UNIVERSITY OF GREENWICH

DISTRIBUTED INFORMATION SYSTEM

NATIONAL RACQUET AND HEALTH CLUB

NGUYEN TAN TAI (GT60185)

4/16/2013

This report contains the requirements, analysis, design, implementation and evaluation of Distributed Information System course work (COM1303 – 2013).

Table of Contents

1. Introduction

1.1. The Scenario

National racquet and health club is a company which operates 20 centres within the United Kingdom. In a near plan, they will build a distributed system that can not only manage all business operations locally but also provide a national report system.

To be more specific, they want each centre will run a local database that store all relevant information to that centre. In addition, all centres must provide same management functionalities such as membership management, booking management, and sale management. These functions will be executed by the local staff through a numbers of client terminals in each centre. In addition to staff, there are local managers. The managers of a local centre can not only do all functions of a staff but also produce sale report and individual staff statistic.

In order to integrate all centres together, National racquet and health club system designate a centre to be a national management centre. This centre has all functions of a normal centre plus some specific operations. The specific functions includes create various reports that summary information for individual centre and collective national summary information. The reports which are need to be produced are weekly, monthly memberships and weekly, monthly sales from all centres.

1.2. Assumptions

In order to do all functions that are mentioned in the coursework, some assumptions must be pointed out. The following table will summary all assumptions.

Table 1: Functions' implementation summary

Function	Include	Exclude
Security	Authentication with username and password Authorization for each type of users	No encryption Web service unsecure temporally
Membership Management	Add, update, delete only	
Activity Management	Add, update, delete only	
Sale Management	Add, update, delete only	
Local report	Report on sale, activity	

	Statistic on individual sale	
National report	Only report function was implemented	
Employee management	No implementation(Employee information was added manually)	
Centre management	No implementation	

We agree that users for this system are divided into three types namely staff, local manager, and national manager. Each user type have different role in the system. The specific role of each user will be described in the following section – requirement.

This project also assumes that member can register with one and only on centre only. In other words, a user cannot register at another centre if he has already registered before.

1.3. Requirement and Use-case diagram

Sale staffs have the capability to manage membership activity such as update, insert, and delete activity order. They also have responsibility to sell food and beverage.

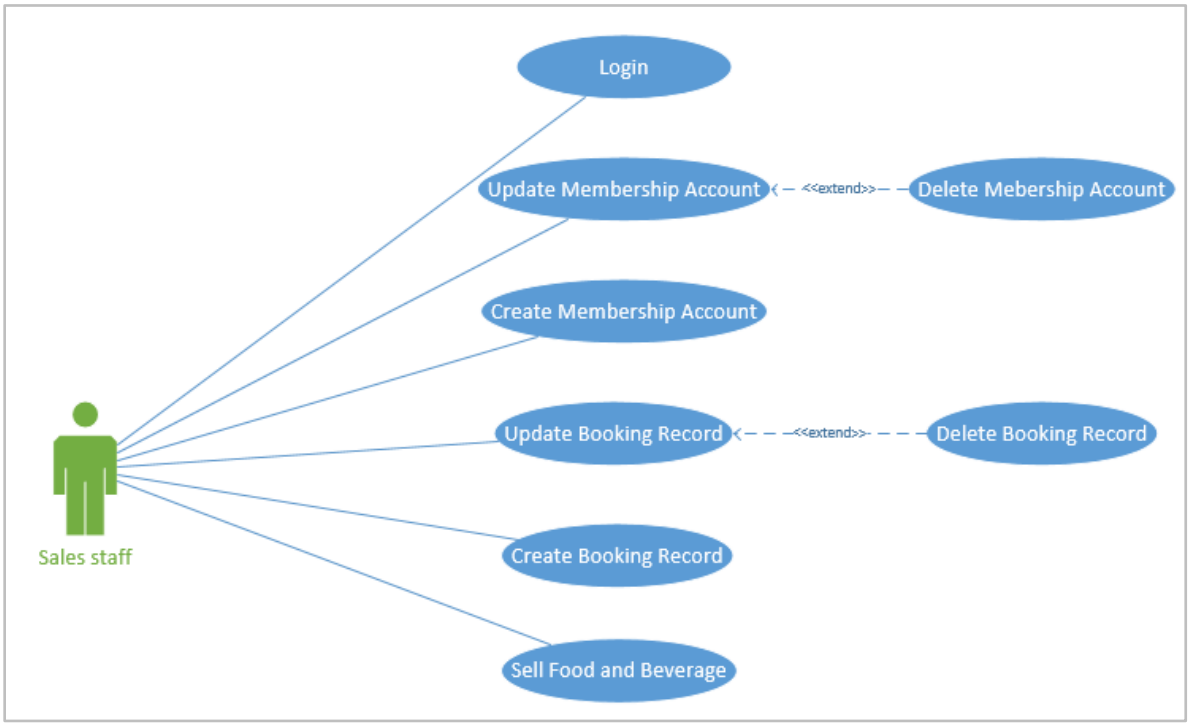


Figure 1: Use-case diagram for Staff

Local managers can not only execute all function of a staff but also produce report and statistic of sale. The reports are weekly or monthly sale, and weekly or monthly membership. The statistics are about individual sales of each staff.

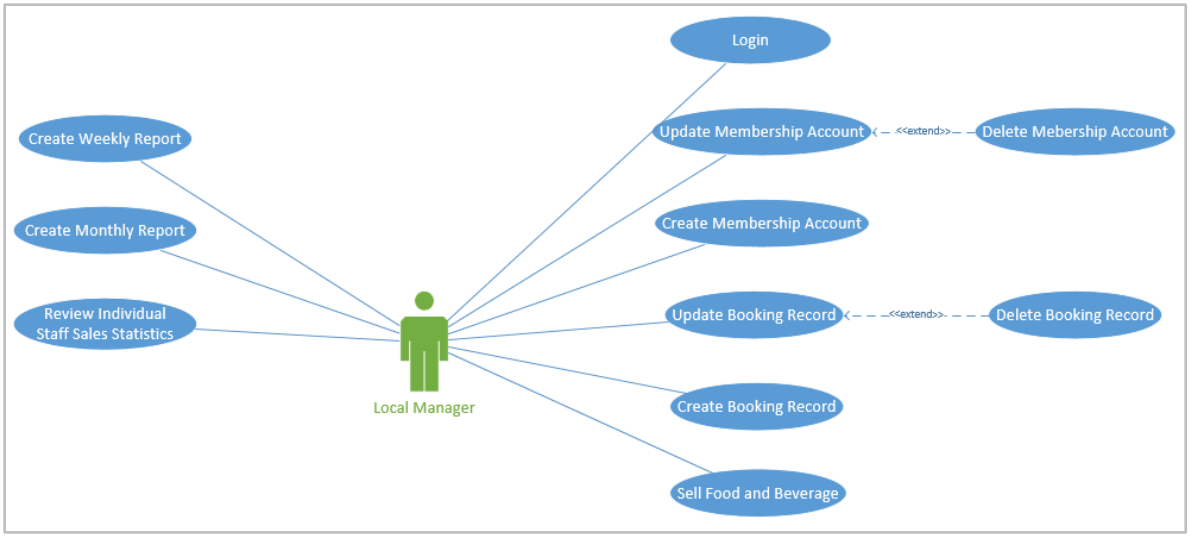


Figure 2: Use-case diagram for Local Manager

National managers, who work at one of the centers of the system, are authorized to execute all functions of any local center such as local reports, statistic. In addition to those functions, nation managers can create national reports and statistics. The reports are national sale/membership report which is summarized by weekly or monthly of some specific centres or all centres. The statistics are national weekly or monthly sale statistic of some centres or all centres.

