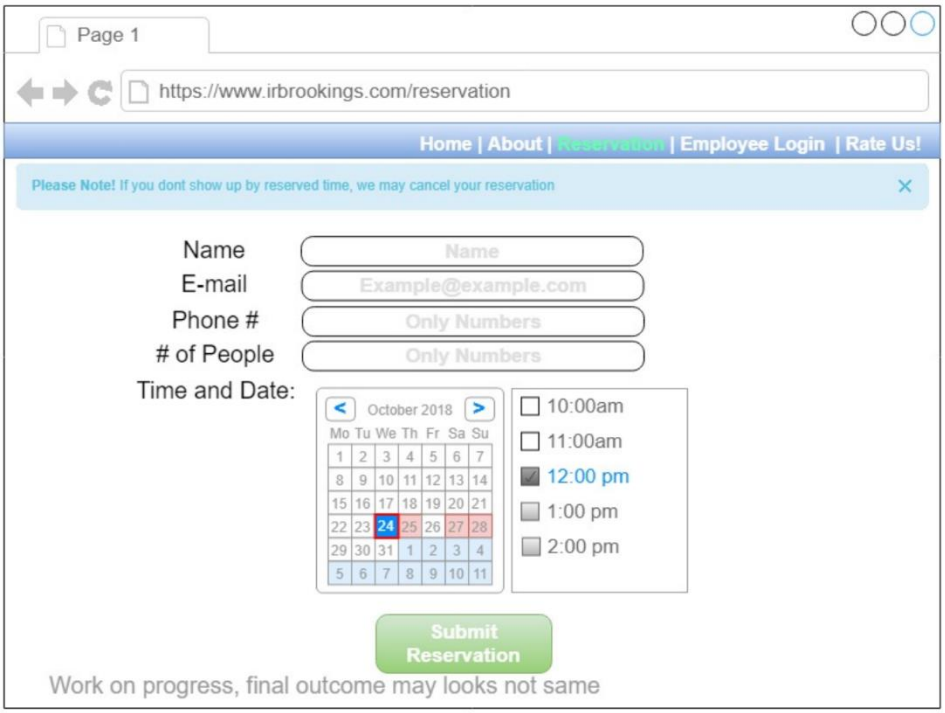
FWBS	Function Name	Input	Output
5.1	MakeReservation	String: Name, Email Int: PhoneNum, NumPeople, Date: getCalender Enum: HourlyTime	None

Function description: function to make reservation, based on date and time that user input call the check reservation function to check it is available.

Function state diagram:



Function mock up

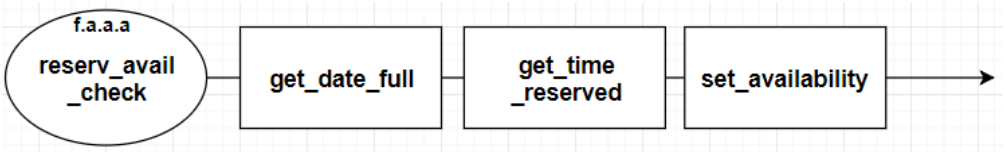


FWBS: 5.1.1

FWBS	Function Name	Input	Output
5.1.1	CheckResAvail	Date: FullReserved Time: RsrvdTime	Bool: Available

Function description: function to return the chosen date by user is available to reserve or not.

Function state diagram:

