1.1. PROJECT SUMMARY

The Restaurant-POS System facilitates the Restaurants/café to handle/track their customer's orders through a computerized system. Customers can place their orders from the tablets and the manager can track their orders from the browser in the computer on his desk. The tablet application is developed in android, and the manager side application of tracking the orders in developed using Php. The android app is developed in two modes – The Customers Mode and The Waiter Mode. The Customer Mode application helps the customers to place the orders directly from the tablets placed on the respective table. It also helps the customers to browse through the internet after providing a valid username and password. The waiter mode helps the waiter to place the order on behalf of a customer. The waiter mode also helps the waiter to view the orders placed from all the tables at a particular time.

The manager at the counter can view the orders coming from different tables from the Live Screen. He can approve/disapprove the orders, Set the order status to served, and generates the bill on customers request. The browser application also helps the manager to add new menu categories, subcategories, and items to the Menu, modify/delete an existing category, subcategory, item. He can also add/modify/delete waiter records, user records, tables and sections records etc. He can set the theme of the android applications.

1.2. PURPOSE

The main purpose of the system is to facilitate the restaurant with various operations which help them in their ordering system. The operations like manage menu items, waiters, users, placing order, generating bills and invoices etc. can all be performed from a central place. The application provides an interactive, user friendly and attractive menu which helps the customer to place their order, which is handled efficiently by the application.

1.3. SCOPE

Functionalities

- Adding and updating menu items to display them on the android app.
- Categorize menu items into categories and subcategories.
- Adding and updating waiters and tables information.
- Adding and updating user information.
- Adding and updating device(tablets) information.
- Assigning device(tablets) to tables/waiters.
- Create live screen layouts as per the physical layout of the tables in the restaurant.
- Change theme of the android application as per the manager's choice.
- Place orders from android app.
- Browse internet from the android application.
- Make a "call the waiter" request from the android app.
- Make "checkout" request from the android app.
- Approve orders, generate bills/invoices for the orders.

1.4. OBJECTIVE

- Provide candidates with field experience in the arena of software development using OOAD and PHP and Android technology including analyzing, designing, coding, testing & implementing an application software system. Provide exposure to corporate working environment and develop confident and employable individuals for the software industry.

1.5. TECNOLOGY AND LITERATURE REVIEW

- **Tools and Technologies used:**
- PHP
- MySQL
- Android SDK
- HTML, JavaScript, jQuery.
- Microsoft Word, PowerPoint, PictureManager.

PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embed.

PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

PHP is widely used for Server-side scripting. This is the most traditional and main target field for PHP. You need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. You need to run the web server, with a connected PHP installation. You can access the PHP program output with a web browser, viewing the PHP page through the server. All these can run on your home machine if you are just experimenting with PHP programming. See the installation instructions section for more information.

PHP can be used on all major operating systems, including Linux, many Unix variants, Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for

most of the web servers today. This includes Apache, IIS, and many others. And this includes any web server that can utilize the FastCGI PHP binary, like lighttpd and nginx.

So with PHP, you have the freedom of choosing an operating system and a web server. Furthermore, you also have the choice of using procedural programming or object oriented programming (OOP), or a mixture of them both.ded into HTML.

MySQL

MySQL is the world's most widely used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

Android SDK

Android applications are written in the Java programming language. The Android SDK tools compile the code—along with any data and resource files—into an Android package, an archive file with an .apk suffix. All the code in a single .apk file is considered to be one application and is the file that Android-powered devices use to install the application.

The Android operating system is a multi-user Linux system in which each application is a different user.

Each process has its own virtual machine (VM), so an application's code runs in isolation from other applications.

By default, every application runs in its own Linux process. Android starts the process when any of the application's components need to be executed, then shuts down the process when it's no longer needed or when the system must recover memory for other applications.

In this way, the Android system implements the principle of least privilege. That is, each application, by default, has access only to the components that it requires to do its work and no more. This creates a very secure environment in which an application cannot access parts of the system for which it is not given permission.

However, there are ways for an application to share data with other applications and for an application to access system services:

An application can request permission to access device data such as the user's contacts, SMS messages, the mountable storage (SD card), camera, Bluetooth, and more. All application permissions must be granted by the user at install time.

That covers the basics regarding how an Android application exists within the system.

The core framework components that define your application.

- The manifest file in which you declare components and required device features for your application.

- Resources that are separate from the application code and allow your application to gracefully optimize its behavior for a variety of device configurations.

HTML, Javascript, jQuery

HTML stands for Hyper Text Markup Language.

HTML is a language for describing web pages. HTML stands for Hyper Text Markup Language HTML is a markup language. A markup language is a set of markup tags

The tags describe document content. HTML documents contain HTML tags and plain text. HTML documents are also called web pages.

JavaScript is a Scripting Language

A scripting language is a lightweight programming language. JavaScript is programming code that can be inserted into HTML pages. JavaScript inserted into HTML pages, can be executed by all modern web browsers.

jQuery is a lightweight, "write less, do more", JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

Microsoft Word, PowerPoint, PictureManager

Microsoft Word 2010 is a word-processing program, designed to help you create professional-quality documents. With the finest document-formatting tools, Word helps you organize and write your documents more efficiently. Word also includes powerful editing and revising tools so that you can collaborate with others easily.

Microsoft PowerPoint 2010 gives you more ways to create and share dynamic presentations with your audience than ever before

With Microsoft Office Picture Manager you can manage, edit, share, and view your pictures from where you store them on your computer. Use the picture editing tools to crop, expand, or copy and paste..

2.1. PROJECT PLANNING

2.1.1. PROJECT DEVELOPMENT APPROACH AND JUSTIFICATION (SDLC MODEL)

The Software development Model used is "Iterative Enhance Model"**.** First we will see the various phases of software development life cycle.

Stages In Software Development Life Cycle (SDLC):

***Requirement Analysis & Definition*:** All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

***System & Software Design*:** Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

***Implementation & Unit Testing*:** On receiving system design documents, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality; this is referred to as Unit Testing. Unit testing mainly verifies if the modules/units meet their specifications.

***Integration & System Testing*:** As specified above, the system is first divided in units which are developed and tested for their functionalities. These units are integrated into a complete system during Integration phase and tested to check if all modules/units

coordinate between each other and the system as a whole behaves as per the specifications. After successfully testing the software, it is delivered to the customer.