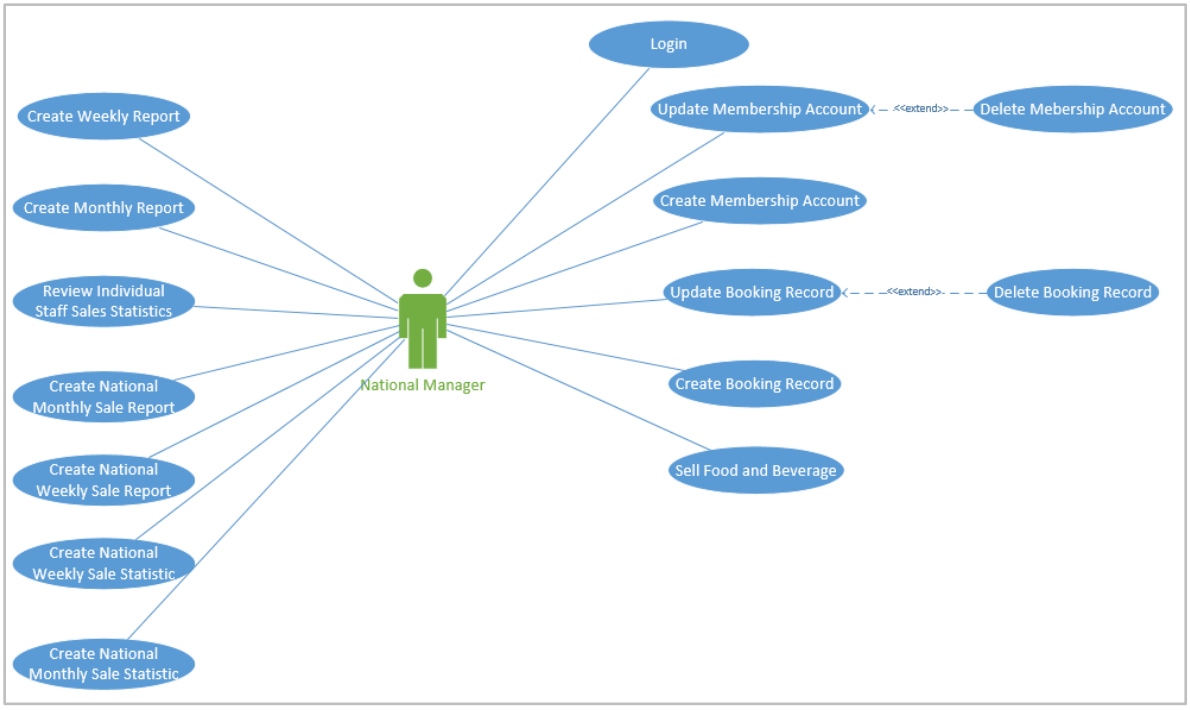


Figure 3: Use-case diagram for National Manager

1.3.1. Use case description

a. Login

Table 2: Use-case description for login

Primary scenario	
ID	**UC-001**
Title	**Login**
Description	The purpose of this use-case is to login to the system with appropriate username and password.
Preconditions	• User must enter login page. • There is no active action available at that time.
Flow of events	1. Browser show login page with two fields, one for username and one for password 2. User enters username and password. Press submit button. 3. System gets login info, transfers it to central service for authentication. Then system returns the login result. (AF1)
Alternative flow:	AF1: If user code is belong to that center. System won't transfer login info to central service. It retrieves information from local center instead.
Post condition:	• User must login successfully into the system

b. Create membership account

Table 3: Use-case description for creating membership

Primary scenario	
ID	**UC-002**
Title	**Create membership account**
Description	The purpose of this use-case is to create a membership account.
Preconditions	• User must login successfully to the system • User must have permission to do this function
Flow of events	1. User clicks create new membership from menu. 2. System shows membership creation form. 3. User enter appropriate information for membership(AF1) 4. User click submit button 5. System check information again and return answer(AF2) 6. System navigates to membership list.
Alternative flow:	AF1: If user enters wrong data, system show hint and feed back AF2: If error occurs, system will bring user back to event 3 and show error messages
Post condition:	• Membership must be added to database successfully into the system

c. Update membership account

Table 4: Use-case description for updating membership

Primary scenario	
ID	**UC-003**
Title	**Update membership account**
Description	The purpose of this use-case is to update membership information.
Preconditions	• User must login successfully to the system • User must have permission to do this function
Flow of events	1. User navigates to membership management page from menu. 2. System show a list of member ships with two buttons, one for edit and one for delete. 3. User press the edit button that belongs to the membership record to edit.(AF1) 4. System navigate to edit link, loads data and show edit page. 5. User change one of the fields(AF2) 6. User clicks save button 7. System saves information and navigates to membership list.

Alternative flow:	AF1: User can click delete button to delete a member account AF2: User can cancel and press back button to navigate to membership list page
Post condition:	• Membership must be updated/deleted from database.

d. View local sale/activity report

Table 5: Use-case description for view local sale/activity report

Primary scenario	
ID	**UC-004**
Title	**View local sale or/and activity report**
Description	The purpose of this use-case is to view local sale or activity report
Preconditions	• User must login successfully to the system • User must have permission to do this function
Flow of events	1. User navigates to local report section 2. System shows filter form for local report section 3. User chooses filter option from the form 4. User presses submit button 5. System summary data and show result table of report
Alternative flow:	None
Post condition:	• Local sale report must be view to user.

e. View local sale/activity statistic

Table 6: Use-case description for view local statistic

Primary scenario	
ID	**UC-005**
Title	**View local sale or/and activity statistic**
Description	The purpose of this use-case is to view local sale and activity statistic
Preconditions	• User must login successfully to the system • User must have permission to do this function
Flow of events	1. User navigates to local statistic section 2. System shows filter form for local statistic section 3. User chooses filter option from the form

	4. User presses submit button 5. System get data and show result table of statistic
Alternative flow:	None
Post condition:	• Local statistic report must be view to user.

f. View national sale/activity report

Table 7: Use-case description for view national report

Primary scenario	
ID	**UC-006**
Title	**View national sale or/and activity report**
Description	The purpose of this use-case is to view national report about sale and activity
Preconditions	• User must login successfully to the system • User must have permission to do this function
Flow of events	1. User navigates to national report section 2. System shows filter form for national report section 3. User chooses filter option from the form 4. User presses submit button 5. System open connection to central server and ask for data. Data was collected and return to system. System show result to user
Alternative flow:	None
Post condition:	• National sale/activity report must be view to user.

g. View national sale/activity statistic

Primary scenario	
ID	**UC-007**
Title	**View national sale or/and activity statistic**
Description	The purpose of this use-case is to view national statistic about sale, activity and membership
Preconditions	• User must login successfully to the system • User must have permission to do this function
Flow of events	1. User navigates to national report section

	2. System shows filter form for nation report section 3. User chooses filter option from the form 4. User presses submit button 5. System open connection to central server and ask for statistical data. Data was collected and return to system. System show result to user.
Alternative flow:	None
Post condition:	• National statistic must be view to user.