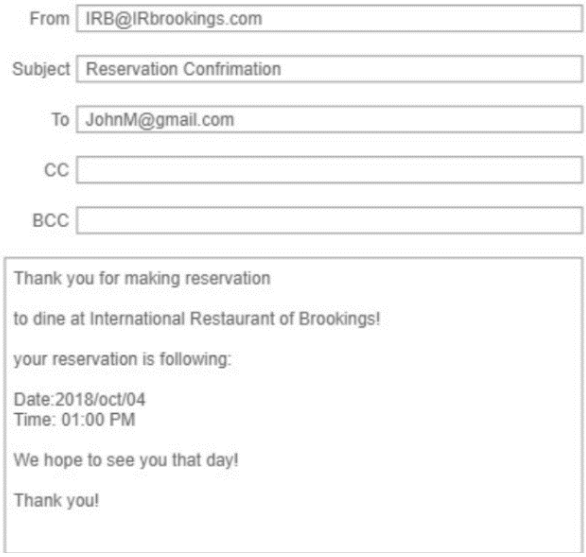


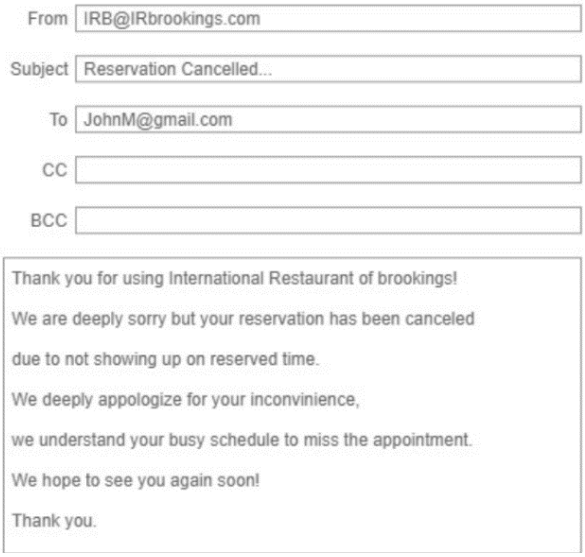
FWBS: 5.4

FWBS	Function Name	Input	Output
5.4	ResvControl	Int: PhoneNumber	String: ResCustName, VIPLevel, PrefSeat Int:Time, PhoneNum, NumResv, Bool:IsHere

Function description: it is function to control the reservation to check if customer is late or not.

Reservation confirmation mail and Cancelation mail





III.I.IV HARDWARE/SOFTWARE INTERFACES

Hardware:

Name	Condition
Display	1280x720 or higher
CPU	2.0 Ghz of Intel Core architecture 2nd 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

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.I.V COMMUNICATION INTERFACES/INTERNET CONNECTION

- ✧ Local Area Network with Ethernet Cable both grade CAT6
- ✧ No Wireless Connection
- ✧ Requires at least 30MBps speed with no limitation on data capacity

III.I.VI ESTIMATE OF CONFIGURATION

- ✧ Application Format: Web Application
- ✧ Server OS: Windows 10 PRO
- ✧ Application Size: 6MB
- ✧ Database Size: 20MB~1TB

III.I.VII INSTALLATION

- ✧ Connect Web start page with Windows IIS manager
- ✧ Set port forwarding at router to server Local IP
- ✧ Connect DNS URL address to router external IP
- ✧ Time estimate of installation less then 1 day in maximum

III.II NON-FUNCTIONAL REQUIREMENTS

III.II.I PERFORMANCE REQUIREMENTS

FWBS number	Function Name	Response Time	People in Charge
F0.0	U.I	0.03sec	Junmo Kim
F6.0	Login	0.1sec	Sanil Khamkar
F1.0	Sales	0.3sec	Sangwon Shin
F2.2.1	Calculate return	0.07sec	Sangwon Shin
F3.1	Create report	0.4sec	Junmo Kim
F3.4	Weekly purchase	0.05sec	Junmo Kim
F5.1	Reservation	0.8sec	Sanil Khamkar
F5.2	Special customer reservation	0.9sec	Sanil Khamkar
F5.3	Total Reservation management	0.9sec	Sangwon Shin

Performance refers to the amount of work achieved by the application in a minimum time. Performance requirement mainly focuses on the cost reduction aspects of an application, how to increase the response time of an operation, resource conservation and improving the efficiency of an application.
The application has been designed in a flexible environment keeping in mind about the future conditions in which the application needs to be modified or extended. The application requires only one user to operate at a time thus reducing the time taken to accomplish a service request or task. The response time of the application will be less than a second at 90% of the time when an operation is performed.

For the best optimal performance, the following hardware configuration is necessary.

Hardware	Name	Cost	Spec
Display	Dell E1916HV	$63	Res: 1366x768
Desktop	HP8300Elite	$194	CPU: i5 3.2Ghz RAM: 8GB Storage: 500GB
Keyboard	Amazon Basic Key	$7	Wired USB 2.0
Mouse	Amazon Basic Mouse	$7	Wired USB 2.0
Router	ASUS RT-ACRH13	$75	AC1300 Gigabit Dual Wi-fi USB 3.0 NAS
RAID HDD SET	Terramaster D2-310	$170	2 Bay 3.5inch HDD USB 3.0 upto 24 TB
Total Cost	$516		

III.II.II OPERATIONAL AND ENVIRONMENTAL REQUIREMENTS

Operational Requirement refers to how the application runs and communicates with the user. These requirements are based on the user needs. The main requirements are listed below.

1. Security

- The application is provided with login system so only the user or people who have access can operate the application. (F6.0)
- The application is restricted to viewing and altering data only by authorized access. (F6.1.2)
- The restaurant manager has full access such as controlling, viewing and altering data but limited to staff within minimum functionalities. (F6.1.2)

2. Recoverability

- The restoration of data in the event of application failure.
- System data back-up is done on a weekly basis.
- In event of failure, the application will restore itself based on the previous data file.
- If the data gets corrupted or destroyer due to external factors, a second back-up file is made available on the cloud server.