***Operations & Maintenance*:** This phase of "The Waterfall Model" is virtually never ending phase (Very long). Generally, problems with the system developed (which are not found during the development life cycle) come up after its practical use starts, so the issues related to the system are solved after deployment of the system. Not all the problems come in picture directly but they arise time to time and needs to be solved; hence this process is referred as Maintenance.

2.1.2. PROJECT PLAN

It deals with the order in which important planning activities may be undertaken. Some of the essential activities are:

- Estimation of some basic attributes of the project such as cost, duration, effort
- Scheduling manpower and other resources

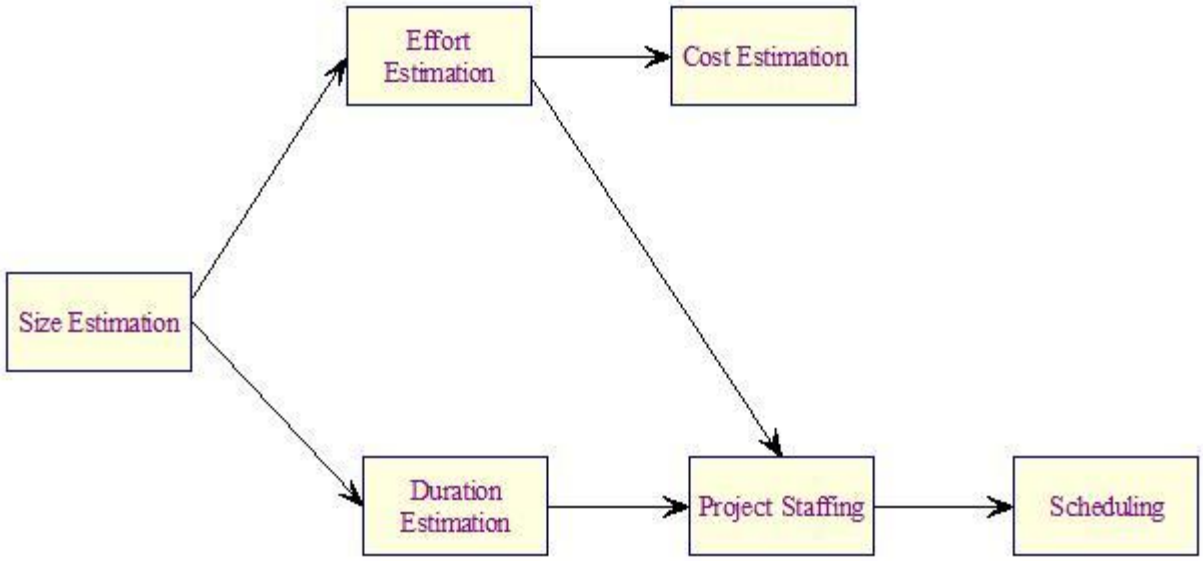

Figure 2.3. Project Plan.

2.1.3. ROLES AND RESPONSIBILITIES

For better development of a project it is better to perform requirement analysis and system design in a group and partition the implementation into modules and each team member should work on different task. Here in our project we have used “divide and conquer”. We had taken individual functionalities and performed implementation and unit testing in it. Later, performed module testing and integrated testing in group.

2.1.4. GROUP DEPENDENCIES

The team structure depends on the management style of the organization, the number of people in the team, their skill levels and the problem difficulty.

The best team structure depends on the Management style. Team Structures are of three types:

1. Democratic Decentralized, 2. Controlled Decentralized, 3. Controlled Centralized.

The team structure chosen by us was **Democratic Decentralized**. The reasons for adapting this structure are, this is the first time we are working on an industry project. None of us has undertaken the job of a team leader before so other two team structures are not possible as well as feasible. A similar skill set is possessed by all of us. We have faith and confidence in each other’s’ decisions and respect each other’s’ advice. Being a team of only three members, delegation of tasks as in a controlled team structures was not feasible.

2.1.5 WORK BREAK DOWN STRUCTURE

Roles & Responsibilities	**Members**
Requirement gathering and analysis	Akshat Gandhi, Kartik Patel, Sukrut Modak
Database handling	Akshat Gandhi, Kartik Patel
Interface designing	Akshat Gandhi, Sukrut Modak
Implementation	Akshat Gandhi, Kartik Patel, Sukrut Modak
Documentation	Akshat Gandhi, Kartik Patel, Sukrut Modak
Testing	Sukrut Modak, Kartik Patel

2.2 FEASIBLILTY STUDY

It is an estimate that is made up of whether the users of the system need to be satisfied using the current software and hardware technologies. The study will decide if the proposed system will be cost-effective from a business point of view and if it can be developed within given existing budgetary constraints. A feasibility study should be relatively cheap and quick. So, the result should inform the decision of whether to go ahead or not.

The feasibility study of the System to be developed and need to be conducted before initiating the project development.

Results of feasibility study of the system are as follows:

2.2.1 Operational feasibility

Operational Feasibility checks whether the system is operationally feasible or not. It will examine the various operational parameters for the same. We will check operational feasibility of the given system by answering certain questions which are listed below.

Does system provide adequate throughput and response time?

Response time is very critical for the ordering system. Computerizing the ordering system will minimize the request and response time of order processing. This system will provide adequate throughput and adequate response time to make the system operationally feasible, make faster operation of different modules of the system.

Does the system provide end-user and manager with timely, patient, accurate and usefully formatted information?

The system will provide end-user with timely information. The system will provide information such as categories, subcategories, items etc. The system will provide this information on time. It will also maintain accurately states of order.

Does the system make maximum use of available resources including people, time, flows of forms, minimum processing delays and the like?

System will operate in two modes: Table Mode and Waiter Mode. Both modes will make maximum use of available resources. In table mode, system will make use of customers only. So no unnecessary use of waiters. Thus order time will be reduced. In waiter mode waiter can place orders from system, and those orders will directly send to manager/receptionist. Thus human errors and processing time can be reduced.

Does system offer adequate service level and capacity to reduce the cost of business? Or increase profit of business?