2. System Design

2.1. Entity relationship diagram

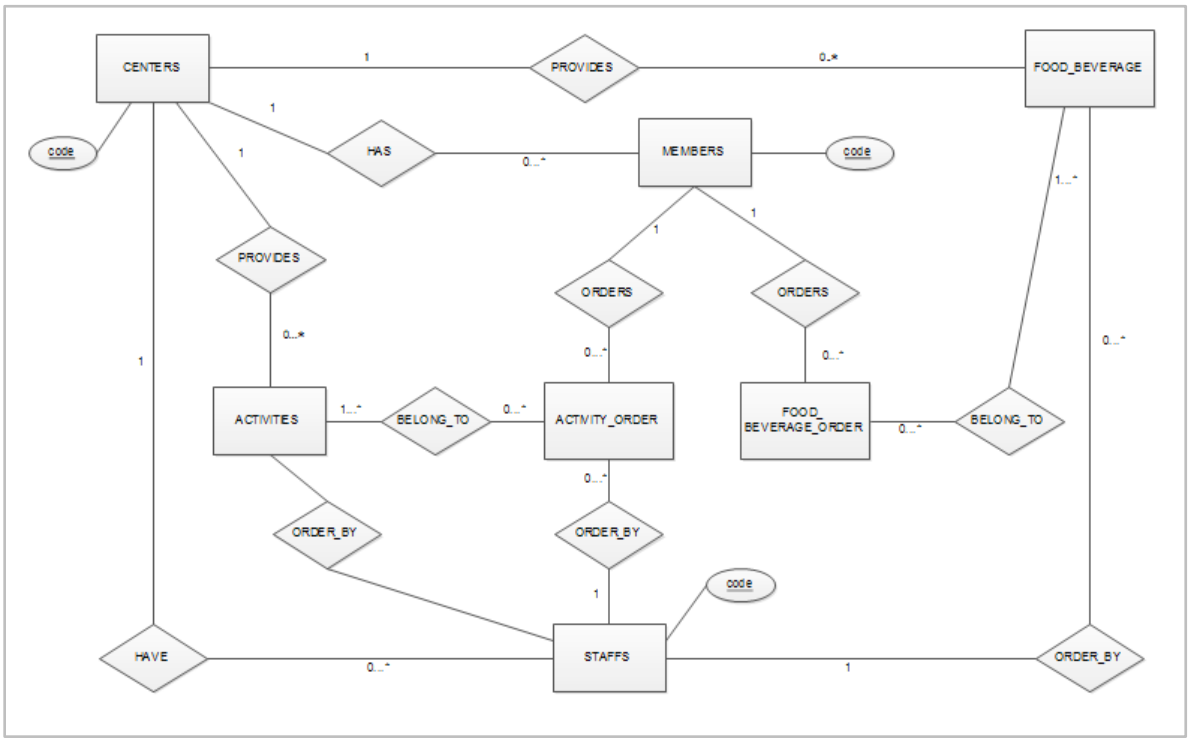


Figure 4: ERD diagram

2.2. Deployment diagram

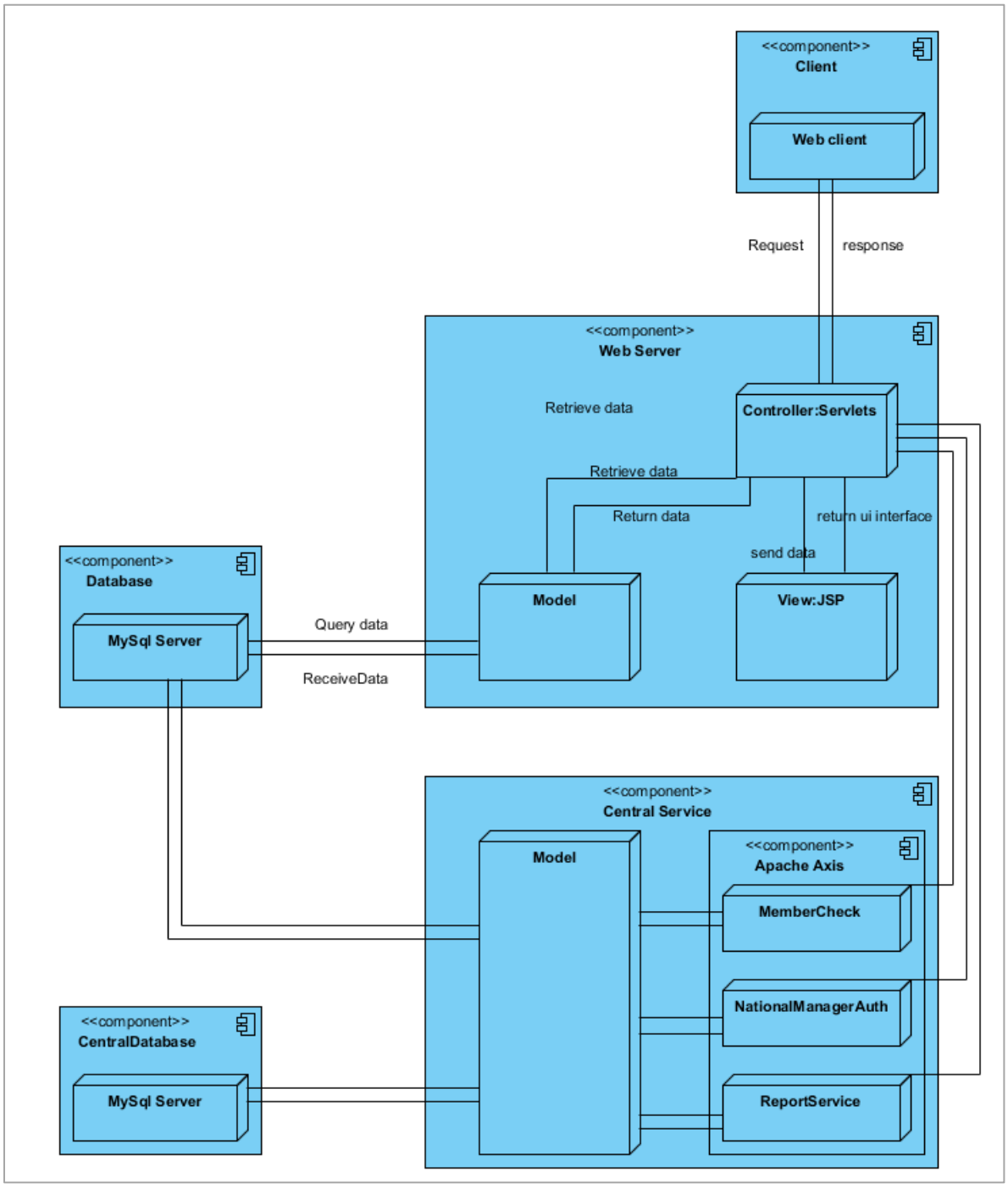


Figure 5: Deployment diagram

2.3. Class diagram

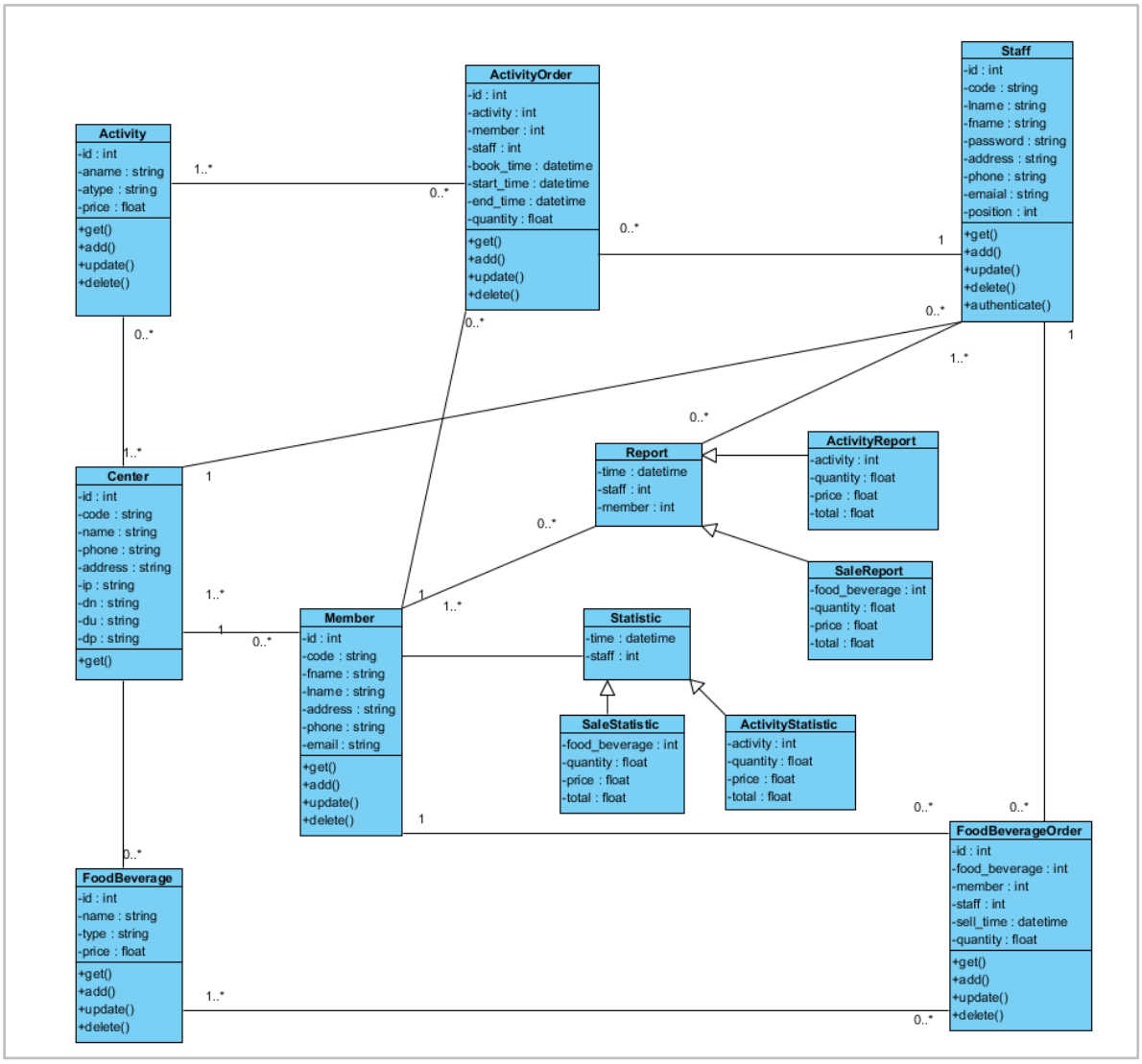


Figure 6: Class diagram

3. IMPLEMENTATION

3.1. Heterogeneity and Middleware

The proposed system is supposed to be a distributed 3-tier system. The system must comprise many parts include some kinds of servers and clients. Because it was a web-based applet, it allows user to access using different device and software. In this system, client can be any type of software such as Mozilla Firefox, Google Chrome, Apple Safari, Microsoft IE, etc… Client also can be run on different type of platforms such as pc, laptop, tablet or smart phones.

As the result, this application prototype is satisfied heterogeneity and middleware characteristics.

3.2. Openness and Interfaces

This software prototype uses J2EE JSP/Servlets technology plus a suit of web services that run on Apache Axis2. These web services provide an opportunity for future extenuation, upgrade. Moreover, web service also provides ability to implement and use by different platforms. With the public API from a web service, any platform with appropriate protocol can connect and use it. In other word, this system is open.

3.3. Security

The proposed system must ensure data integrity and security. To achieve this goal, some security method must be implemented. First, authentication is used to authenticate unrelated user. Second, authorization is used to keep user out of unauthorized action. Third, to secure data, some cryptography methods were used to encrypt data. Sensitive data such as password, credit card details will be encrypted in database to reduce information leaking. Because in distribute environment, we cannot trust anyone, certificates will be used to secure the transaction between nodes in the system.

3.4. Scalability

It is clear that, this prototype is scalability. On the one hand, every centre has its own database and can work independently. When a new centre join, it is easy to operate a same system on that centre without any modification of the current system. On the other hand, a small modification must be made on the centre server to make all things work properly. Administrator just needs to add the new record for new centre to connect to it. This record includes information to access new centre such as ip address, database name, database user and password. As a result, this prototype is a scalable system.

3.5. Concurrency

This prototype uses 3-tier model. At presentation tier, many users with web browsers can connect to the business tier, which are J2EE servers. Depending on the power of this server, it can serve hundreds or even thousands of users at the same time. At the third tier – database tier, MySQL was used. Enterprise version of this server can serve thousands of connection at the same time. All in all,

with the design and technology used, this prototype can serve more than hundreds of user at the same time.

3.6. Transparency

The prototype has implemented to achieve transparency characteristic of a distributed information system.

To achieve access transparency, two authentication methods are provided in login function. As illustrated from the prototype, local user and remote user use the same login page. However, when receive login information, server will determine whether user are local or remote and do different method to check for authentication. All processes are executed silently, behind the user view.

Location transparency is achieved by keep a list of centers' information on central server. Ever remote method was transfer to central server. This server will be responsible for contact other centers and return the result. In the entire progress, user does not know where other centers are.

