Document Description:
This document is one of the most important steps of software project development that plays an important role at the design process before starting to implementation of Restaurant software project RAS-2005 that is, detailed design report of our Restaurant software project RAS-2005.

<table>
<thead>
<tr>
<th>Responsibility</th>
<th>Name</th>
<th>Prepared by</th>
<th>Inspected by</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>Çağdaş EKİNÇİ</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Designer</td>
<td>Atilla ACAR</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designer</td>
<td>Mevlüt BALLI</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designer</td>
<td>Özkan ÇELİK</td>
<td>√</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS

1. ABSTRACT
2. SCOPE
3. DATA DESIGN
4. ARCHITECTUAL AND COMPONENT LEVEL DESIGN
5. USER INTERFACES
6. PROJECT SCHEDULE
7. TESTING
8. CONCLUSION

ATTACHMENTS
   Gantt Chart 1 and 2
1. **ABSTRACT**

Since the demand for restaurant software is growing so rapidly in Turkey market parallel to rapid increase in restaurant market, restaurant software is developing for the effective restaurant organization and automatization of this business. RAS-2005 (Restaurant Automation System) is detailed software for restaurants that helps the organization in restaurants to be handled more easily. Since the market is growing rapidly parallel to the increase in population and demand to eat outside all restaurants need some automation software to become more organized and respectful situation in the market. Most important facilities needed in these software include, low price, easy usage, online ordering and reservation capability, capability to handle basic accounting, easy to adapt to, respectful technical support for free if possible, long term guaranty, security and integrity. RAS-2005 will be developed to serve to better organization in the eating market.

2. **SCOPE**

2.1 **General View**

Our software RAS-2005 is designed for make life easier for internet customers, restaurant owners, waiters in restaurants and customers that will come to restaurant for having their meals. When we look at the advantages for the internet using customers, we can see that life will be easier by online ordering system. They will be able to make online orderings via internet from their homes. Moreover, with the help of user name and password facility they will not be bothering with the internet related detailed jobs. When we come to the advantages of this software for the Restaurant owners, we will see that they will be able to manage their basic accounting such as income / expenditure position at any time they want. With the help of this software there will be an increase in the speed of the service given in the restaurant and the system will be better organized. This will lead to an increase in the number of the customers for the restaurant meaning more money for the restaurant owners. Another advantage of our software will help waiters in the restaurants in a way that they will not be coming and going that many times for a customer than they have been used to. This means waiters won’t get tired any more and will have better communication with the customers resulting in the satisfaction of the customers. In terms of the customer at restaurant, they will not be waiting for waiter anymore and they will save time from this fact. Again this will result in more satisfactory customers. In conclusion, RAS-2005 will
ease and automate ordering system over internet and palm devices. Better management and better organization in restaurants resulting in satisfied customers, restaurant owners, waiters and cooks and other staff in the restaurant.

2.2 Main Objectives
Our restaurant software RAS-2005 is a restaurant automation tool and a database system that increases efficiency in restaurants in respect of time, cost and customer satisfaction. Our software will replace the burdensome and paper based system that is currently in use.

RAS-2005 has designed separately for three types of users respectively cashier, waiters and web users. Each user will be using the software for performing the tasks that will be stated in Major Software Requirements section. Currently, our team members will be the only administrators of the RAS-2005 and will have access to all information stored in the system database. This access to the system resources will facilitate the tracking of errors and the system integrity to provide a better overall communications between the system and users of this software. The interactions with our software will be done through an easy to use Graphical User Interface (GUI), with different panes for several actions. Most of the commands will be done by just one clicking to the related buttons. In short, our software’s facilities will include stock management; online ordering via network enabled handheld computers, Internet online reservation and order management, customer management, simple accounting and executive information system.

2.3 Major Software Requirements
Below you can find the major software requirements of our system for each type of users respectively.

2.3.1 RAS-2005 Software is required to allow web users to accomplish the following:

- Register
- Web Login
- Set/Change Information
- Set Meal Order
- Confirm Meal Order
• Set Reservation
• Confirm Reservation
• View Reservations
• Logoff From Site

2.3.2 **RAS-2005 Software is required to allow waiters to accomplish the following:**
• Palm Login
• Choose Table
• Create Meal Order
• Add Item to Meal
• Remove Item From Meal
• Relate Orders With Orders

2.3.2 **RAS-2005 Software is required to allow cashier to accomplish the following:**
• Login to System
• Choose Accounting Menu
• Choose Profit/Loss Menu
• Choose Income Menu
• View Online Income
• View Restaurant Income
• Select Profit/Loss Period
• Choose Expenditure Menu
• View Feedstock Expenditure
• View General Expenditure
• View Staff Expenditure
• Choose Box Menu
• View Capital in Box
• Add Money to Box
• Take Money from the Box
• Choose Reservation Menu
• Select Reservations Period
• Cancel Reservations
• Confirm Made Reservations
• Print
• Choose Table Management Menu
• Close Table/View Bill of a Table
• View Order of a Table
• Change Order of a Table
• Cancel Order of a Table
• Transfer Order to another Table
• Change Table Properties
• Remove a Table
• Add New Table
• Choose Stock Control/Management Menu
• Choose Items Menu
• Delete an Item
• Change an Item’s Properties
• Print Stock Information
• Add New Item
• Choose Customer Menu
• Add New Customer
• Choose Customer’s List Menu
• Delete a Customer
• Change a Customer’s Information
• Print Customer’s Information
• Choose Workers Menu
• Add a New Worker
• Choose Worker’s List Menu
• Print Worker’s Information
• Delete a Worker
• Change a Worker’s Information
• Choose Settings Menu
• Set GUI color
• Set Message On/Off
• Set Table View

2.4 Major Hardware Requirements
RAS-2005 need following computer systems:
One computer system for being server of web site and database.
Some palms (number of it changes from restaurant to restaurant).

For developing processes:

★ Four computers those have at least following properties:

IBM PC or compatible
Intel Pentium III 500Mhz processor
128 MB SDRAM
20 GB Hard disk
Internet connection

★ One palm which runs on Windows CE operating system.
★ Bluetooth for providing connection between palm and server.

2.5 Major Design Constraints and Limitations

• Durability

Since the failure rate of web based and database related applications is higher than anyone expects, we will try to build our software as durable as possible during the building up process.

• Reliability

Most of the restaurant owners state that reliability is one of the most important factors to be considered carefully at developing stages of the software. They don’t want to have problems in terms of finding dependable software suppliers which will include adequate technical support in addition to purchasing high quality software components with as cheap as possible cost.

• Ease of Use

Since most of restaurant owners, waiters and web users are not very much familiar with computer world, it is very important for us to have software with as easy usable as possible. We plan to spent a lot of time on this issue for satisfaction of our customers namely, restaurant owners.
• **Enhanced Security**

Since our software will be web based and database related, security is another important issue for us to consider. We are planning to give importance to this matter more than ever and take our position in the market with the help of this topic.

• **Lack of Time**

RAS-2005 is expected to be finished by June 2004.

• **Portability**

RAS-2005 will be designed compatible with the Microsoft Windows and LINUX operating systems. Java compatible internet browser will be required for this purpose.

• **Extendibility**

RAS-2005 will be designed so that any needed additional modules will be able to added to the current system without any change in the current design of the system.