System will provide functionality of digital menu, reducing cost of printing/reprinting menu cards when minor changes in menu happen. By providing waiter mode application system can increase number of customers served in particular time period as discussed earlier.

2.2.2 Technical Feasibility

Technical Feasibility refers to the ability of process to take advantages of current state of technology with pursuing further improvement. Technical feasibility will be addressed by following explanation.

Is the proposed technology or solution practical?

The proposed technology is practically quite feasible. It is mature enough to tackle the problems and give necessary services.

2.2.3 Schedule Feasibility

As per our application, we have already divided tasks among all members. Generally we work in “democratic decentralized” manner for completing the given work including analysis, designing and coding.

At the time of completion, we worked in a centralized manner for integrating the given work. So it includes both centralized while integration and decentralizes while individual.

2.2.4 Economic Feasibility

Economic feasibility has been done in advanced for estimating the cost of the proposed system. Cost-benefit analysis of system is done. Here all the costs scheduled to the system i.e. direct and indirect costs are calculated and it is checked to see whether it is affordable or not. This checks the economic feasibility. Our system will have affordable costs both direct and indirect costs. So the system will be economically feasible. Our system will need tools like Android SDK, PHP and MySQL. All these software are Open Source and freely available to use for developing the system. That's how our system will reduce time and also money for the given functionalities required for developing the system. So the system will be economically feasible given the conditions and functionalities completed in given amount of time.

.

2.3 PROJECT SCHEDULE

Software Project Scheduling Principles:

- **Compartmentalization** - the product and process must be decomposed into a manageable number of activities and tasks.
- **Interdependency** - tasks that can be completed in parallel must be separated from those that must be completed serially.
- **Time allocation** - every task has start and completion dates that take the task interdependencies into account.
- **Effort validation** - project manager must ensure that on any given day there are enough staff members assigned to completed the tasks within the time estimated in the project plan.
- **Defined Responsibilities** - every scheduled task needs to be assigned to a specific team member.
- **Defined outcomes** - every task in the schedule needs to have a defined outcome (usually a work product or deliverable).
- **Defined milestones** - a milestone is accomplished when one or more work products from an engineering task have passed quality review.

Timeline (Gantt) chart enables software planners to determine what tasks will be need to be conducted at a given point in time (based on estimates for effort, start time, and duration for each task).

ID	Task Name	Start	Finish	Duration	2012			
					Jan	Feb	Mar	Apr
1	Requirement Collection and Analysis	1/1/2013	31/1/2013	4w				
2	Planning	1/2/2013	14/2/2013	2w				
3	Design	15/2/2013	5/3/2013	3w				
4	Implementation	6/3/2013	3/4/2013	4w				
5	Testing	25/3/2013	16/4/2013	2w				

Figure 2.4 Gantt chart

2.4 RISK MANAGEMENT

There are different categories of risk. Risk that are to be analyzed like project risks , business risks , technical risks , known risks , predictable risks and unpredictable risks. *Project risks* identify potential budgetary, schedule, personal that includes staff and organization, resources, customers and requirement problems and they impact on software projects. *Technical risks* identify potential design; implementation, interface, verification, maintenance problems, specification ambiguity, technical uncertainness and technical obsolesce. *Business risks* threaten the viability of the software to be built. *Known risks* are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed. *Predictable risks* are extrapolated from past project experience. *Unpredictable risks* are extremely difficult to identify in advance.

2.4.1. Risk Identification

1. Technology: While system is building / compiling and end-user request to access and manipulate information then system get re-configure and user get configuration error.

2. Hardware: System runs on some network. So all hardware related problems like connection failure and server down problems and power failure must be managed effectively by our team.

3. Software: System is depending on the database. There are other external libraries and tools like Android api, and client tools etc. requires regular maintenance so that they do not fail and we can prevent data lost.

4. People: As System is a database and network driven which contains lots of concepts and connection mechanisms that are used, it require understanding all features and functionalities, also how it get implemented and affected RMI architecture. So we need to interact to Technical Manager and other employees to understand tool and concepts effectively. So we try to make this risk as small as by writing tutorials about tools and techniques we will be using.

6. Schedule: Each team member should respect the deadlines that the team has decided. When a team member knows that he will be unable to respect a given deadline, he should inform the other team members as soon as possible (preferably more than two days before the deadline), so that the team can find a way to solve the problem.

7. Delay Due to illness: This is one natural risk that can occur at any moment during project development. Illness of any person working for project in future can cause delay to the project completion.

2.4.2 Risk Analysis

Probability of the risks might be assessed as very low (<10%), low (10-25%), moderated (25-50%), high (50-75%), or very high (>75%).

Effects of the risk might be assessed as catastrophic, serious, tolerable or insignificant.

Risk	Probability	Effect
Technology	Low	Catastrophic
Hardware	Moderated	Critical
Software	Moderated	Critical
People	Very Low	Catastrophic
Schedule	Low	Critical
Delay due to Illness	Very Low	Tolerable

Table 2.3. Risk Analysis

2.4.3 RISK ABATEMENT PROCEDURE / PLANNING

Risk	Strategy
Hardware	We will give the detail of hardware that works fine with this software. And we instruct our customer to use that hardware for successful running of this software. Minimization strategy
Software	We instruct our software users to check the hardware resources first for success working of software. Avoidance strategy
People	Three of us, the team members keep track of the project progress and the work done by each other. This can enable us to work individually and finish the project in case of any illness or other unknown problem prevents one of us, the members from working on the project. Avoidance strategy
Schedule	For the purpose of development for which we are not so sure about the time allocation. We have given some extra time to solve the development problem like coding problems. This will cause the feasible schedule. Minimization strategy

Table 2.4. Risk Planning

3.1 STUDY OF CURRENT SYSTEM

Currently, the bulk operations of ordering system performed in restaurant are done manually by making a waiter approach the table to note down the order for the customer on a paper. The waiter needs to physically approach the manager/receptionist so that he can approve the order. While checkout manager/receptionist needs to enter the order data in computer to generate invoice. The system needs automation as well as ease in these operations. Hereby are the details of certain operations which have been proposed to be automated and eased.