III.II.III RELIABILITY AND AVAILABILITY

Reliability:

Reliability of a software refers to the probability of failure free software operation for a specified period of time in a specified environment.

- Fault in an application leads to unexpected results or output which can be considered as a major or minor flaw in the application. Yearly maintenance of the application reduces the risk of fault or errors in the system.
- Failure of an application causes the system to completely shut down or affects the system to not work properly. Restarting the system may fix the problem or incase of system failure, the application can be restored from the previous back-up files.

Availability:

Availability of a software refers to the probability that a system is operational and delivers the requested services in a particular time.

- Availability can be referred in terms of percentage(%) such the application is running for 99% of the time.

III.II.IV BACK UP AND SECURITY

Back up:

The term backup refers to copying of application data into the existing system or storing it in a cloud server and restoring the original data in case of system failure or data loss.

- The application has a backup system which is usually done at the end of the day at midnight at 00:00 hours.
- RAID – I technology is being used for data backup.
- In event of system failure, the application is restored to its previous state hence preventing data loss.

Security:

- The application is provided with login system so only the user or people who have access can operate the application.
- The application is restricted to viewing and altering data only by authorized access.
- The restaurant manager has full access such as controlling, viewing and altering data but limited to staff within minimum functionalities.

III.II.V MAINTAINABILITY

Maintainability refers to modifying, updating or providing service to the existing application for performance improvements or correction of faults and bugs.

- The application is provided with yearly maintenance for fault reduction.
- Software updates are made available and prior notice is provided to the user.
- Immediate service is provided within a two day period if the system is unable to start or needs fixing of bugs.

III.II.VI TRANSFERABILITY/PORTABILITY/USABILITY

Transferability/Portability:

Portability refers to usability of the same software in different environments.

- The software application can be installed on different platforms to achieve the same tasks.
- The application data is stored on the database which easily allows to retrieve data on different platforms.

Usability:

Usability refers to the measure of how easy it is to use a product to achieve effectiveness, efficiency and satisfaction in a quantified context of use.

- The User Interface is built to be simple as possible that is user friendly in an easy to learn environment.

III.II.VII DOCUMENTATION AND TRAINING

Documentation for a software application refers with written documents and materials related to the software applications development and use.

- Application document describes how various tasks can be performed on the software.

- System document describes the overall architecture of the system which includes designs, activity graphs, flowcharts and source code.
- User documentation is prepared for the end-users like administrators and staff. Maintenance manual for administrator and simple page user manual for the staff.

Training:

Training refers to the time taken to teach the administrator and staff to learn to use the application.

- Training for administrator to learn the software can take around 5-7days.
- Training for staff to the learn the software can take up to 5-7days.

III.II.VIII EXCEPTION HANDLING

Exception handling deals with real-time systems or embedded systems. It is usually used to detect and recover from exceptional conditions.

- Programmed exception handling modules are built in the system, to prevent further damage done to the application. The system will recover to the previous point if the condition fails to recover.
- Default exception handling is also used in the system to prevent the system from containing the design effects

III.II.IX TESTING REQUIREMENTS

Work will be done in System Testing Plan

IV. LOG OF MEETING

Developer Team meeting every Monday and Friday 2PM
Log from Sep/10th to Sep/17th

Sep/10th Monday Meeting
Check progress of previously assigned works

Sep/14th Friday
Meeting with Client, Proposal Meeting
Obtained customers feedback about proposal
Organized feedback with team
Declared what to modify

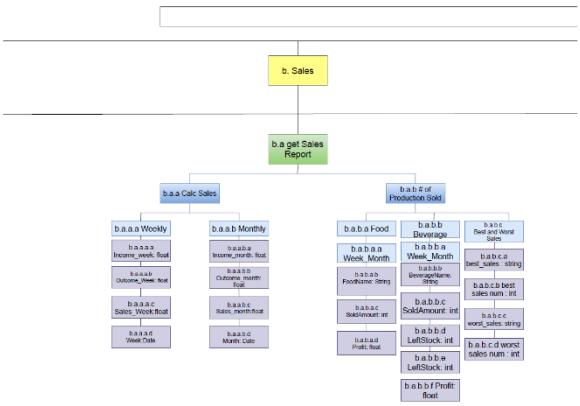
Assigned final modification jobs to Junmo
Requirement Documentation ideas