• **Maintainability**

The GUI and system modules of our system RAS-2005 are all separate, just requiring input and output from each other. Because of this modular design, we could change the internals of any module without causing problems for the other untouched modules. This makes the RAS-2005 easy to maintain, because changes to one module do not require the rewriting of other modules. Object-Oriented programming is a key fact that makes life easier for this purpose.
3. DATABASE

Figure 1: Entity Relationship Diagram
## 3.2 Data Dictionary

### Entities

<table>
<thead>
<tr>
<th>USERS</th>
<th>USER ID</th>
<th>PASS WORD</th>
<th>ROLE</th>
<th>NAME</th>
<th>SUR NAME</th>
<th>PHONE</th>
<th>MOBILE</th>
<th>ADDRESS</th>
<th>ZIP CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOMER</td>
<td>CUSTID</td>
<td>PASSWORD</td>
<td>NAME</td>
<td>SUR NAME</td>
<td>PHONE</td>
<td>MOBILE</td>
<td>ADDRESS</td>
<td>ZIPCODE</td>
<td></td>
</tr>
<tr>
<td>ORDER</td>
<td>ORDERID</td>
<td>USERID</td>
<td>CUSTID</td>
<td>MENUID</td>
<td>DATE</td>
<td>PAYMENT TYPE</td>
<td>TOTAL_PAYMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESERVE</td>
<td>CONFOR_NO</td>
<td>CUSTID</td>
<td>TABLEID</td>
<td>RES_DATE</td>
<td>RES_TIME</td>
<td>MADE_DATE</td>
<td>PARTY SIZE</td>
<td>COMMENTS</td>
<td></td>
</tr>
<tr>
<td>ASSIGN</td>
<td>USERID</td>
<td>TABLEID</td>
<td>CUSTID</td>
<td>STATUS</td>
<td>PARTYSIZE</td>
<td>TIME</td>
<td>DATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MENU</td>
<td>MENUID</td>
<td>NAME</td>
<td>PRICE</td>
<td>DESCRIPTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INVENTORY</td>
<td>ITEMID</td>
<td>SUPID</td>
<td>NAME</td>
<td>PRICE</td>
<td>QUANTITY</td>
<td>THRESHOLD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INV_MENU</td>
<td>ITEMID</td>
<td>MENUID</td>
<td>CATEGORYID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUPPLIER</td>
<td>SUPID</td>
<td>NAME</td>
<td>ADDRESS</td>
<td>CITY</td>
<td>ZIPCODE</td>
<td>PHONE</td>
<td>FAX</td>
<td>EMAIL</td>
<td>NOTE</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>CATEGORYID</td>
<td>DESCRIPTION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Explanation of Entities

**Table**: USERS  
**Key**: USERID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERID</td>
<td>Char</td>
<td>16</td>
<td>User ID: UNIQUE, NoEDIT</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Char</td>
<td>16</td>
<td>User password</td>
</tr>
<tr>
<td>ROLE</td>
<td>Char</td>
<td>10</td>
<td>Role of user: {“Admin”, “Waiter”, “Cashier”}</td>
</tr>
<tr>
<td>NAME</td>
<td>Char</td>
<td>20</td>
<td>Name of the user</td>
</tr>
<tr>
<td>SURNAME</td>
<td>Char</td>
<td>20</td>
<td>Surname of the user</td>
</tr>
<tr>
<td>PHONE</td>
<td>Char</td>
<td>10</td>
<td>Phone number of the user</td>
</tr>
<tr>
<td>MOBILE</td>
<td>Char</td>
<td>10</td>
<td>Mobile phone number of the user</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>Char</td>
<td>100</td>
<td>Address of the user</td>
</tr>
<tr>
<td>ZIPCODE</td>
<td>Char</td>
<td>5</td>
<td>Zipcode</td>
</tr>
</tbody>
</table>

**Table**: Customer  
**Key**: CUSTID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTID</td>
<td>Char</td>
<td>16</td>
<td>Cust ID: UNIQUE, NoEDIT</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>Char</td>
<td>16</td>
<td>Password of customer</td>
</tr>
<tr>
<td>NAME</td>
<td>Char</td>
<td>20</td>
<td>Name of the customer</td>
</tr>
<tr>
<td>SURNAME</td>
<td>Char</td>
<td>20</td>
<td>Surname of the customer</td>
</tr>
<tr>
<td>PHONE</td>
<td>Char</td>
<td>10</td>
<td>Phone number of the customer</td>
</tr>
<tr>
<td>MOBILE</td>
<td>Char</td>
<td>10</td>
<td>Mobile phone number of the customer</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>Char</td>
<td>100</td>
<td>Address of the customer</td>
</tr>
<tr>
<td>ZIPCODE</td>
<td>Char</td>
<td>5</td>
<td>Zipcode</td>
</tr>
</tbody>
</table>
### Table: Order
**Key:** ORDERID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORDERID</td>
<td>Char</td>
<td>8</td>
<td>Order ID: UNIQUE, NoEDIT</td>
</tr>
<tr>
<td>USERID</td>
<td>Char</td>
<td>16</td>
<td>User ID: flow from USERS.USERID, NoEDIT</td>
</tr>
<tr>
<td>CUSTID</td>
<td>Char</td>
<td>16</td>
<td>Customer ID: flow from CUSTOMER.CUSTID</td>
</tr>
<tr>
<td>MENUID</td>
<td>Num</td>
<td>3</td>
<td>Menu ID: flow from MENU.MENUID</td>
</tr>
<tr>
<td>DATE</td>
<td>Date</td>
<td>10</td>
<td>Date of order, assigned from system time</td>
</tr>
<tr>
<td>PAYMENT_TYPE</td>
<td>Char</td>
<td>15</td>
<td>Payment Type: (“Credit Card”, “Cash”, “Ticket”)</td>
</tr>
<tr>
<td>TOTAL_PAYMENT</td>
<td>Num</td>
<td>4</td>
<td>Total payment</td>
</tr>
</tbody>
</table>

### Table: Reserve
**Key:** CONFOR_NO

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONFOR_NO</td>
<td>Char</td>
<td>16</td>
<td>Conformation number: UNIQUE, NoEDIT</td>
</tr>
<tr>
<td>CUSTID</td>
<td>Char</td>
<td>16</td>
<td>Customer ID:flow from CUSTOMER.CUSTID</td>
</tr>
<tr>
<td>TABLEID</td>
<td>Num</td>
<td>3</td>
<td>Table ID of the selected table, MUST exist in TABLE.TABLEID</td>
</tr>
<tr>
<td>RES_DATE</td>
<td>Char</td>
<td>20</td>
<td>Reservation date, assigned by the customer</td>
</tr>
<tr>
<td>RES_TIME</td>
<td>Date</td>
<td>10</td>
<td>Reservation time, assigned by the customer</td>
</tr>
<tr>
<td>MADE_DATE</td>
<td>Char</td>
<td>15</td>
<td>The date at the moment of the making reservation</td>
</tr>
<tr>
<td>PARTYSIZE</td>
<td>Num</td>
<td>2</td>
<td>Total coming people</td>
</tr>
<tr>
<td>COMMENTS</td>
<td>Char</td>
<td>100</td>
<td>Additional information for reservation made by customer.</td>
</tr>
</tbody>
</table>
### Table: Assign

**Key:** USERID, TABLEID, CUSTID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERID</td>
<td>Char</td>
<td>16</td>
<td>User ID: flow from USERS.USERID</td>
</tr>
<tr>
<td>TABLEID</td>
<td>Char</td>
<td>16</td>
<td>Table ID of the selected table: MUST exist in TABLE.TABLEID</td>
</tr>
<tr>
<td>CUSTID</td>
<td>Char</td>
<td>16</td>
<td>Customer ID: flow from CUSTOMER.CUSTOMERID</td>
</tr>
<tr>
<td>STATUS</td>
<td>Char</td>
<td>1</td>
<td>Status of the table: (“Open”, “Closed”, “Ordered”, “Not Ordered”)</td>
</tr>
<tr>
<td>PARTYSIZE</td>
<td>Num</td>
<td>2</td>
<td>Total coming people</td>
</tr>
<tr>
<td>TIME</td>
<td>Char</td>
<td>15</td>
<td>The date at the moment of assigning the table</td>
</tr>
<tr>
<td>DATE</td>
<td>Num</td>
<td>4</td>
<td>The time of the assigning</td>
</tr>
</tbody>
</table>

### Table: Menu

**Key:** MENUID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MENUID</td>
<td>Num</td>
<td>3</td>
<td>Menu ID: AUTO, NoEDIT</td>
</tr>
<tr>
<td>NAME</td>
<td>Char</td>
<td>30</td>
<td>Name of the menu</td>
</tr>
<tr>
<td>PRICE</td>
<td>Num</td>
<td>4</td>
<td>Price of the menu</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Char</td>
<td>100</td>
<td>Description of the menu</td>
</tr>
</tbody>
</table>

### Table: Inventory

**Key:** ITEMID, SUPID, MENUID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEMID</td>
<td>Num</td>
<td>3</td>
<td>Item ID: AUTO, NoEDIT</td>
</tr>
<tr>
<td>SUPID</td>
<td>Num</td>
<td>3</td>
<td>Supplier ID: flow from SUPPLIER.SUPID</td>
</tr>
<tr>
<td>PRICE</td>
<td>Num</td>
<td>4</td>
<td>Price of the item</td>
</tr>
<tr>
<td>QUANTITY</td>
<td>Num</td>
<td>4</td>
<td>Total number of item</td>
</tr>
<tr>
<td>THRESHOLD</td>
<td>Num</td>
<td>4</td>
<td>The level of the quantity for ordering the item</td>
</tr>
</tbody>
</table>
Table: Inv_Menu