3.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM

Managing Menu Items

This is the basic operation of a restaurant which is to be more user friendly. Currently, the manager needs to change the menu cards, every time there is a change in any menu item details. This results in a lot of expense in changing the menu card which may be cut-off and utilized in some other productive works. Also it takes lot of time to perform these operations. Our system removes these limitations and allows a fast and easy way to perform the above mentioned.

Tracking Orders

This is the most complex operation which must be handled accurately and efficiently. Currently orders are taken by the waiters on a piece of paper, which are difficult to handle and modify. It also requires a lot of man resource. Generation of bills/invoices are done by feeding the ordered item details from the piece of paper into a computer. There are chances of human mistakes. In the proposed system orders are tracked by the system itself, which eliminates human mistakes and order processing time is also minimized. On checkout invoices are automatically generated.

Maintaining Past Records

Since the orders were taken on a piece of paper searching past records is very difficult job. With the proposed system order and invoice data are stored in a database which makes retrieval of past records easier.

3.3 REQUIREMENTS OF NEW SYSTEM

3.3.1. USER REQUIREMENTS

- Order processing must be efficient and accurate
- Menu management must be easy to maintain
- Maintaining the log of users and orders
- System should be flexible.
- System should be able to handle large amount of data.
- Less frequency of failure.
- System must be User friendly.
- Data Security.

3.3.2. HARDWARE AND SOFTWARE REQUIREMENTS

- WAMP Server
- Compatible Browser
- A Computer with minimum core 2 Duo Processor, 2 Gb Ram, and NIC card.
- A wireless router
- 7 inch Android Tablets

3.4 FUNCTIONAL MODULES OF THE CURRENT SYSTEM

The sytem is divided into various modules to simply working and understanding of the system. The modules of the system are shown in below figure :

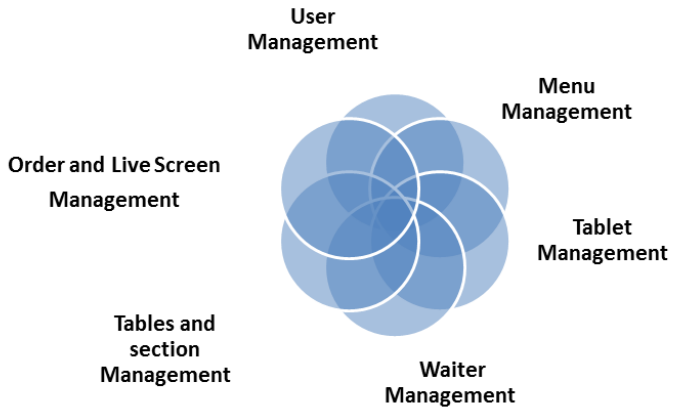

Figure 3.1 Functional Modules

1. Menu Management

Menu Management module allows basic add, update, delete functionality for items that restaurant/café is serving. System can select which items are in menu today. For adding a menu item one needs to provide information like name of item, description of item, images for that item, ingredients , preparation process, preparation duration, notes etc..

Menu management module also allows items to be categorized. In adding/updating an item system will ask for that item's parent category. Supplying category information will put that item into that category. Same way add/update of category / subcategory can be done.

Now customer side application can fetch menu from server. After fetching menu from server application will display interactive menu. Here customer can select category from the list of category. Selecting category will display subcategories for the selected category. After customer selects a subcategory, items for that selected subcategory will be displayed. Now customer can select items and then order those selected items. After

placing order customer will not be able to update or cancel the order. Modifications should be done before placing the order.

2. Waiter Management

Waiter Management module allows basic add, update, delete functionality for waiter details that restaurant/café recruited. For adding/updating a waiter detail one needs to provide information like name of name, age, notes etc.

3. Table and Section Management

Table and Section Management module relates to the actual layout of tables and sections (First Floor, Second Floor, etc.) of that restaurant/café. Section management will allow user to set sections, update sections and delete sections. For a particular section user will be able to add/update/delete tables. On screen a table will be shown by a square. User can add a Table (Square) in particular section. Then move that Table (Square) in the selected area as par the actual layout of that table in restaurant/café. Now after moving it to the designated area, table position will be saved. User can also assign names to tables.

4. Device Management

Device Management module will allow user to add/update/delete device details like name, assigned to, notes etc. This module will allow user to set a device to particular table or waiter. Adding/ Updating device details will allow user to select table or waiter that has been assigned to this device.

5. User Management

User management module will allow user to manage users for Internet Access Module. As per government rules, if you are allowing some user to access Internet, you need to have his identity proof in case things go wrong. So our system will take names of documents and their unique number as necessary detail while adding/updating user detail. After registering with system, user will be given ID & Password to access Internet module on customer side application. User can sign out of Internet module, when he wants to sign out. On checking out of restaurant/café user will automatically be signed out of this module.

6. Order & Live Screen Management

Order management refers to managing orders those have been placed from customer side application. Managing orders includes keeping track of order (via status), approve/disapprove order, generate invoice for a particular order. Keeping track of order includes order states like created, seen, approved/disapproved, served, checked out, and billed. Manager/Receptionist will be able to approve/disapprove order as human errors may occur while placing order. On this page there will be two buttons for approving/disapproving orders. And there will be one check box for setting item status if it is served or not. If Manager/Receptionist checks that checkbox, item status will be changed to served.

Server side order management is achieved through LIVE SCREEN. This screen will display orders on the tables we discussed in Table/Section Management. If an order is received from Tabel1 then square related to that Table will show new order received state. New order state will be shown by blinking Red color on that particular square. Now Manager/Receptionist will click on that new arrived order. After clicking, status of the order will be updated to seen and Manager/Receptionist will approve/disapprove order. And color of that square will be changed to Yellow. After order is prepared and served status of order will be updated to served, and color of the square related to table from which order has arrived will be changed to Green. If any new order placed from the same table, after the first order has been served, the status of that order will be changed to created. At end when user checks out, status of order will be updated to checked out and table color which was related to the order/user will be changed to blinking Yellow. Now Manager/Receptionist will generate invoice for order. Invoice page takes input customer name, customer address for generating invoice. Status of the order will be updated to billed. Color of the table related to order will be converted back to Gray.

This module also includes a facility to call waiter right from the application. If user has any query to be answered he can press call waiter button in customer side application. This will display table name in Red color on LIVE SCREEN of server side application. Manager/Receptionist can handle waiter request manually, and send one of waiters to that table.