Concurrency, performance, scalability and mobility transparency is achieved by using 3-tier, client-server model. This kind of model allows multiple user do their task independently. User without appropriate privileges will never know how many users are working on the system. User can change their client freely. All they need is a web browser with internet connection. When system need to be upgraded or extended, no big modification need to be made. Developer only needs to change or replace more powerful resource. Then the system can continue to work.

Failure transparency can be achieved by implement a good error handler. In addition, a good recovery function will be built to make the system run thoroughly. By doing this, user will never know what really occur inside the system, although they can work effective and safely.

3.7. Failure

This prototype is supposed to be partial failure. That is to say that failure of one centre does not affect other centres. This is very good strategy despite some disadvantages. The most serious disadvantage is national report and statistic function. To execute this function, a number of connections were created to connect to each centre. The problem is that it will take a long time to execute the function fully. During that time, many network errors may occur and prevent the success of the function. To overcome this problem, partial report or statistic is recommended. Instead of query all databases of every centre, system will query each centre, store result and keep track of every centred whether

success or not. After all centres in the list are queried, system will reconnect to the centres that failed before. This recovery method may prevent the missing data when central server has to deal with a numbers of database connections.

4. DESIGN

4.1. Login and Home page

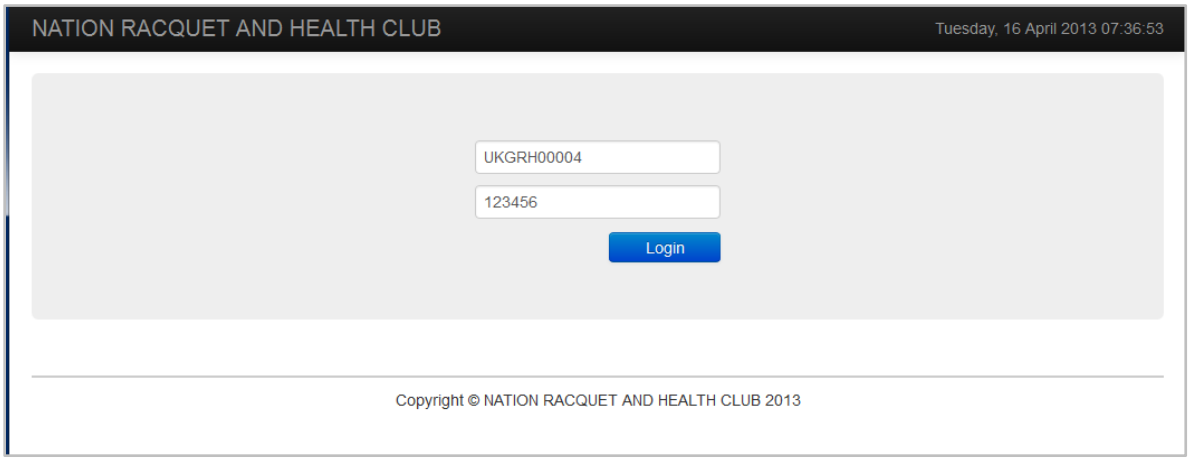


Figure 7: Login Interface

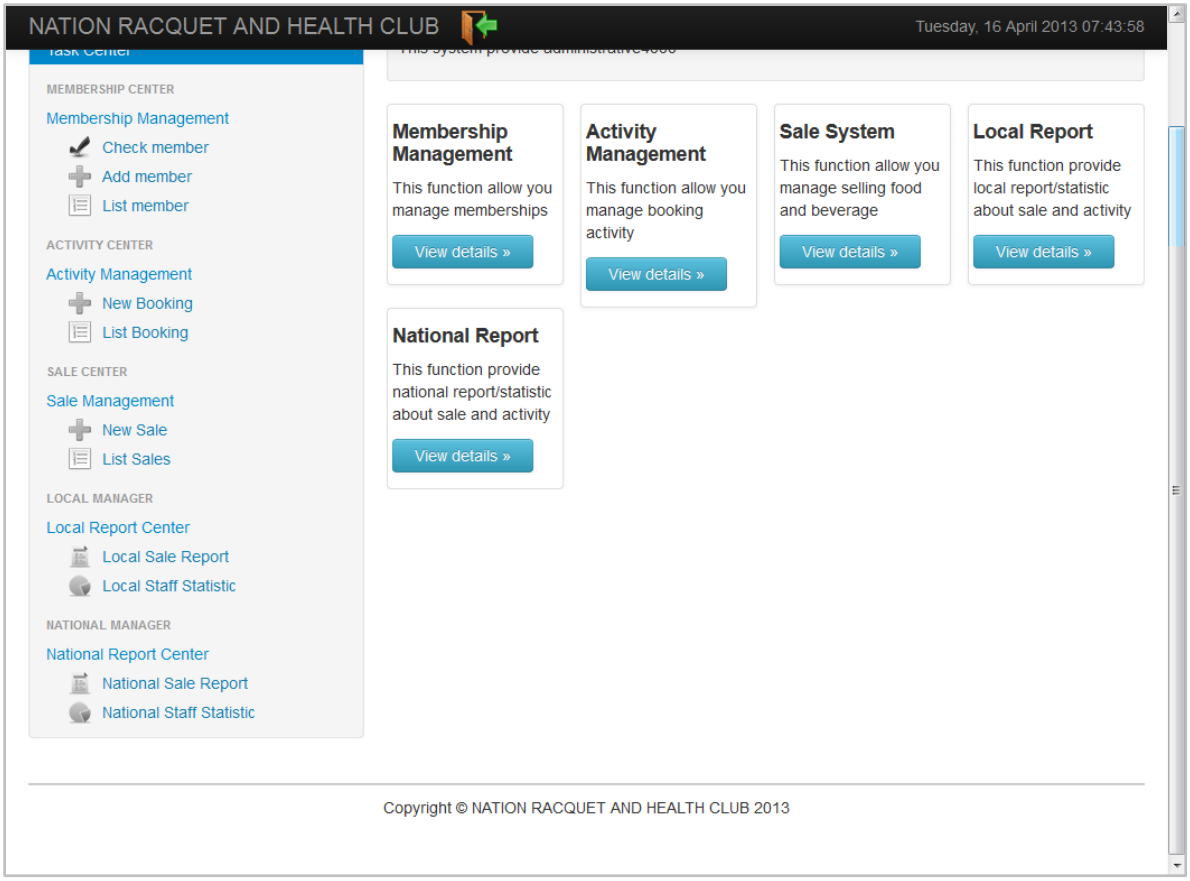


Figure 8: Home page

4.2. Membership Management

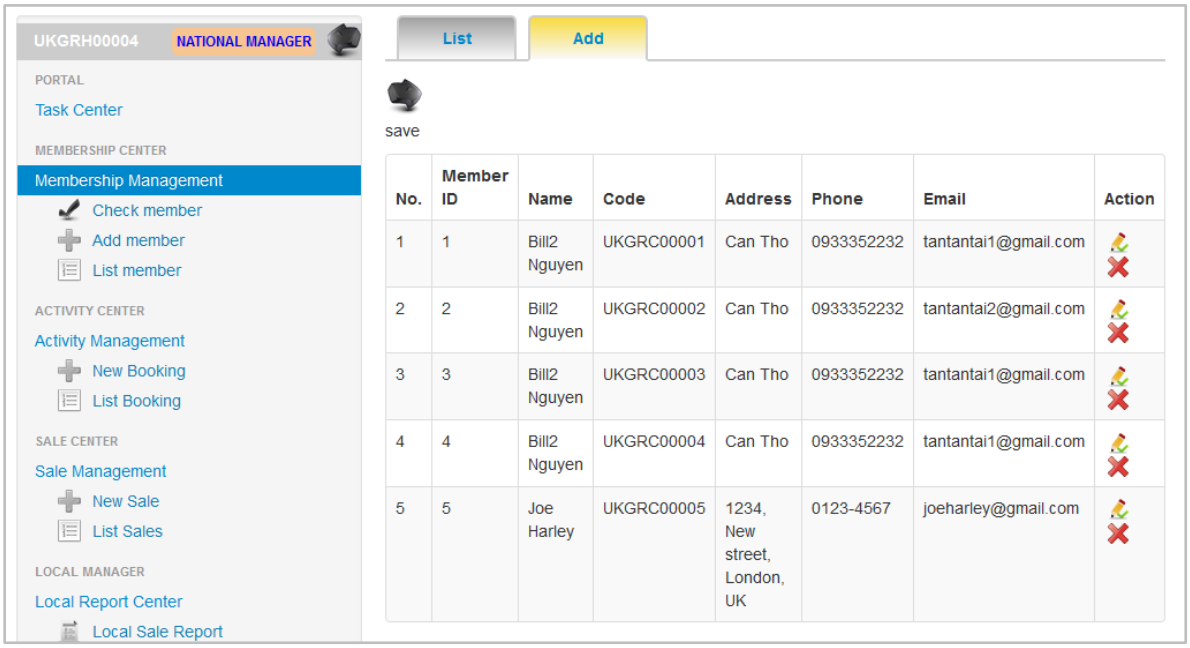


Figure 9: List Local Membership

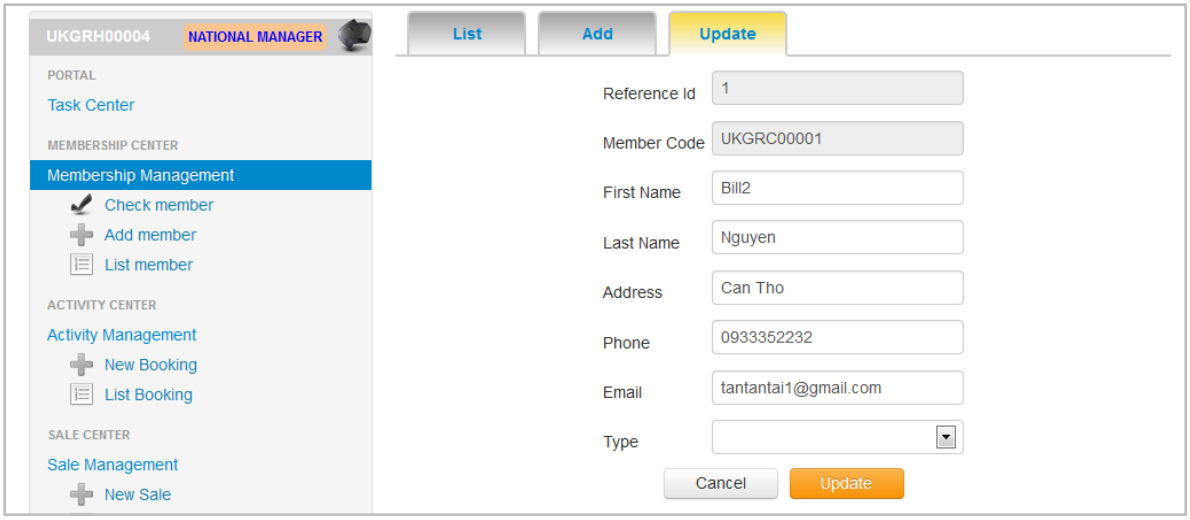


Figure 10: Update membership information

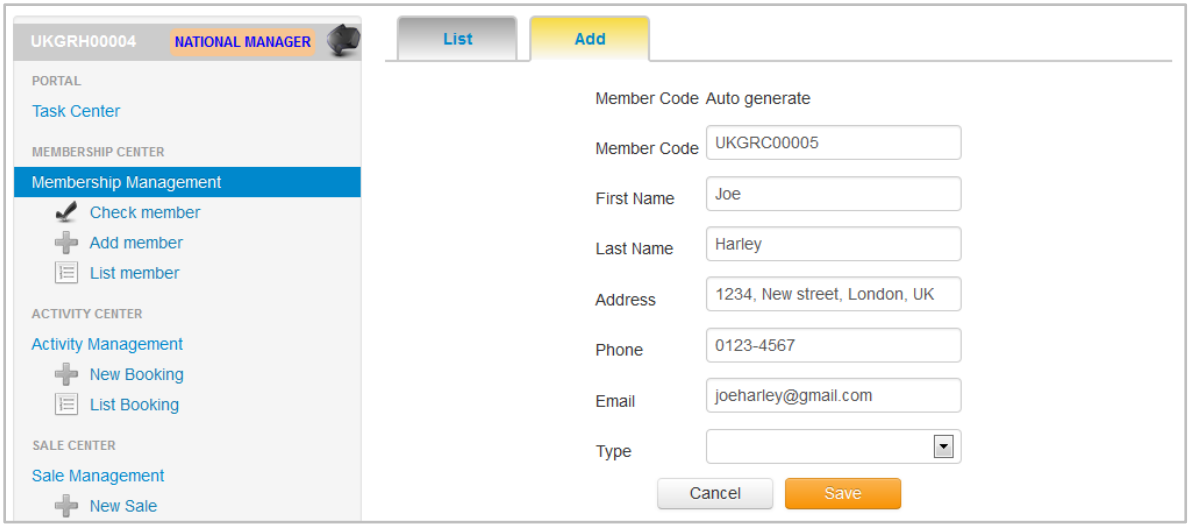


Figure 11: Add new membership

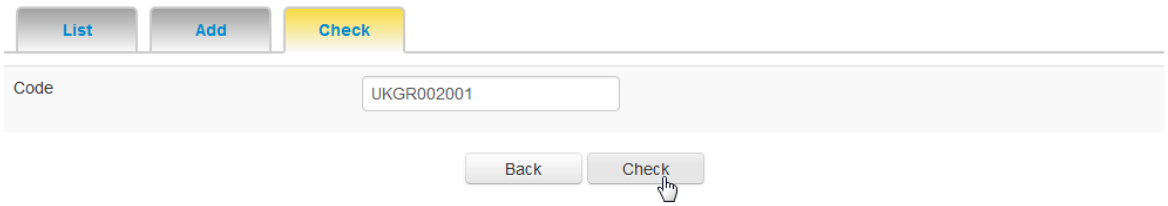


Figure 12: Check member

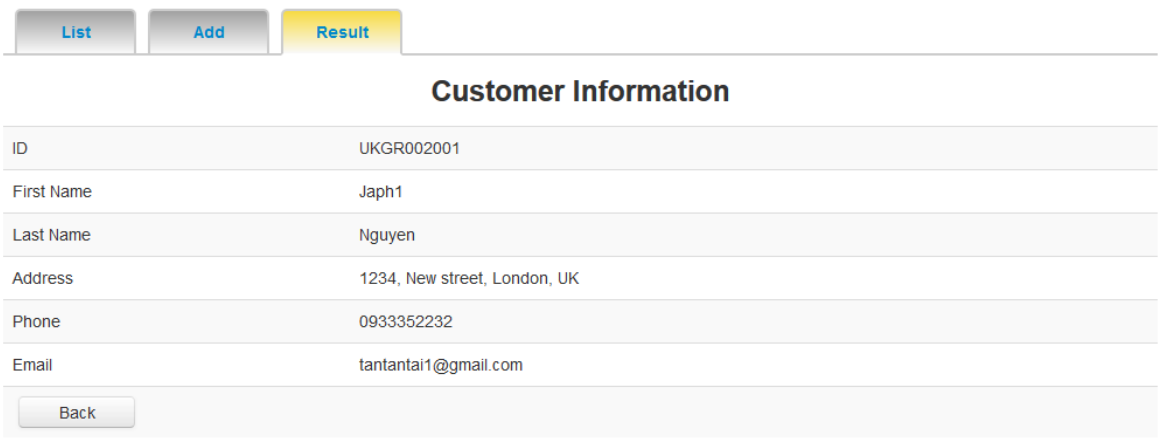


Figure 13: Check member result

4.3. Sale Management

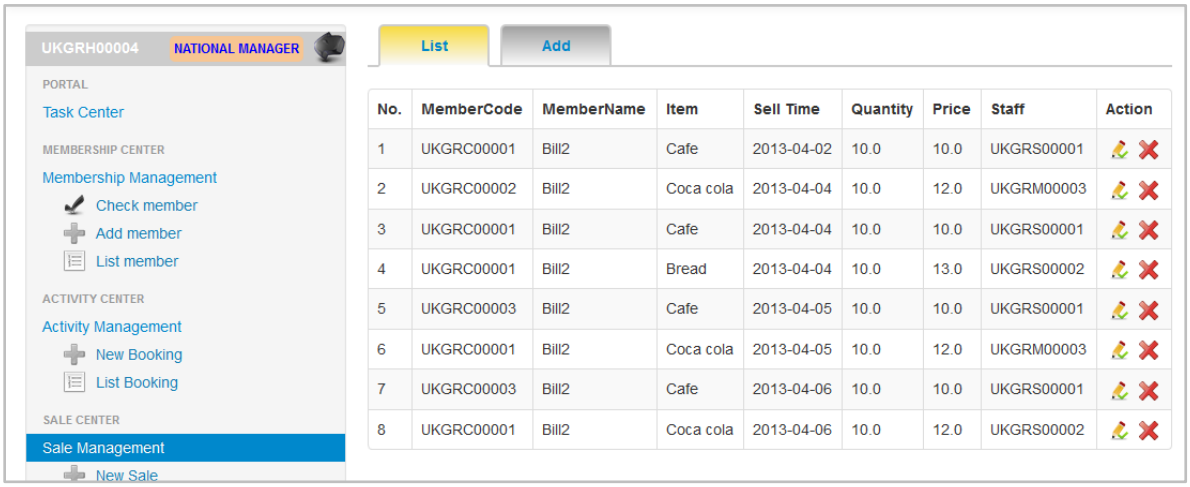


Figure 14: Sale record list

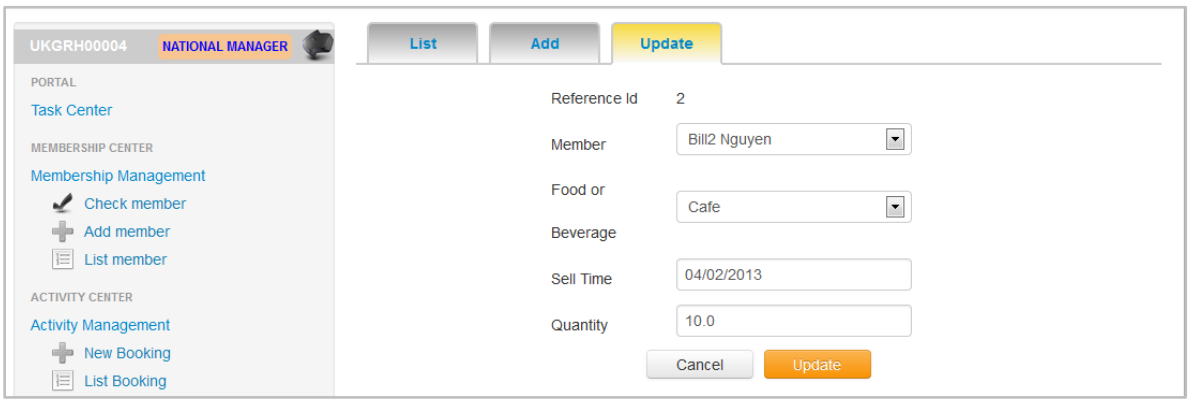


Figure 15: Update existing record

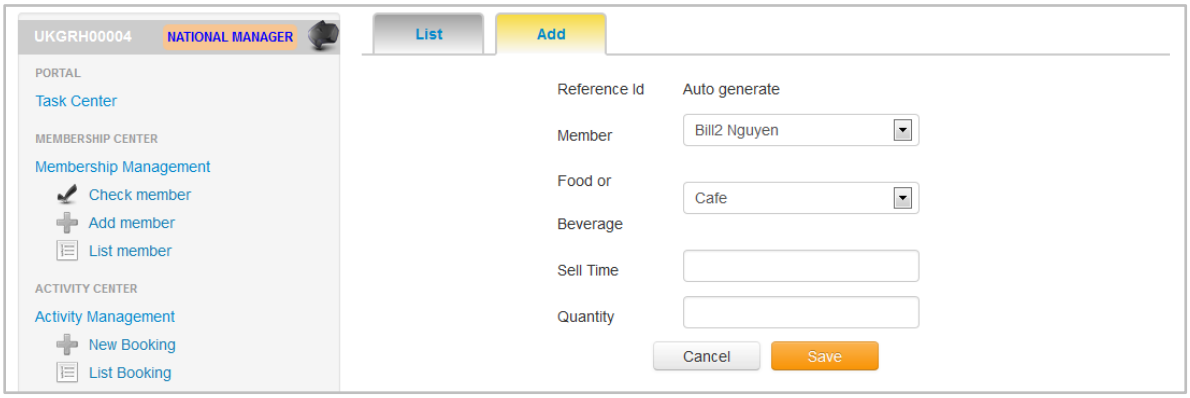
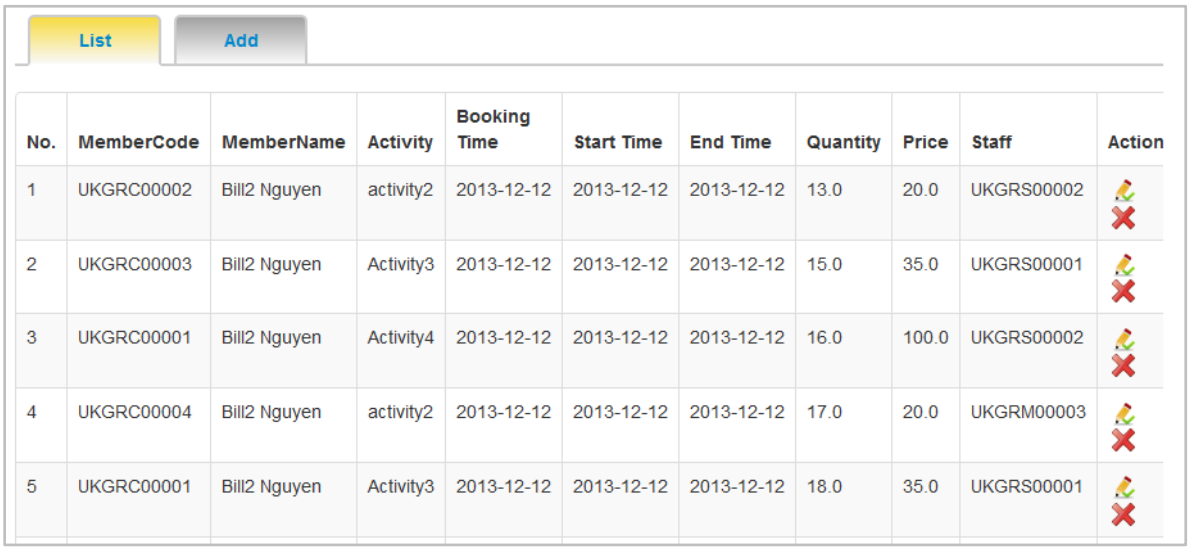


Figure 16: Add new record

4.4. Activity Management



No.	MemberCode	MemberName	Activity	Booking Time	Start Time	End Time	Quantity	Price	Staff	Action
1	UKGRC00002	Bill2 Nguyen	activity2	2013-12-12	2013-12-12	2013-12-12	13.0	20.0	UKGRS00002	
2	UKGRC00003	Bill2 Nguyen	Activity3	2013-12-12	2013-12-12	2013-12-12	15.0	35.0	UKGRS00001	
3	UKGRC00001	Bill2 Nguyen	Activity4	2013-12-12	2013-12-12	2013-12-12	16.0	100.0	UKGRS00002	
4	UKGRC00004	Bill2 Nguyen	activity2	2013-12-12	2013-12-12	2013-12-12	17.0	20.0	UKGRM00003	
5	UKGRC00001	Bill2 Nguyen	Activity3	2013-12-12	2013-12-12	2013-12-12	18.0	35.0	UKGRS00001	