**Key:** ITEMID, MENUID, CATEGORYID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEMID</td>
<td>Num</td>
<td>3</td>
<td>Item ID: flow from INVENTORY.ITEMID</td>
</tr>
<tr>
<td>MENUID</td>
<td>Num</td>
<td>3</td>
<td>Menu ID: flow MENU.MENUID</td>
</tr>
<tr>
<td>CATEGORYID</td>
<td>Num</td>
<td>3</td>
<td>Category ID: flow from CATEGORY.CATEGORYID</td>
</tr>
</tbody>
</table>

Table: Supplier

**Key:** SUPID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUPID</td>
<td>Num</td>
<td>3</td>
<td>Supplier ID: AUTO, NoEDIT</td>
</tr>
<tr>
<td>NAME</td>
<td>Char</td>
<td>50</td>
<td>Name of the supplier</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>Char</td>
<td>100</td>
<td>Address of the supplier</td>
</tr>
<tr>
<td>CITY</td>
<td>Char</td>
<td>30</td>
<td>City of the supplier</td>
</tr>
<tr>
<td>ZIPCODE</td>
<td>Char</td>
<td>5</td>
<td>Zip code</td>
</tr>
<tr>
<td>PHONE</td>
<td>Char</td>
<td>10</td>
<td>Phone number of the supplier</td>
</tr>
<tr>
<td>FAX</td>
<td>Char</td>
<td>10</td>
<td>Fax number of the supplier</td>
</tr>
<tr>
<td>EMAIL</td>
<td>Char</td>
<td>50</td>
<td>Email of the supplier</td>
</tr>
<tr>
<td>NOTE</td>
<td>Char</td>
<td>100</td>
<td>Notes related with the supplier</td>
</tr>
</tbody>
</table>

Table: Category

**Key:** CATEGORYID

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATEGORYID</td>
<td>Num</td>
<td>3</td>
<td>Category ID: AUTO, NoEDIT</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Char</td>
<td>50</td>
<td>Description of the category</td>
</tr>
</tbody>
</table>
**Table:** Table

**Key:** TABLEID

<table>
<thead>
<tr>
<th>TABLEID</th>
<th>Num</th>
<th>3</th>
<th>Table ID: UNIQUE, NoEDIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIZE</td>
<td>Num</td>
<td>2</td>
<td>Total capacity of the table</td>
</tr>
</tbody>
</table>

**Table:** Account

**Key:** No Key

<table>
<thead>
<tr>
<th>DATE</th>
<th>Date</th>
<th>10</th>
<th>Table ID: UNIQUE, NoEDIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>Char</td>
<td>50</td>
<td>Type of the transaction: (&quot;Expenditure&quot;, &quot;Income&quot;)</td>
</tr>
<tr>
<td>AMOUNT</td>
<td>Num</td>
<td>4</td>
<td>Amount of the expenditure or income</td>
</tr>
</tbody>
</table>

**Legend**

<table>
<thead>
<tr>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIQUE</td>
</tr>
<tr>
<td>AUTO</td>
</tr>
<tr>
<td>NoEDIT</td>
</tr>
<tr>
<td>Flow ...</td>
</tr>
<tr>
<td>Exist ...</td>
</tr>
</tbody>
</table>
3.3 SQL Codes

CREATE TABLE USER (  
  USERID CHAR(16) NOT NULL,  
  PASSWORD CHAR(16) NOT NULL,  
  ROLE CHAR(10) NOT NULL,  
  NAME CHAR(20) NOT NULL,  
  SURNAME CHAR(20) NOT NULL,  
  PHONE CHAR(10) NOT NULL,  
  MOBILE CHAR(10),  
  ADDRESS CHAR(100) NOT NULL,  
  ZIPCODE CHAR(5),  
  PRIMARY KEY (USERID)  
);  

CREATE TABLE CUSTOMER (  
  CUSTIDID CHAR(16) NOT NULL,  
  PASSWORD CHAR(16) NOT NULL,  
  NAME CHAR(20) NOT NULL,  
  SURNAME CHAR(20) NOT NULL,  
  PHONE CHAR(10) NOT NULL,  
  MOBILE CHAR(10),  
  ADDRESS CHAR(100) NOT NULL,  
  ZIPCODE CHAR(5),  
  PRIMARY KEY (CUSTID)  
);  

CREATE TABLE ORDER (  
  ORDERID CHAR(16) NOT NULL,  
  USERID CHAR(16) NOT NULL,  
  CUSTID CHAR(20) NOT NULL,  
  MENUID INT(3) NOT NULL,  
  PRIMARY KEY (ORDERID)  
);
DATE DATE NOT NULL,
PAYMENT_TYPE CHAR(15) NOT NULL,
TOTAL_PAYMENT REAL NOT NULL,
PRIMARY KEY (CUSTID)
);

CREATE TABLE RESERVE
(
CONFOR_NO CHAR(16) NOT NULL,
CUSTID CHAR(20) NOT NULL,
TABLEID INT(3) NOT NULL,
RES_DATE DATE NOT NULL,
RES_TIME DATE NOT NULL,
MADE_DATE DATE NOT NULL,
PARTYSIZE INT(2) NOT NULL,
COMMENTS CHAR(100),
PRIMARY KEY (CONFOR_NO),
FOREIGN KEY (CUSTID) REFERENCES CUSTOMER(CUSTID)
);

CREATE TABLE ASSIGN
(
USERID CHAR(16) NOT NULL,
CUSTID CHAR(20) NOT NULL,
TABLEID INT(3) NOT NULL,
STATUS CHAR(1) NOT NULL,
PARTYSIZE CHAR(2) NOT NULL,
TIME DATE NOT NULL,
DATE DATE NOT NULL,
PRIMARY KEY (USERID, CUSTID),
FOREIGN KEY (CUSTID) REFERENCES CUSTOMER(CUSTID),
FOREIGN KEY (USERID) REFERENCES CUSTOMER(USERID)
);
CREATE TABLE MENU
(
  MENU INT(3) NOT NULL,
  NAME CHAR(30) NOT NULL,
  PRICE REAL NOT NULL,
  DESCRIPTION CHAR(100) NOT NULL,
  PRIMARY KEY (MENUID)
);

CREATE TABLE INVENTORY
(
  ITEMID INT(3) NOT NULL,
  SUPID INT(3) NOT NULL,
  PRICE REAL NOT NULL,
  QUANTITY INT(4) NOT NULL,
  THRESHOLD INT(4) NOT NULL,
  PRIMARY KEY (ITEMID),
);

CREATE TABLE INV_MENU
(
  ITEMID INT(3) NOT NULL,
  MENUID INT(3) NOT NULL,
  CATEGORYID INT(3) NOT NULL,
  PRIMARY KEY (ITEMID),
  FOREIGN KEY (ITEMID) REFERENCES INVENTORY(ITEMID),
  FOREIGN KEY (MENUID) REFERENCES MENU(MENUID),
  FOREIGN KEY (CATEORYID) REFERENCES CATEGORY(CATEORYID)
);

CREATE TABLE SUPPLIER
(
  SUPID INT(3) NOT NULL,
  NAME CHAR(50) NOT NULL,
  ADDRESS CHAR(100) NOT NULL,
  CITY CHAR(30) NOT NULL,
  PRIMARY KEY (SUPID)
);
CREATE TABLE CATEGORY (  CATEGORYID INT(3) NOT NULL,  DESCRIPTION CHAR(50) NOT NULL,  PRIMARY KEY (CATEGORYID) ) ;

CREATE TABLE TABLE (  TABLEID INT(3) NOT NULL,  SIZE INT(2) NOT NULL,  PRIMARY KEY (TABLEID) ) ;

CREATE TABLE ACCOUNT (  DATE INT(10) NOT NULL,  TYPE CHAR(50) NOT NULL,  AMOUNT(4) NOT NULL,  PRIMARY KEY (TYPE) ) ;
4. ARCHITECTURAL DESIGN

USE CASE DIAGRAMS:
Use case analysis performed by Atilla Acar and Çağdaş Ekinci. Our main aim here is to determine actors and their activities. Use case diagrams based on their actor group. They are web users, Cashiers/Administrators and Waiters. In diagrams one can see related actions that can be performed by each user group.