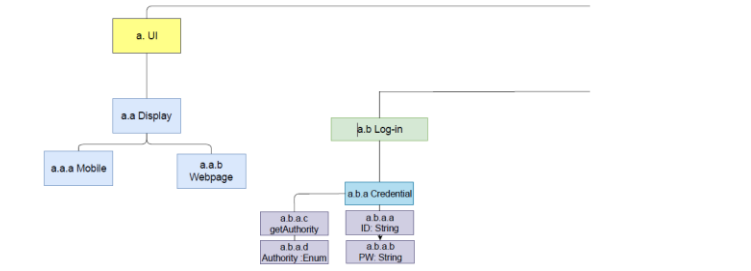
Sep/17th Monday
Assigned RD jobs to team
Introduction – Sanil
System Overview Diagram – Sangwon
General Description - Sangwon
Functional Requirement
Function flow state diagram - Junmo
Mock Up – Sangwon
Requirements and I/O table - Sangwon
Non-Functional Requirement – Sanil

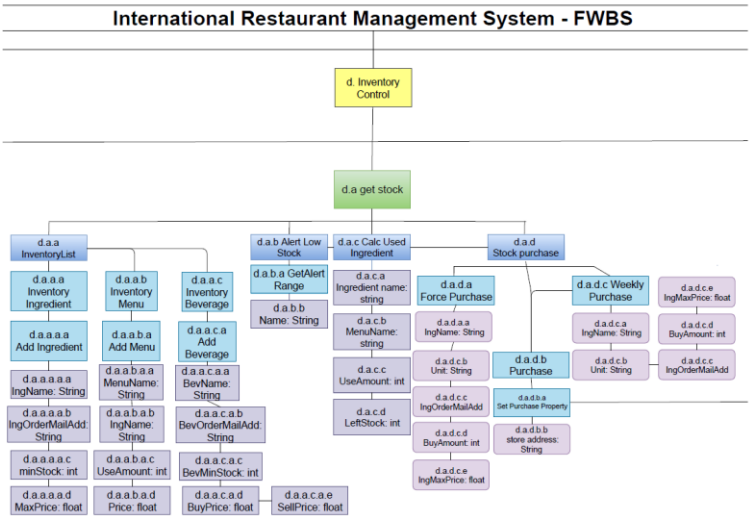
V. CHANGE CONTROL

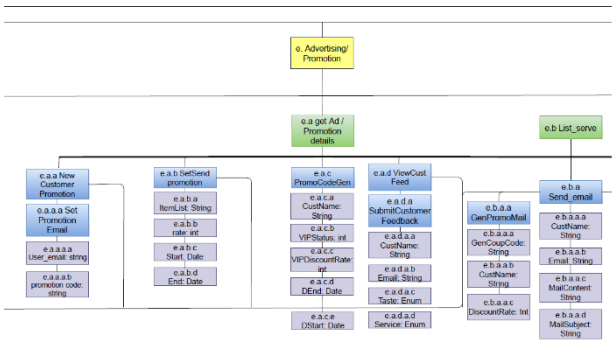
Change Control section introduce any systematical feature change happens from proposal. Since by many reason system may change. Following change is required to be assigned with client for new feature to be added as contract.

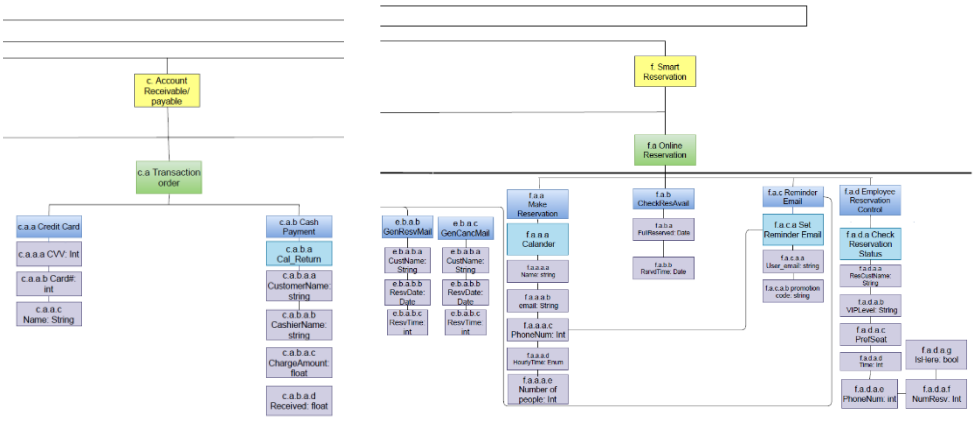
Modified FWBS

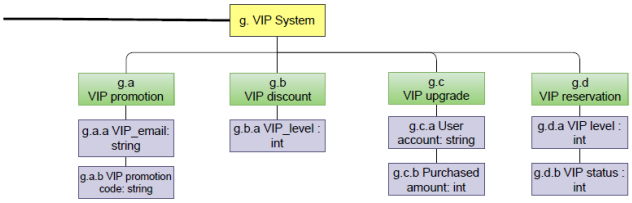












If you accept above changes to FWBS, Please Sign: _____ Date:_____