Figure 17: List all activity orders

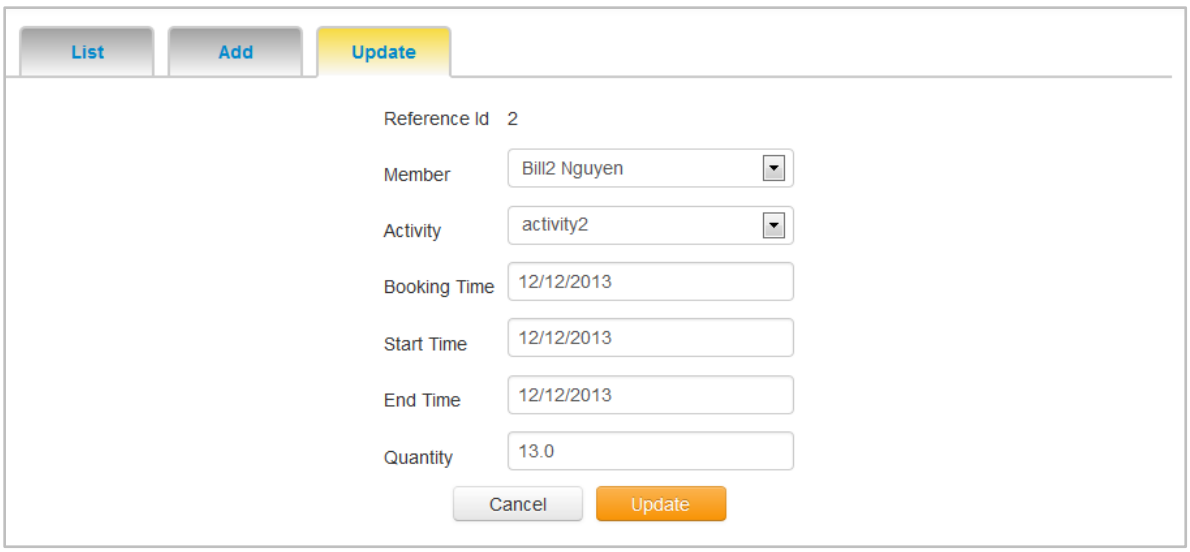


Figure 18: Update an order

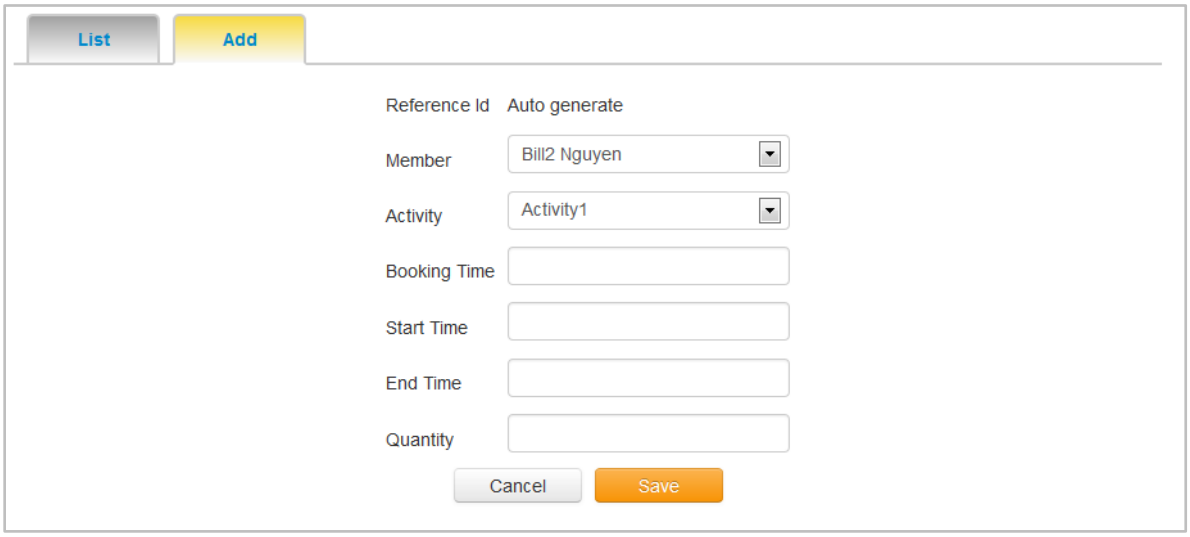


Figure 19: Add new activity order

4.5. Local report

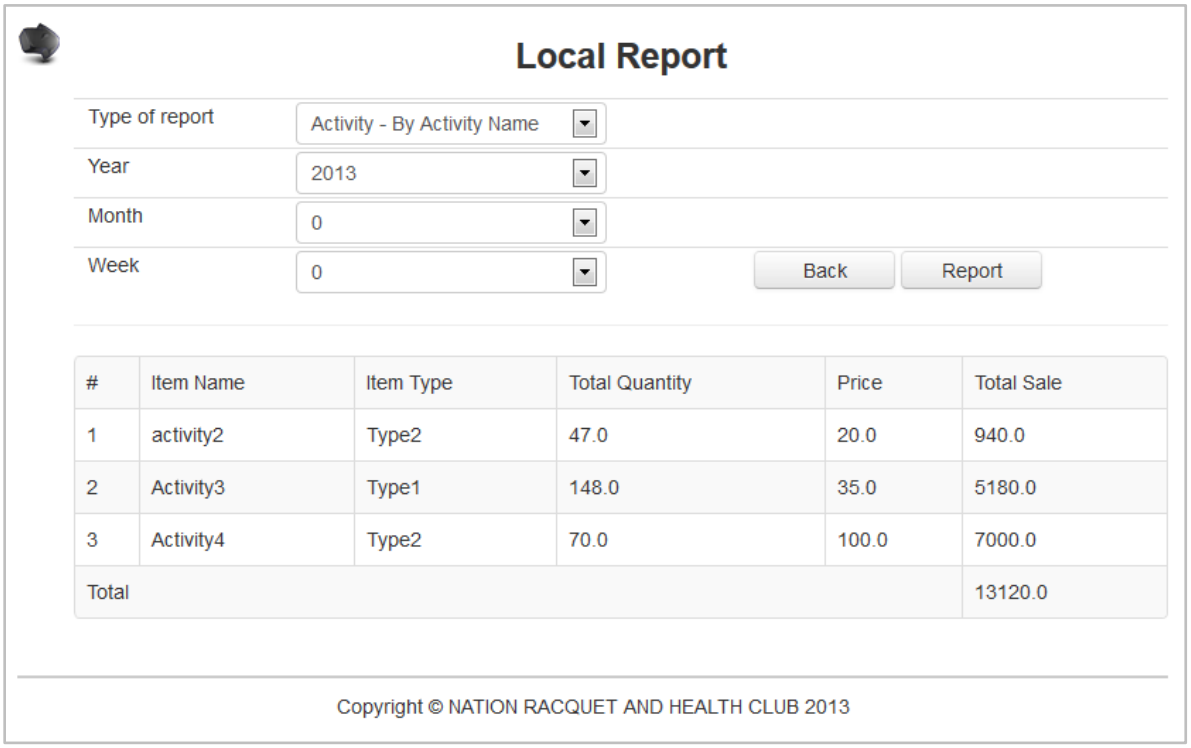