Figure 1: Use case of WebUser

Figure 2: Use case of Waiter
Figure 3: Use case of Cashier
DATA FLOW DIAGRAMS

In this section we try to explain our system functionality with Data flow diagrams and state transition diagrams.

Data flow diagrams is listed below:
Data Flow diagram for Web Based Application

Data Dictionaries for Web Based Application of RAS-2005:

Name: User Data and Commands
Input: Interact with user
Output: Online reservation or Online Ordering
Explanation: After a successful login user send his/her wishes through internet browser and our web-based application should be show him Online reservation panel or Online ordering panel and wait for user request.

Name: User name and Password
Input: Interact with user
Output: Login successful (Online reservation and Online Ordering panel)
Explanation: To enter our system user should be enter his password and username and our system seek them in database if it finds such a user with valid password user will be able to make an order or reservation.
Name: Valid visa card
Input: BANK
Output: Online ordering
Explanation: To make an order user must have a valid visa card. Our system decide a visa card valid or not valid and to achieve this our system connect with bank and check visa card validity.

Name: Request Order
Input: Online ordering
Output: System Server
Explanation: User request an order and our system back to user with his request show delivery time and other issues.

Name: Request reservation
Input: Online reservation
Output: System Server
Explanation: User request a reservation and our system back to user with his request show his request.

Name: Messages back to user
Input: Display messages and status for internet user
Output: Online display
Explanation: In each step user see messages and status trough web browser. If user select a operation our system perform this operation and send back to user.
Data Flow diagram for palm application of our System

Data Dictionaries for palm application of RAS-2005:
Name: User Data and Commands
Input: Interact with palm user
Output: Palm operation
Explanation: After a successful login user send his/her wishes through palm and our palm application should show him Palm operations which include select table, select order, select customer.

Name: User name and Password
Input: Interact with user
Output: Login successful (palm operations)
Explanation: To enter our system user should enter his password and username and our system seek them in database if it finds such a user with valid password user will be able to chose palm operation.

Name: Messages back to user
Input: Display messages and status for palm user
Output: Palm display
Explanation: In each step user see messages and status through palm. If user selects an operation, our system perform this operation and send back to user.
Data Flow diagram for main application in restaurant

Data Dictionaries for application in restaurant of RAS-2005:
Name: User Data and Commands
Input: Interact with user
Output: System server
Explanation: After a successful login user send his/her wishes through our main application GUI and our main application should be show him admin operations system server application panel and wait for user request.

Name: User name and Password
Input: Interact with user
Output: Login successful (admin operations system server)
Explanation: To enter our system user should be enter his password and username and our system seek them in database if it finds such a user with valid password user will be able to enter system server and admin operations.

Name: Print document or bill
Input: System server
Output: Cashier printer cook printer
Explanation: After closing table our system must be send bill info to cashier printer. Also if an order requested from palms or internet our system must be inform cook with help of cook printer.
Name: Messages back to user
Input: Display messages and status for main application user
Output: main application panel
Explanation: In each step user see messages and status. If user select an operation, our system perform this operation and send back to user.

Data Flow diagram for system server

Data Dictionaries for System server:
Name: User Data and Commands
Input: Interact with user
Output: select operation
Explanation: After a successful login user send his/her wishes through our main application gui and our main application should be show him select operation panel and wait for user request.

Name: Update customer information
Input: Stock control
Output: Customer management
Explanation: Admin and cashier should be able to update customer information or our system automatically achieve this for online customers. For admin and cashier our system get related information adn change database for it.

Name: Request for reservation
Input: Table reservation
Output: Stock control
Explanation: a request from internet or from main aplication handled by our system. If table is free for given time user informed that it is free else user informed that it is not free.

Name: Related information for accounting
Input: Database
Output: Accounting
Explanation: Admin and cashier should be able to view profit/loss and other accounting data. Our system connect with database and return requested data.
Web based application state transition diagram
Palm state Transition diagram
Main application State Transition diagram.
SYSTEM HIRARCY CHARTS

Hierarchy Chart for Web based application.

Hierarchy Chart for Palm application.
ACTIVATION DIAGRAMS

Activity diagrams show our system and user interactions they are designed by Atilla ACAR for initial design part and we added some explanation if necessary. Below there exits all activations of our program and it includes user and system interaction. If user is just stated as “USER” it means it can be a waiter an admin a cashier or a customer. Because some of the activation is same for all user groups we decided to show them in a general diagram.
Activity diagram for registration a new user
Activity diagram for login process

Activity diagram for logout process
Activity diagram for set of process of web user
Activity diagram for set of process of waiter
Activity diagram for accounting processes
Activity diagram for reservation processes
Activity diagram for table and order management processes
Activity diagram for customer management processes
Activity diagram for workers management processes
Activity diagram for stock control and management processes
Activity diagram User interface and other Settings processes
Activity diagram for message handling processes
SEQUENCE DIAGRAMS
1_ Sequence Diagrams of WebUser

1.1_ MakeOnlineOrder

1: Enter username and password
2: Login failed
3: Make meal choice
4: Express address of ordering
5: Determine payment choice
6: Do confirmation
7: Anonimous login

1.2_ MakeOnlineReservation

1: Enter username and password
2: Login failed
3: Make date choice
4: Confirm reservation
5: Anonimous login
1.3_ ChangePersonalInformation

: WebUser

1: Enter username and password

2: Login failed

3: Make change

1.4_ ViewReservation

: WebUser

1: Enter username and password

2: Login failed

3: View reservations
1.5_ MakeRegistration

2_ Sequence Diagrams of Waiter

2.1_ TakeOrdering
3 _ Sequence Diagrams of Cashier

3.1 _ PrintLastMonthsOnlineIncome

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select online income
6: Adjust period to last month
7: Print document

3.2 _ PrintLastMonthsRestaurantIncome

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select restaurant income
6: Adjust period to last month
7: Print document
3.3_ PrintLastWeeksFeedstockExpenditure

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select feedstock expenditure
6: Adjust period to last week
7: Print document

3.4_ PrintLastWeeksStuffExpenditure

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select stuff expenditure
6: Adjust period to last week
7: Print document
3.5_ PrintLastWeeksGeneralExpenditure

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select general expenditure
6: Adjust period to last week
7: Print document

3.6_ AddMoneyToBox

1: Enter username and password
2: Login failed
3: Look box process
4: Select view
5: Add money to box
6: Add money to box
3.7_ TakeMoneyFromBox

1: Enter username and password
2: Login failed
3: Look box process
4: Select view
5: Take money from box
6: Take money from box

3.8_ CancelReservation

1: Enter username and password
2: Login failed
3: Look reservation process
4: Select period of reservations
5: Choose reservation
3.9_ConfirmReservation

1. Enter username and password
2. Login failed
3. Look reservation process
4. Select period of reservations
5. Choose reservation

3.10_PrintReservation

1. Enter username and password
2. Login failed
3. Look reservation process
4. Select period of reservations
5. Choose reservation
3.11_PrintBillOfTable

1: Enter username and password
2: Login failed
3: Look table management process
4: Close table / view bill
5: Print document

3.12_ChangeOrderOfTable

1: Enter username and password
2: Login failed
3: Look table management process
4: View order of a table
5: Change order
3.13_ CancelOrder

1. Enter username and password
2. Login failed
3. Look table management process
4. View order of a table
5. Cancel order

3.14_ TransferTable

1. Enter username and password
2. Login failed
3. Look table management process
4. Choose table
5. Transfer table
3.15 RemoveTable

1. Enter username and password
2. Login failed
3. Look table management process
4. Change table properties
5. Choose table

3.16 AddNewTable

1. Enter username and password
2. Login failed
3. Look table management process
4. Set table properties
3.17_ DeleteItemFromStock

1: Enter username and password
2: Login failed
3: Look stock management process
4: Select item
5: Delete selected item

3.18_ AddNewItemToStock

1: Enter username and password
2: Login failed
3: Look stock management process
4: Set item properties
3.19_ PrintStockItem

Cashier

1: Enter username and password
2: Login failed
3: Look stock management process
4: Print items in the stock

3.20_ ChangeItemsProperties

Cashier

1: Enter username and password
2: Login failed
3: Look item control process
4: Choose an item
5: Change items properties
3.21_ DeleteRegisteredCustomer

1: Enter username and password
2: Login failed
3: Look customer control process
4: Select customer
5: Delete selected customer