3.5. FEATURES OF NEW SYSTEM

1. Live Screen Management. System allows handling all orders right from the live screen.

2. Internet Access Module, allows registered users to use internet right from the application.

3. Call Waiter functionality, allows customers to call waiter right from the application.

4. Checkout functionality, allow customers to checkout from the application.

5. Invoice Generation, will generate report of bill details.

4.1 USE CASE DIAGRAM

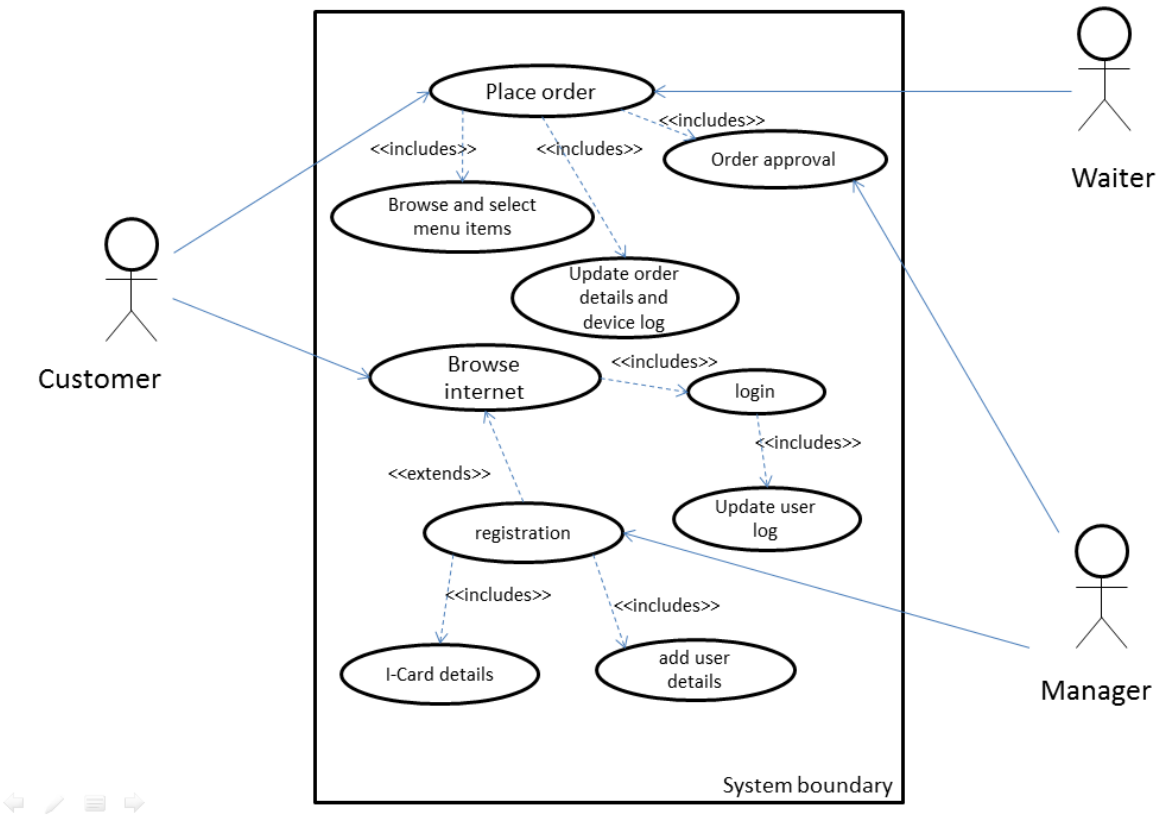

Figure 4.1 Use Case Diagram (1)

Use Case - 1

Use case	**Place order**
Description	This use case will allow user to place the order
Primary Actors	Customer, Waiter
Triggering Condition	The user will opt for placing an order
Pre-condition	The user will select the order item list
Post-condition	A new order will be added to the system
Basic Flow	• User opts for adding a order • System will allow user to add menu items to the order • User clicks on the order button • The order is added • A notification is shows by making the ordered items green • Use case ends

Use Case - 2

Use case	**Browse Internet**
Description	This use case will allow user to Access the internet from the application
Primary Actors	Customer
Triggering Condition	The user will opt for Accessing the internet
Pre-condition	The user is authenticated and his/her credentials are verified
Post-condition	User can access the internet
Basic Flow	• User opts for accessing the internet • System will allow user to his/her credentials • User clicks on the login button • The user is allowed to access the internet • Use case ends
Alternate Flow	**Alternate Flow 1: Invalid Credentials** • Error message will be displayed when incorrect data is entered • The user will fill up proper data for authentication • The flow resumes with step 3 of basic flow **Alternate Flow 2: Quit to access internet** • User selects to go back to menu • Use case ends **Alternate Flow 3: User is not registered** • User will go to reception desk and ask for registration • User details will be added by manager/receptionist after verifying the user's documents • The flow resumes with step 2 of basic flow