Figure 20: Local report - Activity report

Figure 21: Local report - Sale report

4.6. Local statistic

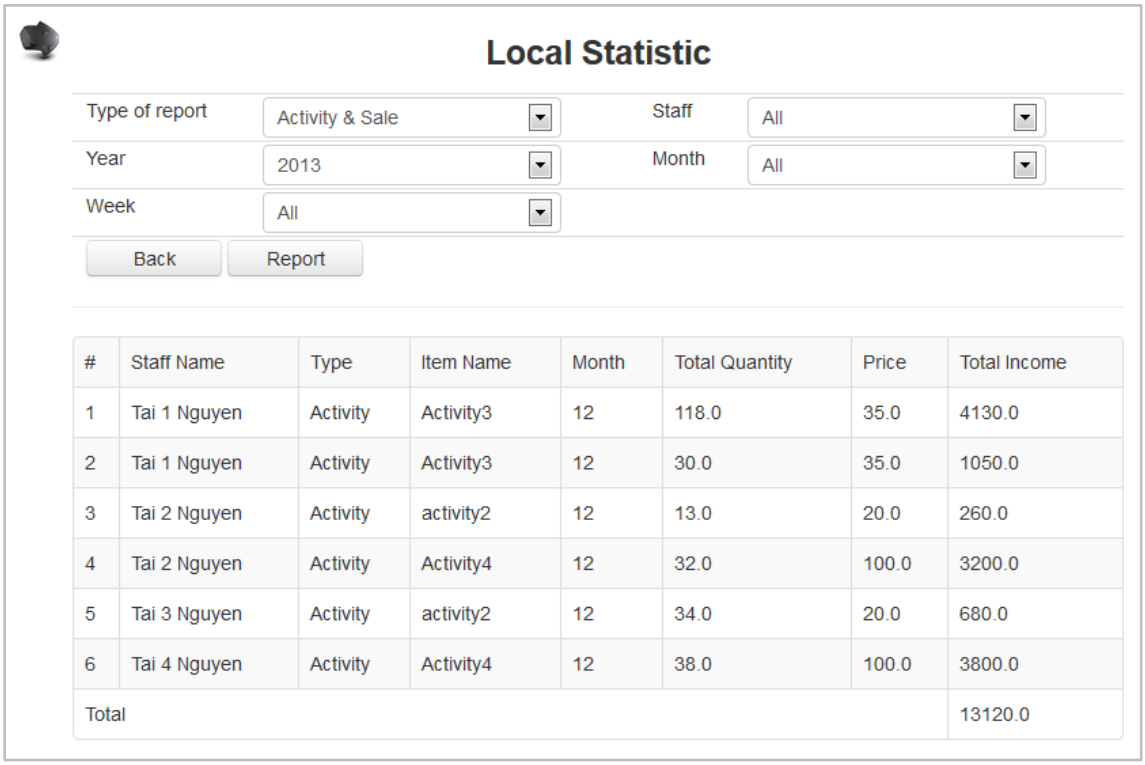


Figure 22: Local statistic - Activity statistic

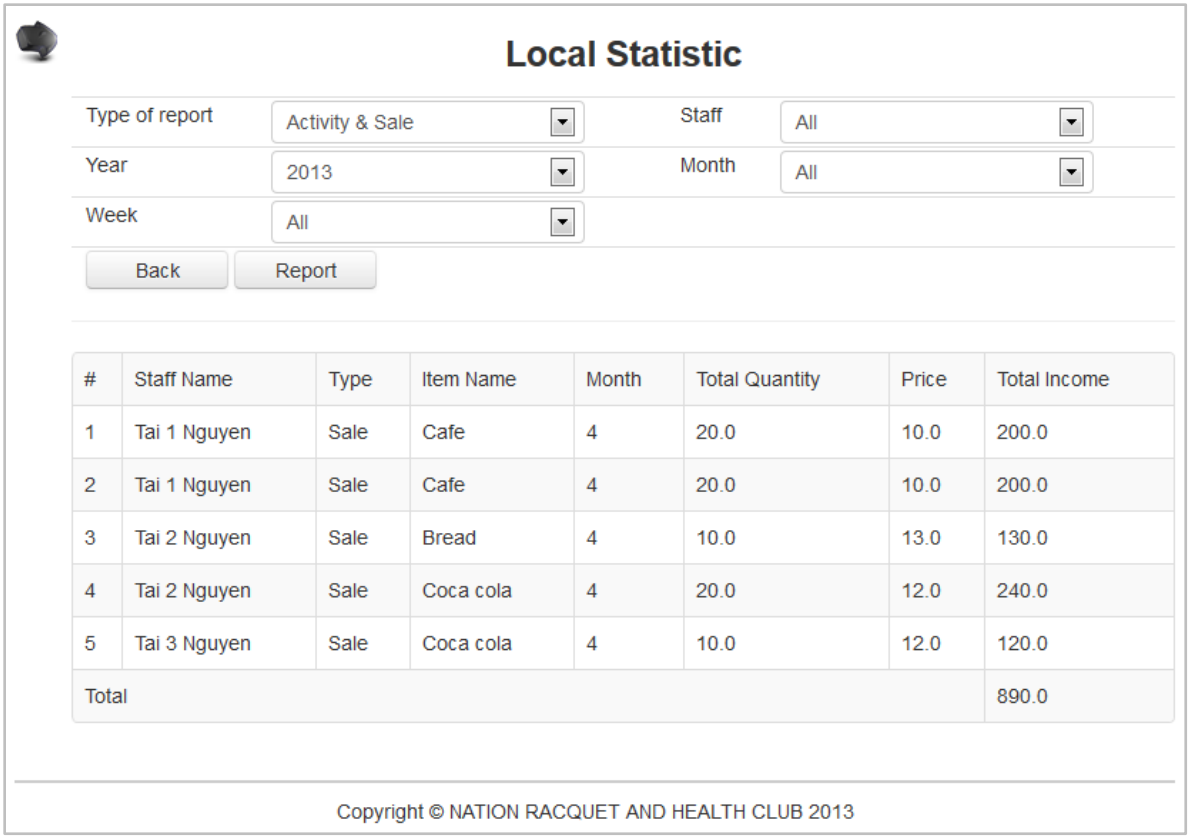
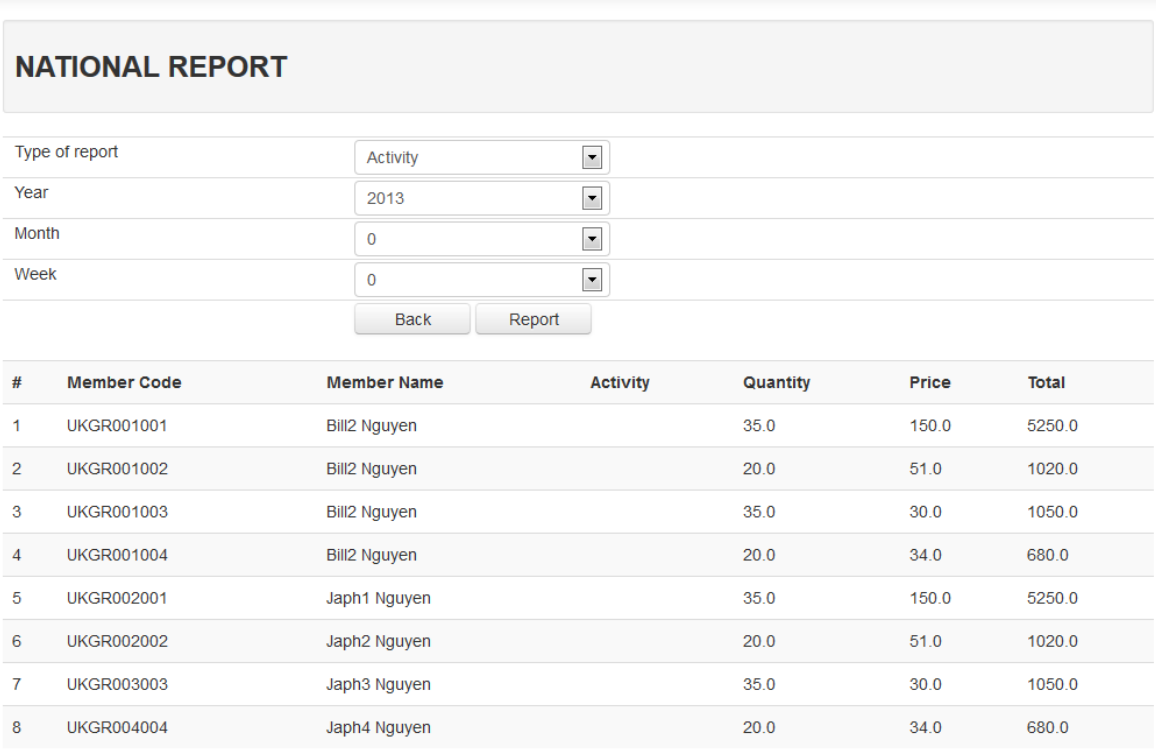