3.22_ AddNewCustomer

1: Enter username and password
2: Login failed
3: Look customer management process
4: Set customers properties
3.23_ PrintCustomerInformation

3.24_ ChangeCustomersProperties
3.25_ DeleteWorker

1: Enter username and password
2: Login failed
3: Look worker control process
4: Select worker
5: Delete selected worker

3.26_ ChangeWorkersProperties

1: Enter username and password
2: Login failed
3: Look worker control process
4: Choose worker
5: Change workers properties
3.27 AddNewWorker

Cashier

1: Enter username and password

2: Login failed

3: Look worker management process

4: Set workers properties

Worker Management Menu

Add Worker

3.28 PrintWorkerInformation

Cashier

1: Enter username and password

2: Login failed

3: Look worker control process

4: Choose worker

5: Print information of worker

Login

Worker Management Menu

Worker List

Printer Output
3.29_ SetGUIColor

1: Enter username and password
2: Login failed
3: Look settings process
4: Choose color

3.30_ SetMessageOnOff

1: Enter username and password
2: Login failed
3: Look settings process
4: Choose on or off
3.31_ SetTable View

- Cashier

1: Enter username and password

2: Login failed

3: Look settings process

4: Choose listview or overview
COLLABORATION DIAGRAMS

1. Collaboration Diagrams of WebUser

1.1. MakeOnlineOrder

1: Enter username and password
2: Login failed
3: Make meal choice
4: Express address of ordering
5: Determine payment choice
6: Do confirmation
7: Anonymous login

1.2. MakeOnlineReservation

1: Enter username and password
2: Login failed
3: Make date choice
4: Confirm reservation
5: Anonymous login

1.3_ ChangePersonalInformation

1. Enter username and password
2. Login failed
3. Make change

1.4_ ViewReservation

1. Enter username and password
2. Login failed
3. View reservations

1.5_ MakeRegistration

1. Set user information

: WebUser

User Entered Data

Login

Confirmed Registrations

Confirm Registration

: WebUser
2_ Collaboration Diagrams of Waiter

2.1_ TakeOrdering

1: Enter username and password  
2: Login failed  
3: Mark the customers table  
4: Organize ordering; add-remove items  
5: If available identify customer  

: Waiter

Meals List

Relate Customer

Table Selection

3_ Collaboration Diagrams of Cashier

3.1_ PrintLastMonthsOnlineIncome

1: Enter username and password  
2: Login failed  
3: Look accounting process  
4: Choose profit/loss section  
5: Select online income  
6: Adjust period to last month  
7: Print document  

: Cashier

Profit/Loss Operations

Income List

Printer Output

Determine Period

Accounting Menu
3.2_ PrintLastMonthsRestaurantIncome

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select restaurant income
6: Adjust period to last month
7: Print document

3.3_ PrintLastWeeksFeedstockExpenditure

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select feedstock expenditure
6: Adjust period to last week
7: Print document
3.4_ PrintLastWeeksStuffExpenditure

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select stuff expenditure
6: Adjust period to last week
7: Print document

3.5_ PrintLastWeeksGeneralExpenditure

1: Enter username and password
2: Login failed
3: Look accounting process
4: Choose profit/loss section
5: Select general expenditure
6: Adjust period to last week
7: Print document
3.6_ AddMoneyToBox

2: Login failed

1: Enter username and password

3: Look box process

5: Add money to box

6: Add money to box

3.7_ TakeMoneyFromBox

2: Login failed

1: Enter username and password

3: Look box process

5: Take money from box

6: Take money from box
3.8 CancelReservation

1: Enter username and password
2: Login failed
3: Look reservation process
4: Select period of reservations
5: Choose reservation

: Cashier

Reservation Lists

Cancel Reservation

3.9 ConfirmReservation

1: Enter username and password
2: Login failed
3: Look reservation process
4: Select period of reservations
5: Choose reservation

: Cashier

Reservation Lists

Confirm Reservation
3.10_ PrintReservation

1: Enter username and password
2: Login failed
3: Look reservation process
4: Select period of reservations
5: Choose reservation

Cashier → Login → Reservation Lists → Printer Output

3.11_ PrintBillOfTable

1: Enter username and password
2: Login failed
3: Look table management process
4: Close table / view bill
5: Print document

Cashier → Login → Table Management Menu → Printer Output
3.12_ ChangeOrderOfTable

1: Enter username and password
2: Login failed
3: Look table management process
4: View order of a table
5: Change order

Cashier
Login
Table Management Menu

Confirm
Change
Table Operations

3.13_ CancelOrder

1: Enter username and password
2: Login failed
3: Look table management process
4: View order of a table
5: Cancel order

Cashier
Login
Table Management Menu

Confirm
Change
Table Operations
3.14_ TransferTable

1: Enter username and password
2: Login failed
3: Look table management process
4: Choose table
5: Transfer table

3.15_ RemoveTable

1: Enter username and password
2: Login failed
3: Look table management process
4: Change table properties
5: Choose table
3.16_ AddNewTable

1: Enter username and password

2: Login failed

: Cashier

3: Look table management process

4: Set table properties

Table Management Menu

Add Table

3.17_ DeleteItemFromStock

1: Enter username and password

2: Login failed

: Cashier

3: Look stock management process

4: Select item

5: Delete selected item

Stock Control/Management Menu

Items List

Confirm Deletion
3.18_ AddNewItemToStock

1: Enter username and password → Login

Cashier → Login

2: Login failed

3: Look stock management process

Add Item → Stock Control/Management Menu

4: Set item properties

3.19_ PrintStockItem

1: Enter username and password → Login

Cashier → Login

2: Login failed

3: Look stock management process

Printer Output → Stock Control/Management Menu

4: Print items in the stock
3.20_ ChangeItemsProperties

1: Enter username and password
2: Login failed
3: Look item control process
4: Choose an item
5: Change items properties

Cashier
Stock Control/Management Menu
Items List
Confirm Change

3.21_ DeleteRegisteredCustomer

1: Enter username and password
2: Login failed
3: Look customer control process
4: Select customer
5: Delete selected customer

Cashier
Customer Management Menu
Customers List
Confirm Deletion
3.22_ AddNewCustomer

1: Enter username and password
2: Login failed
3: Look customer management process
4: Set customers properties

Add Customer

Customer Management Menu

3.23_ PrintCustomerInformation

1: Enter username and password
2: Login failed
3: Look customer control process
4: Choose customer
5: Print information of customer
3.24_ ChangeCustomersProperties

1: Enter username and password

2: Login failed

4: Choose customer

5: Change customers properties

Customer Management Menu

Login

1: Enter username and password

3: Look customer control process

Customer List

Confirm Change

3.25_ DeleteWorker

1: Enter username and password

2: Login failed

3: Look worker control process

4: Select worker

5: Delete selected worker

Worker Management Menu

Login

Worker List

Confirm Deletion
3.26 ChangeWorkersProperties

- 1: Enter username and password
- 2: Login failed
- 3: Look worker control process
- 4: Choose worker
- 5: Change workers properties

Worker Management Menu

Worker List

3.27 AddNewWorker

- 1: Enter username and password
- 2: Login failed
- 3: Look worker management process
- 4: Set workers properties

Worker Management Menu

Add Worker

Cashier
3.28_ PrintWorkerInformation

1: Enter username and password
2: Login failed
3: Look worker control process
4: Choose worker
5: Print information of worker

Cashier

Login

Worker Management Menu

Worker List

Printer Output

3.29_ SetGUIColor

1: Enter username and password
2: Login failed
3: Look settings process
4: Choose color

Cashier

Login

Set GUI Color

Settings Menu
3.30_ SetMessageOnOff

1: Enter username and password

2: Login failed

: Cashier

3: Look settings process

4: Choose on or off

Set Message Warning

3.31_ SetTableView

1: Enter username and password

2: Login failed

: Cashier

3: Look settings process

4: Choose listview or overview

Set TableView
MODULE DESIGN
In this section, you will find the procedural design of the RAS2005. The first subsection is the
login module which is the primary entry module for the other modules except registration
module, Above all, only if the login module permits, the other modules are executed again
except registration module.