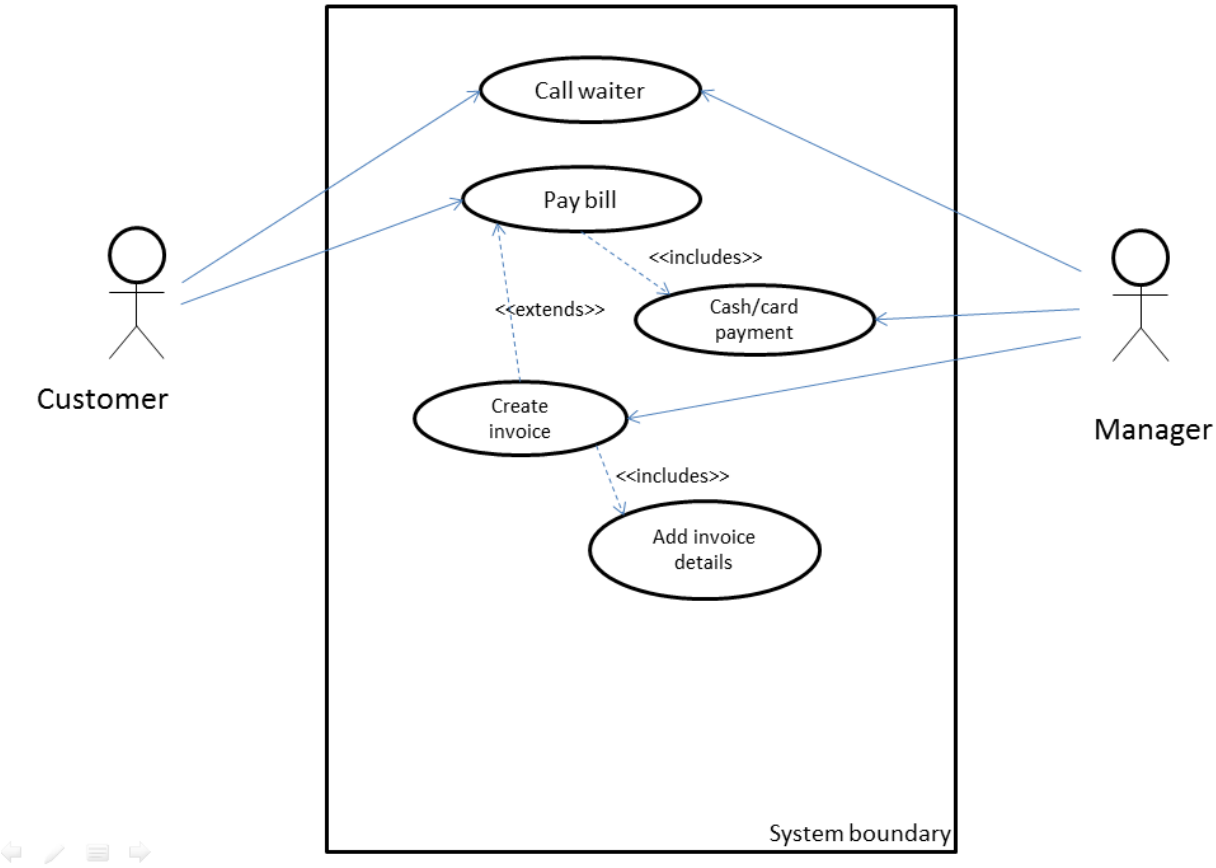

Figure 4.2 Use Case Diagram (2)

Use Case - 3

Use case	**Call Waiter**
Description	This use case will allow customer to call waiter for any help/query
Primary Actors	Customer
Triggering Condition	The user will opt for calling a waiter
Pre-condition	The user should be customer
Post-condition	A notification is sent to manager/receptionist about waiter request
Basic Flow	• User opts for calling a waiter • System will allow user to call a waiter • User clicks on the call waiter button • The request is sent to manager • A notification “waiter is coming” is displayed to user • Use case ends

Use Case - 4

Use case	**Pay Bill**
Description	This use case will allow user to request a bill
Primary Actors	Customer, Waiter
Triggering Condition	The user will opt for calling a waiter
Pre-condition	The user should be customer
Post-condition	A notification is sent to manager/receptionist about waiter request
Basic Flow	• User opts for calling a waiter • System will allow user to call a waiter • User clicks on the call waiter button • The request is sent to manager • A notification “waiter is coming” is displayed to user • Use case ends

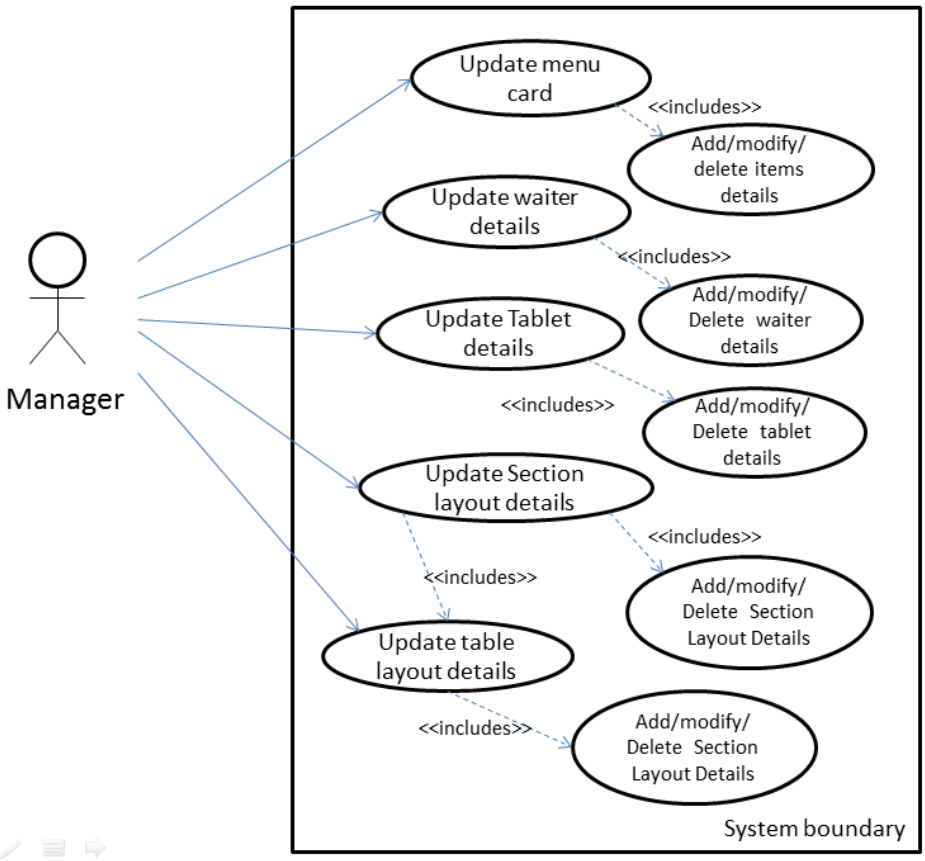

Figure 4.3 Use Case Diagram (3)

Use Case 3

Use case	**Update Menu Card**
Description	This use case will allow manager to add/update/delete the menu items
Primary Actors	Manger
Triggering Condition	The user will opt for Menu management
Pre-condition	The user is authenticated and his/her credentials are verified
Post-condition	A Menu card details will be updated
Basic Flow	• User opts for Menu management • User will choose either add/edit/delete option • System will allow user to enter details • User selects to save the details • Menu item details will be updated • A notification “Details have been saved successfully” is displayed to user