Figure 23: Local statistic - Sale statistic

4.7. National report



#	Member Code	Member Name	Activity	Quantity	Price	Total
1	UKGR001001	Bill2 Nguyen		35.0	150.0	5250.0
2	UKGR001002	Bill2 Nguyen		20.0	51.0	1020.0
3	UKGR001003	Bill2 Nguyen		35.0	30.0	1050.0
4	UKGR001004	Bill2 Nguyen		20.0	34.0	680.0
5	UKGR002001	Japh1 Nguyen		35.0	150.0	5250.0
6	UKGR002002	Japh2 Nguyen		20.0	51.0	1020.0
7	UKGR003003	Japh3 Nguyen		35.0	30.0	1050.0
8	UKGR004004	Japh4 Nguyen		20.0	34.0	680.0

Figure 24: National statistic

5. CRITICAL EVALUATION

This software prototype is 3-tier architecture. There is no doubt that there are three separate tiers in prototype architecture. Web browsers (e.g. Mozilla Firefox, Google Chrome, Microsoft IE, etc…) are represented as presentation tier. This tier is very flexible. User can use any kind of browser to connect to server. The logic tier is JSP/Servlets web container. This tier is responsible for handling business logic of the application. It also can be extended or narrowed due to the business need. The data tier, in this applet, is MySQL server. This tier store, process information, and provide data for logic tier. On the one hand, the three tiers cooperate tightly to produce an efficient system. On the other hand, each tier is also scalable. Every tier can be changed without affect the other tier. For instance, a new user can easy connect to the existing server using any kind of web client, even mobile phone web browser. Similarly, a new web or database server can be added to serve more users easily without disturb the existing servers.

The prototype software provides different interfaces for different users. To be more specific, users with staff permission (staff) can access all daily functions such as membership management,

activity order system, and sale system. However, they cannot access to restricted functions such as local and national report and statistic.

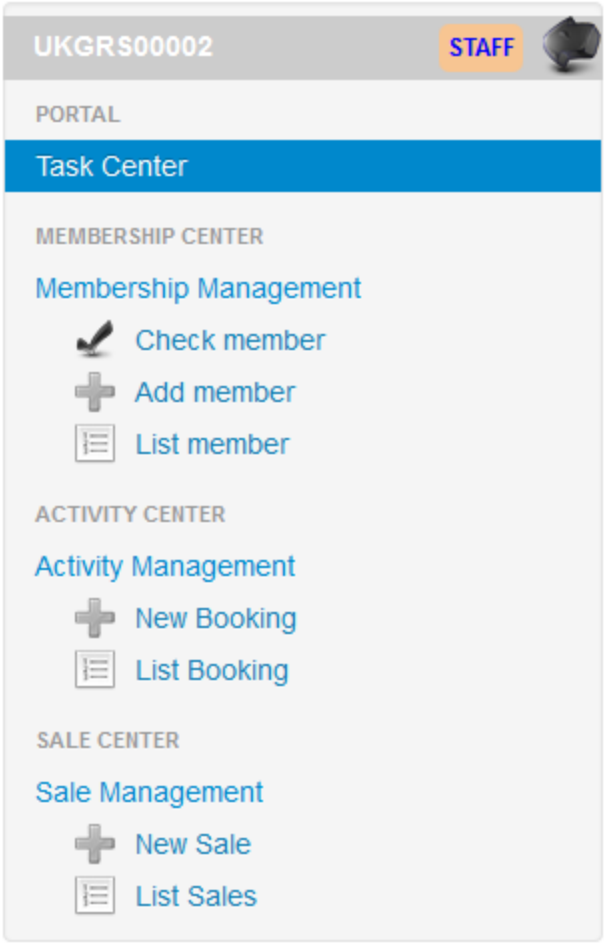


Figure 25: Functions for staff

Local managers are more powerful, they can do all functions of a normal staff plus local report function.

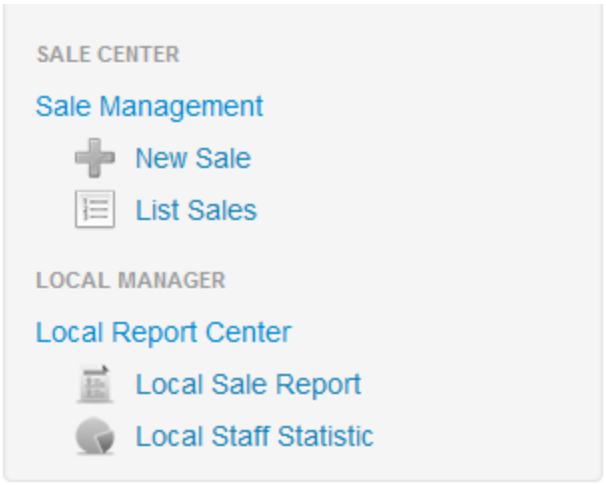


Figure 26: Functions for local manager

National manager are the most powerful user in the prototype. Besides the functions of a local manager, national manager can summary national data.

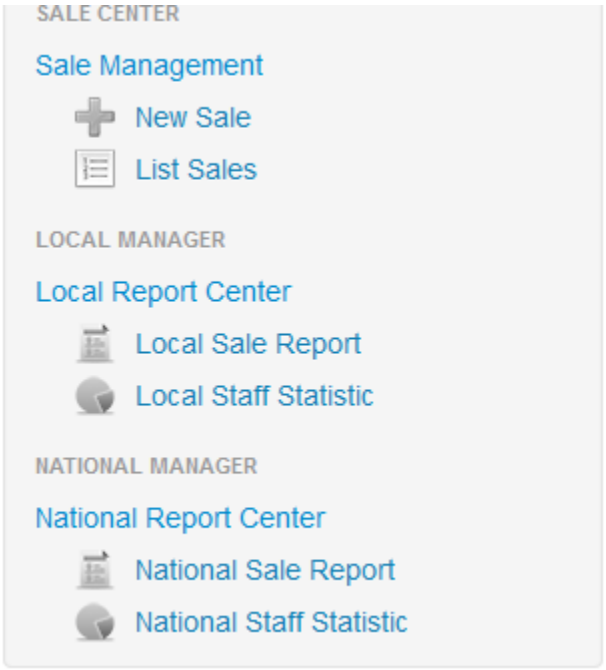


Figure 27: Function for national manager

Another vital important module in this prototype is the central web services. This web service is a bridge module which connects all centers together. It acts as a central gate for all national functions request. Without this module, the system will become a multiple single 3-tier applet.

LIST OF FIGURES:

LIST OF TABLES