MODULES FOR WEB PAGE
1. Registration Module
   //Registration Module for web based application
   BEGIN registration.module
   connect.with.database
   display.typing.columns.customer
   IF user.OK.button.PRESSED
      IF column.is.filled
         record.new.customer.to.DATABASE
         END IF
      ELSE
         send.back.error.message(“Fill the blanks”)
      END IF
   END BEGIN

2. Login Module for Web based application
   //Login Page
   BEGIN login.module
   connect.with.database
   IF password.of.the.username == password.recorded.in.database
      THEN goto.customer.module
   END IF
   ELSE redisplay.login.page
   END BEGIN

3. Customer module for Web based application
   //Customer module page
   BEGIN customer.module
CASE OF button.PRESSED(bP)
    WHEN bP=Set/Change Information
    BEGIN
        display.typing.columns.customer
        record.customer.to.DATABASE
        IF back.button.PRESSED
            display.user.operations.menu
        END IF
    END BEGIN
END BEGIN

WHEN bP=Set/Meal order
BEGIN
    display.available.meals.customer
    IF selected.ITEM(sI)
        add.item(sI).to.basket
    END IF
    IF unselected.ITEM(sI)
        remove.item(sI).from.basket
    END IF
    IF OK.button.PRESSED
        display.confirmation.menu
        IF OK.button.PRESSED
            record.order.to.DATABASE
            display.warning.cashier
            print.bill.cook
        END IF
    END IF
    IF back.button.PRESSED
        display.user.operations.menu
    END IF
END BEGIN

WHEN bP=Set Reservation
BEGIN
    display.available.tabledates.customer
    IF selected.ITEM(sI)
diplay.confirmation.menu
IF OK.button.PRESSED
record.order.to.DATABASE
display.warning.cashier
END IF
END IF
IF back.button.PRESSED
display.user.operations.menu
END IF
END BEGIN
WHEN bP=View Reservation
BEGIN
select.reservation.for.customer.DATABASE
display.reservation.for.customer
IF selected.ITEM(rS)
   IF DELETE.button.PRESSED
      Delete.rS.from.DATABASE
      Display.warning.cashier
   END IF
END IF
END IF
IF back.button.PRESSED
display.user.operations.menu
END IF
END BEGIN
WHEN bP=View Order
BEGIN
select.order.for.customer.DATABASE
display.order.for.customer
IF selected.ITEM(rS)
   IF DELETE.button.PRESSED
      Display.warning.cashier
      IF Get.result.cashier==OK
         Delete.rS.from.DATABASE
      END IF
      Display.warning.cashier
   END IF
END IF
END IF
IF back.button.PRESSED
    display.user.operations.menu
END IF
IF back.button.PRESSED
    display.user.operations.menu
END IF
END BEGIN
WHEN bP=Log out
BEGIN
    Logout.session.customer
END BEGIN
END CASE OF
END BEGIN

MODULES FOR PALM
1. Login Module for Web based application
//Login menu
BEGIN login.module
    connect.with.database
    IF password.of.the.username == password.recorded.in.database
        THEN goto.waiter.module
    END IF
ELSE redisplay.login.page
END BEGIN

2. Waiter Module for Palm
//Waiter module menu
BEGIN waiter.module
    display.user.operations.menu
    CASE OF button.PRESSED(bP)
        WHEN bP==Choose Table
            BEGIN
display.available.tables.waiter
IF back.button.PRESSED
display.user.operations.menu
END IF
IF selected.table==sT
display.available.meals.waiter
IF back.button.PRESSED
display.available.tables.waiter
END IF
END IF
END IF
IF selected.meals==sM
display.available.customers.waiter
IF back.button.PRESSED
display.available.meals.waiter
END IF
END IF
IF selected.customer==sC
display.confirmation.menu.
IF OK.button.PRESSED
record.(sT,sM,sC).to.DATABASE
print.bill.cook
END IF
END IF
END BEGIN
END CASE OF
END BEGIN

MODULES FOR MAIN APLICATION IN RESTAURANT (MODULE FOR ADMIN/CASHIER)

1. Login Module for Admin and Cashier
   //Login menu
BEGIN login.module
connect.with.database
IF password.of.the.username == password.recorded.in.database
    THEN goto.cashier.module
END IF
ELSE redisplay.login.page
END BEGIN

2. Cashier Module for Admin and Cashier
   
   // Cashier/Admin Module
   BEGIN cashier.module
   display.user.operations.menu
   CASE OF button.PRESSED(bP)
   WHEN bP==Accoting Menu
   BEGIN
   go.to.Accounting.module
   END BEGIN
   WHEN bP==Reservation Menu
   BEGIN
   go.to.Reservation.module
   END BEGIN
   WHEN bP==Table Managment Menu
   BEGIN
   go.to.Table.Managment.module
   END BEGIN
   WHEN bP==Stock Managment Menu
   BEGIN
   go.to.Stock.Managment.module
   END BEGIN
   WHEN bP==Customer Menu
   BEGIN
   go.to.Customer.Managment.module
   END BEGIN
   WHEN bP==Workers Menu
   BEGIN
   go.to.Workers.module
   END BEGIN
WHEN bP==Settings Menu
BEGIN
go.to.Settings.module
END BEGIN
END CASE OF
END BEGIN

3. Accounting Module for Admin and Cashier
   // Accounting Module
BEGIN accounting.module
display.user.operations.menu
CASE OF button.PRESSED(bP)
   WHEN bP=Profit/Loss Menu
   BEGIN
      CASE OF button.PRESSED(bPL)
      WHEN bPL=Income Menu
      BEGIN
         IF button.PRESSED==ViewOnline Income
            display.online.income.menu
         END IF
         IF button.PRESSED==ViewRestaurantIncome
            display.in.restaurant.income.menu
         END IF
         IF button.PRESSED==Select period
            display.income.overperiod
         END IF
         IF button.PRESSED==Print
            print.bill.cashier
         END IF
      END BEGIN
   WHEN bPL=Expenditure Menu
   BEGIN
      IF button.PRESSED==ViewFeedstockExpenditure
         display.feedstock.expenditure.menu
      END IF
END IF
IF button.PRESSED==ViewGeneralExpenditure
    display. general.expenditure.menu
END IF
IF button.PRESSED==ViewStuffExpenditure
    display. stuff.expenditure.menu
END IF
IF button.PRESSED==Select period
    display.income.overperiod
END IF
IF button.PRESSED==Printit
    print.bill.cashier
END IF
END BEGIN

WHEN bP=Choose Box Menu
BEGIN
    CASE OF button.PRESSED(bPC)
    WHEN bPC==View Capital in BOX
        BEGIN
            Display.capital.in.box
        END BEGIN
    END BEGIN
    WHEN bPC==Add Money.to.BOX
        BEGIN
            add.money.to.DATABASE
        END BEGIN
    WHEN bPC==take Money from BOX
        BEGIN
            Take.money.from.DATABASE
        END BEGIN
    END BEGIN
    END CASE OF
END BEGIN

3. Reservation module for Admin and Cashier
   // Reservation Module
BEGIN Reservation.module

display.user.operations.menu

CASE OF button.PRESSED(bP)

WHEN bP=View Reservation
BEGIN
select.reservation.for.customer.DATABASE
display.reservation.for.customer
IF selected.ITEM(rS)
   IF DELETE.button.PRESSED
      Delete.rS.from.DATABASE
   END IF
END IF
END IF
IF back.button.PRESSED
   diplay.user.operations.menu
END IF
END BEGIN
END CASE OF
END BEGIN

3.Reservation module for Admin and Cashier

// Table Managment Module
BEGIN Table.Managment.module

display.user.operations.menu

CASE OF button.PRESSED(bP)
WHEN bP==CloseTable
BEGIN
    set.close.tableid.DATABASE
    print.bill.cashier
END BEGIN
WHEN bP==View order of a Table
BEGIN
    display.order.of.table
    IF Change.button.PRESSED
        display.change.order.menu
    END IF
    IF Cancel.button.PRESSED
        delete.order.of.table.DATABASE
    END IF
END BEGIN
WHEN bP==Transfer order to Another Table
BEGIN
    display.available.table.cashier
    IF selected.Table(sT)
        record.sT.to.DATABASE
        delete.order.table.DATABASE
    END IF
END BEGIN
WHEN bP=Change Table Properties
BEGIN
    get.new.properties(pro)
    record.pro.DATABASE
    IF remove.button.PRESSED
        Remove.table.from.DATABASE
    END IF
END BEGIN
WHEN bP=Add new Table
BEGIN
    Get.new.table.properties(pro)
    Record.pro.DATABASE
END BEGIN
4. Stock Management