	• Use case ends
Alternate Flows	**Alternate Flow 1: Invalid Entry** • Error message will be displayed to user when incorrect data is entered • Then user will fill up proper data • The flow resumes with step 4 of basic flow **Alternate Flow 2: Quit Option** • The user opts for clear/cancel option • Use case ends

4.2 CLASS DIAGRAM

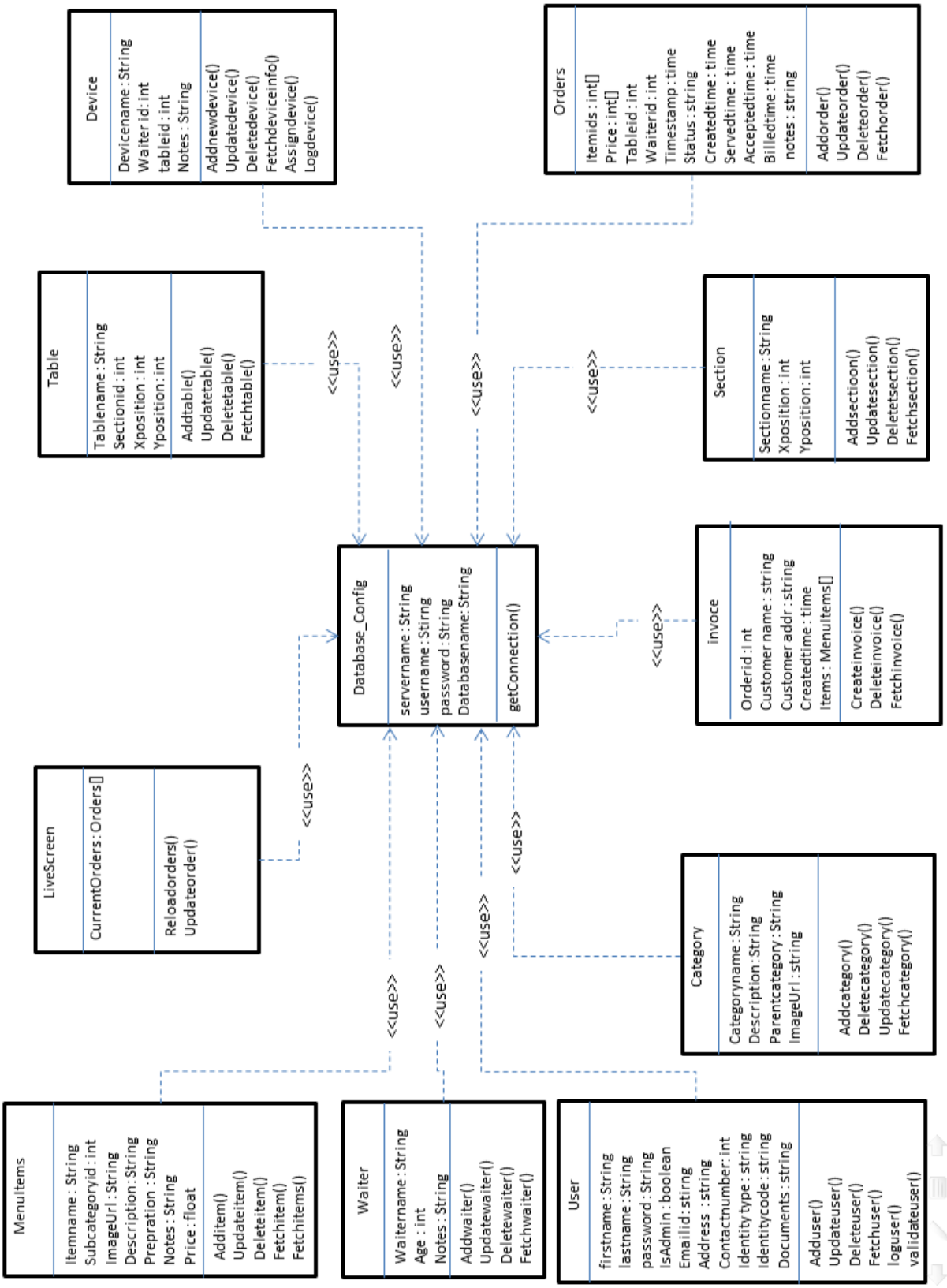

Figure 4.4 Class Diagram (1)