// Stock Management Module
BEGIN Stock_Management.module
display_user_operations.menu

CASE OF button.PRESSED(bP)
    WHEN bP=Choose Item Menu
        BEGIN
            display_all_items_in_DATABASE
            IF selected_Item(sI)
                IF remove_button.PRESSED
                    Remove_sI_from_DATABASE
                END IF
            END IF
            IF add_button.PRESSED
                display_blanks_to_cashier
                record_new_item_to_DATABASE
            END IF
        END BEGIN
    WHEN bP==Print Stock Management
        BEGIN
            print_stock_info_cashier
        END BEGIN
END CASE OF
END BEGIN

5. Customer Management

// Customer Management Module
BEGIN Customer_Management.module
display_user_operations.menu

CASE OF button.PRESSED(bP)
    WHEN bP=Choose Customer Menu
        BEGIN
display.all.customers.in.DATABASE
IF selected.customer(sC)
    IF remove.button.PRESSED
        Remove.sC.from.DATABASE
    END IF
END IF
END IF
IF add.button.PRESSED
    display.blanks.to.cashier
    record.new.customer.to.DATABASE
END IF
END BEGIN
WHEN bP==Print Customer
BEGIN
    print.customer.info.cashier
END BEGIN
END CASE OF
END BEGIN

6. Worker Management
   // Worker Management Module
BEGIN worker.Management.module
display.user.operations.menu
CASE OF button.PRESSED(bP)
    WHEN bP=Choose Worker Menu
    BEGIN
        display.all.workers.in.DATABASE
        IF selected.worker(sW)
            IF remove.button.PRESSED
                Remove.sW.from.DATABASE
            END IF
        END IF
    END IF
    IF add.button.PRESSED
        display.blanks.to.cashier
        record.new.worker.to.DATABASE
    END IF
END CASE OF
END BEGIN
END IF
END BEGIN
WHEN bP==Print worker
BEGIN
    print.worker.info.cashier
END BEGIN
END CASE OF
END BEGIN
5. USER INTERFACE

Order GUI of cashier
Inventory GUI of cashier
Meal selection GUI of web user
6. PROJECT SCHEDULE

   a. Gantt Chart for 1st Semester

   The detailed Gantt chart of 1st semester is attached.

   b. Gantt Chart for 2nd Semester

   The detailed Gantt chart of 2nd semester is attached.

7. TESTING

The Restaurant Automation Software will be used by users. The functional requirements for each user were identified in the software requirements phase. The software will be tested to ensure that each functional requirement of the system is met.

The software support modules will need to be tested and debugged to certify proper operation. A standardized Software Testing Form has been created to facilitate the testing procedure. The software form includes areas to verify that data functions/modules function properly. This will consist of creating test data, where the results are known, and comparing them to the output of the program. In addition, this form will test the interactions between the software modules and ensure proper operation. Lastly, the integration test will confirm that the complete software system operates as expected.

<table>
<thead>
<tr>
<th>RAS2005 - Software Testing Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Name: ____________________</td>
</tr>
<tr>
<td>Tester Name: __________________</td>
</tr>
<tr>
<td>Testing Date/Time: ____________</td>
</tr>
<tr>
<td>COMMENTS: ____________________</td>
</tr>
</tbody>
</table>


Test Guideline
Testing procedures will include:

- Unit Testing

  Individual modules of RAS2005 will be tested separately. For each of the components, tests will be conducted until the expected results are obtained.

- Integration Testing

  The components will be tested as a whole to test the orchestration of the modules. Special test data will be created to control the system as a whole. The database module will be tested to ensure proper communication between the interface and the database.

- High-Order Testing

  This test will be performed on the complete, integrated system. Alpha testing will take place. The system will simulate the system with fake inputs. A comparison between the program output and desired values will be discussed.

- Beta Testing

  The tester will be asked to help with the final testing phase. Each beta tester will be given a copy of the software, and the preliminary help files. Beta testers will be expected to submit bug reports and any opinions they may have concerning the software (especially the interface layout.)
8. CONCLUSION

Building restaurant automation system is challenging because there is lots of alternatives in the market that supports most of the functionalities. But as senior programmers this project is our first major project. Under these circumstances, the expectations from our team are really high. Up to now as a group we can say that we work hard for a good design and mostly we achieved. During the design process, we searched new technologies alternative design approaches, thought about what we can add to distinguish our system from others.

We hope that in the second semester, we will implement all proposed functionalities of the system. Maybe one day we drive it to the markets.

Finally by the layered approach the system is gained more flexible and easily maintainable structure. With the help of it, we separate the task into modules easily we hope that coding process will be easier than analysis and design part. We think that layering, Object Oriented Design Principles and Unified Modeling Language shows the power of engineering rather than programming.
### Attachment 1: Detailed Gantt chart of 1st Semester

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>% Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Scope / Project Plan</td>
<td>5.57 days</td>
<td>Fri 10.10.04</td>
<td>Sun 11.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>Determine Project Scope</td>
<td>0.53 days</td>
<td>Fri 10.10.04</td>
<td>Sun 13.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>3</td>
<td>Define Preliminary Proj.</td>
<td>1 day</td>
<td>Mon 11.10.04</td>
<td>Wed 13.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>4</td>
<td>Investigating the topics</td>
<td>0.53 days</td>
<td>Wed 10.10.04</td>
<td>Thu 13.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>5</td>
<td>Project Planning</td>
<td>0.53 days</td>
<td>Fri 10.10.04</td>
<td>Sun 13.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>6</td>
<td>Project Plan Report Cor</td>
<td>0 days</td>
<td>Mon 11.10.04</td>
<td>Mon 11.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>7</td>
<td>ANALYSIS</td>
<td>22 days</td>
<td>Mon 11.10.04</td>
<td>Tue 12.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>8</td>
<td>Define and share res.</td>
<td>0.53 days</td>
<td>Mon 11.10.04</td>
<td>Tue 13.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>9</td>
<td>Draft Preliminary Software</td>
<td>1 day</td>
<td>Tue 12.10.04</td>
<td>Thu 14.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>10</td>
<td>Review Software Specifications</td>
<td>1 day</td>
<td>Thu 14.10.04</td>
<td>Mon 11.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>11</td>
<td>Drawing ER diagrams</td>
<td>0.53 days</td>
<td>Sat 16.10.04</td>
<td>Tue 19.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>12</td>
<td>Develop Estimation</td>
<td>0.67 days</td>
<td>Mon 18.10.04</td>
<td>Wed 20.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>13</td>
<td>Customer Meetings</td>
<td>0.67 days</td>
<td>Wed 20.10.04</td>
<td>Fri 22.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>14</td>
<td>Other software inspections</td>
<td>0.53 days</td>
<td>Fri 22.10.04</td>
<td>Sun 24.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>15</td>
<td>Web page design</td>
<td>1 day</td>
<td>Wed 20.10.04</td>
<td>Mon 25.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>16</td>
<td>Gantt Chart Drawing</td>
<td>1 day</td>
<td>Tue 20.10.04</td>
<td>Thu 26.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>17</td>
<td>Define Functional Res.</td>
<td>0.53 days</td>
<td>Thu 29.10.04</td>
<td>Fri 30.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>18</td>
<td>Define Non-Functional Res.</td>
<td>0.53 days</td>
<td>Fri 29.10.04</td>
<td>Sat 30.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>19</td>
<td>Design of DFD-STD diagram</td>
<td>0.53 days</td>
<td>Fri 29.10.04</td>
<td>Sat 31.10.04</td>
<td>100%</td>
</tr>
<tr>
<td>20</td>
<td>Finalize Analysis Report</td>
<td>0.53 days</td>
<td>Mon 31.10.04</td>
<td>Tue 02.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>21</td>
<td>Analysis complete</td>
<td>0 days</td>
<td>Tue 02.11.04</td>
<td>Tue 02.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>22</td>
<td>INITIAL DESIGN</td>
<td>28 days</td>
<td>Tue 20.11.04</td>
<td>Tue 30.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>23</td>
<td>Review Analysis part</td>
<td>0.53 days</td>
<td>Tue 20.11.04</td>
<td>Wed 03.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>24</td>
<td>Develop functional specific</td>
<td>0.67 days</td>
<td>Thu 24.11.04</td>
<td>Sat 05.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>25</td>
<td>Review functional specific</td>
<td>0 days</td>
<td>Sat 26.11.04</td>
<td>Sun 07.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>26</td>
<td>Object-Oriented Design</td>
<td>0.53 days</td>
<td>Mon 30.11.04</td>
<td>Tue 02.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>27</td>
<td>Use-case Analysis</td>
<td>0.53 days</td>
<td>Sat 29.11.04</td>
<td>Tue 02.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>28</td>
<td>Activity Diagram designs</td>
<td>1 day</td>
<td>Fri 12.11.04</td>
<td>Tue 16.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>29</td>
<td>Class Diagram Creation</td>
<td>1 day</td>
<td>Sat 13.11.04</td>
<td>Wed 17.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>ID</td>
<td>Task Name</td>
<td>Duration</td>
<td>Start</td>
<td>Finish</td>
<td>% Complete</td>
</tr>
<tr>
<td>----</td>
<td>----------------------------------</td>
<td>----------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>30</td>
<td>Sequence Diagram design</td>
<td>1 day</td>
<td>Sat 11.11.04</td>
<td>Wed 17.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>31</td>
<td>Class Diagram</td>
<td>0.33 days</td>
<td>Mon 13.11.04</td>
<td>Tue 15.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>32</td>
<td>Collaboration Diagram desc</td>
<td>1 day</td>
<td>Thu 18.11.04</td>
<td>Tue 23.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>33</td>
<td>structure chart design</td>
<td>1.33 days</td>
<td>Thu 18.11.04</td>
<td>Wed 24.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>34</td>
<td>Finalizing Class Diagram</td>
<td>1 day</td>
<td>Wed 17.11.04</td>
<td>Sat 20.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>35</td>
<td>Finalize Initial Design Report</td>
<td>0.67 days</td>
<td>Thu 25.11.04</td>
<td>Sun 28.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>36</td>
<td>Initial Design Complete</td>
<td>0 days</td>
<td>Tue 30.11.04</td>
<td>Tue 30.11.04</td>
<td>100%</td>
</tr>
<tr>
<td>37</td>
<td>TEAM PRESENTATION</td>
<td>7 days</td>
<td>Tue 07.12.04</td>
<td>Tue 14.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>38</td>
<td>Decide topics of presentation</td>
<td>0.33 days</td>
<td>Tue 07.12.04</td>
<td>Wed 08.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>39</td>
<td>Reviewing documentation</td>
<td>0.67 days</td>
<td>Wed 08.12.04</td>
<td>Thu 09.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>40</td>
<td>Primitive GUI design</td>
<td>0.67 days</td>
<td>Wed 08.12.04</td>
<td>Fri 10.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>41</td>
<td>Preparing the program</td>
<td>0 days</td>
<td>Sat 11.12.04</td>
<td>Sun 12.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>42</td>
<td>Preparing Presentation</td>
<td>0 days</td>
<td>Sun 12.12.04</td>
<td>Mon 13.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>43</td>
<td>Making Presentation</td>
<td>0 days</td>
<td>Tue 14.12.04</td>
<td>Tue 14.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>44</td>
<td>DETAILED DESIGN</td>
<td>40 days</td>
<td>Wed 01.01.04</td>
<td>Mon 04.04.05</td>
<td>100%</td>
</tr>
<tr>
<td>45</td>
<td>Reviewing Project scope a</td>
<td>2 days</td>
<td>Wed 01.01.04</td>
<td>Wed 03.01.04</td>
<td>100%</td>
</tr>
<tr>
<td>46</td>
<td>Database Design</td>
<td>3.33 days</td>
<td>Wed 01.01.04</td>
<td>Wed 15.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>47</td>
<td>Revealing Architectural dec</td>
<td>0 days</td>
<td>Wed 01.01.04</td>
<td>Fri 24.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>48</td>
<td>Preparing revisited Use-C</td>
<td>2 days</td>
<td>Sat 11.12.04</td>
<td>Mon 20.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>49</td>
<td>Preparing Data Dictionary</td>
<td>2 days</td>
<td>Wed 15.12.04</td>
<td>Wed 22.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>50</td>
<td>Preparing Sequence diag</td>
<td>2 days</td>
<td>Wed 15.12.04</td>
<td>Thu 23.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>51</td>
<td>Detailed Class diagrams test</td>
<td>4 days</td>
<td>Thu 16.12.04</td>
<td>Mon 03.01.05</td>
<td>100%</td>
</tr>
<tr>
<td>52</td>
<td>Revealing related SQL code</td>
<td>2 days</td>
<td>Sat 18.12.04</td>
<td>Tue 28.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>53</td>
<td>Preparing Activity diagram</td>
<td>2 days</td>
<td>Tue 21.12.04</td>
<td>Tue 28.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>54</td>
<td>Preparing Collaboration Di</td>
<td>1 day</td>
<td>Fri 24.12.04</td>
<td>Tue 28.12.04</td>
<td>100%</td>
</tr>
<tr>
<td>55</td>
<td>Extended GUI Designs</td>
<td>1.67 days</td>
<td>Sat 28.12.04</td>
<td>Tue 04.01.05</td>
<td>100%</td>
</tr>
<tr>
<td>56</td>
<td>Revealing Project Schedule</td>
<td>1.25 days</td>
<td>Sat 01.01.05</td>
<td>Fri 07.01.05</td>
<td>100%</td>
</tr>
<tr>
<td>57</td>
<td>Finalizing Detailed Design</td>
<td>0.67 days</td>
<td>Thu 06.01.05</td>
<td>Sat 08.01.05</td>
<td>100%</td>
</tr>
<tr>
<td>58</td>
<td>Detailed Design Complete</td>
<td>0 days</td>
<td>Mon 10.01.05</td>
<td>Mon 10.01.05</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Legend:**
- Task
- Progress
- Milestone
- Summary
- Rolled Up Task
- Rolled Up Milestone
- Rolled Up Progress
- External Tasks
- Project Summary
- Group By Summary

**Project: DETDES**

**Date:** Mon 10.01.05

**Page 2**
<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>% Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>PROTOTYPE DEMO</td>
<td>16 days</td>
<td>Sun 02.01.05</td>
<td>Tue 14.01.05</td>
<td>36%</td>
</tr>
<tr>
<td>60</td>
<td>Reviewing web and spec</td>
<td>1 day</td>
<td>Sun 02.01.05</td>
<td>Tue 04.01.05</td>
<td>100%</td>
</tr>
<tr>
<td>61</td>
<td>Revisited GUI or web user</td>
<td>2 days</td>
<td>Fri 07.01.05</td>
<td>Fri 14.01.05</td>
<td>25%</td>
</tr>
<tr>
<td>62</td>
<td>Revisiting GUI for vanier</td>
<td>2 days</td>
<td>Fri 07.01.05</td>
<td>Fri 14.01.05</td>
<td>25%</td>
</tr>
<tr>
<td>63</td>
<td>Finalizing the Prototype De</td>
<td>0.53 days</td>
<td>Fri 14.01.05</td>
<td>Mon 17.01.05</td>
<td>0%</td>
</tr>
<tr>
<td>64</td>
<td>Making the prototype dents</td>
<td>0 days</td>
<td>Tue 16.01.05</td>
<td>Tue 18.01.05</td>
<td>0%</td>
</tr>
<tr>
<td>ID</td>
<td>Task Name</td>
<td>Duration</td>
<td>Start</td>
<td>Finish</td>
<td>% Complete</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------</td>
<td>----------</td>
<td>-------------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td>31</td>
<td>Incorporate user documentation feedback</td>
<td>1 day?</td>
<td>Tue 17.05.05</td>
<td>Wed 18.05.05</td>
<td>0%</td>
</tr>
<tr>
<td>32</td>
<td>Documentation complete</td>
<td>0 days</td>
<td>Thu 19.05.05</td>
<td>Thu 20.05.05</td>
<td>0%</td>
</tr>
<tr>
<td>33</td>
<td>Pilot</td>
<td>16 days</td>
<td>Mon 23.05.05</td>
<td>Wed 25.05.05</td>
<td>0%</td>
</tr>
<tr>
<td>34</td>
<td>Identify test group</td>
<td>3 days</td>
<td>Wed 22.05.05</td>
<td>Thu 23.05.05</td>
<td>0%</td>
</tr>
<tr>
<td>35</td>
<td>Develop software delivery mechanism</td>
<td>7 days</td>
<td>Wed 22.05.05</td>
<td>Wed 28.05.05</td>
<td>0%</td>
</tr>
</tbody>
</table>