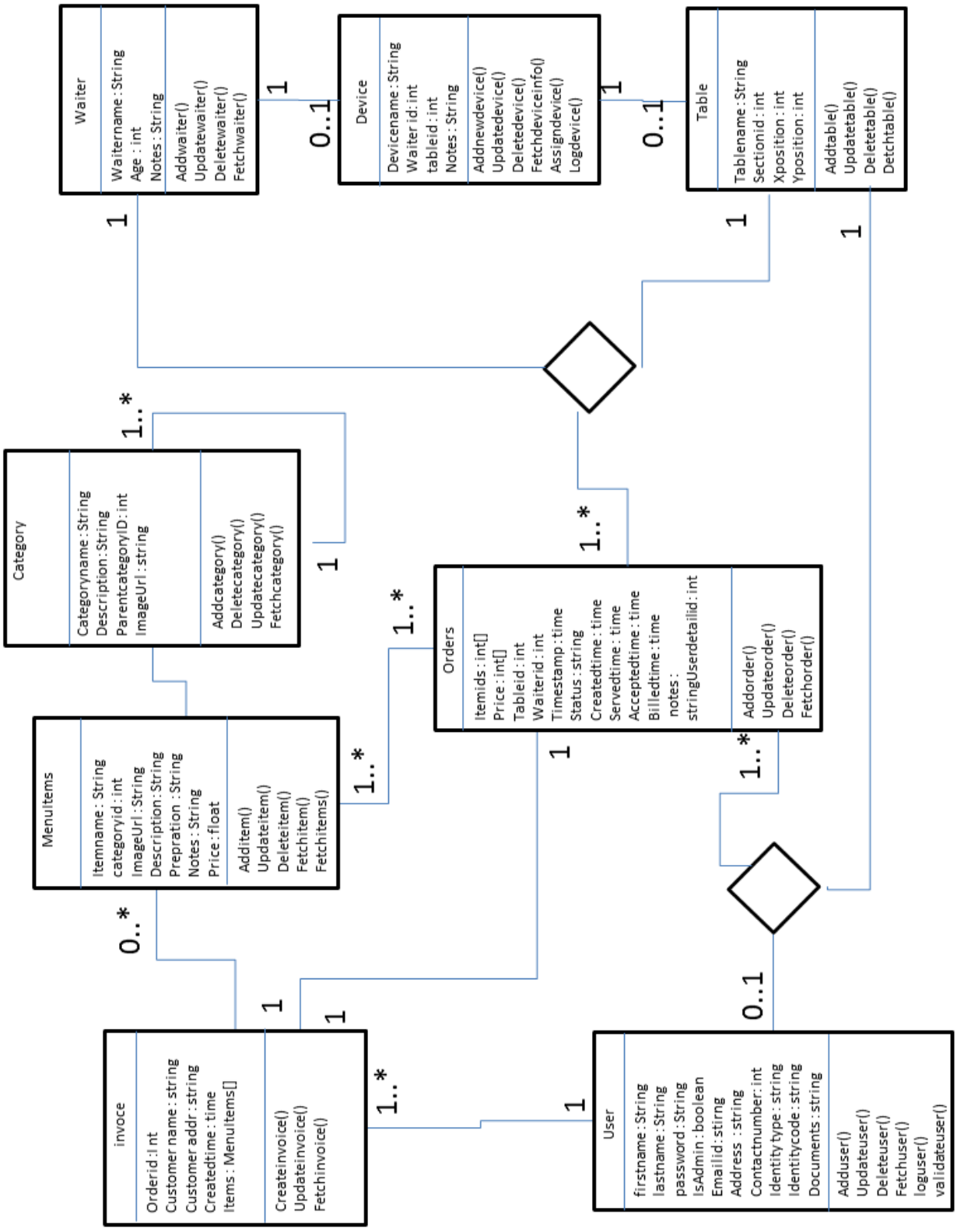

Figure 4.5 Class Diagram (2)

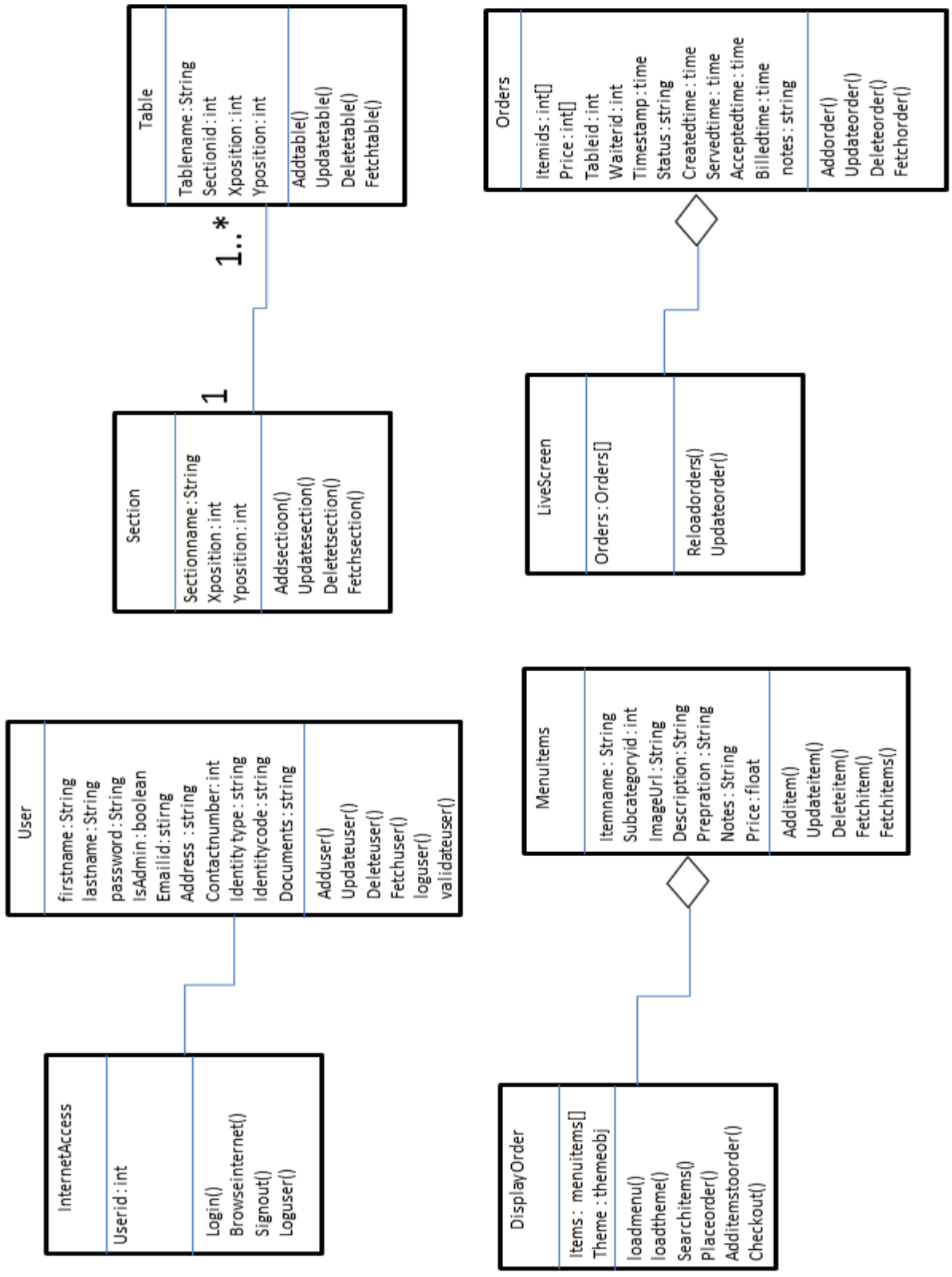

Figure 4.6 Class Diagram (3)

4.3 ACTIVITY DIAGRAM

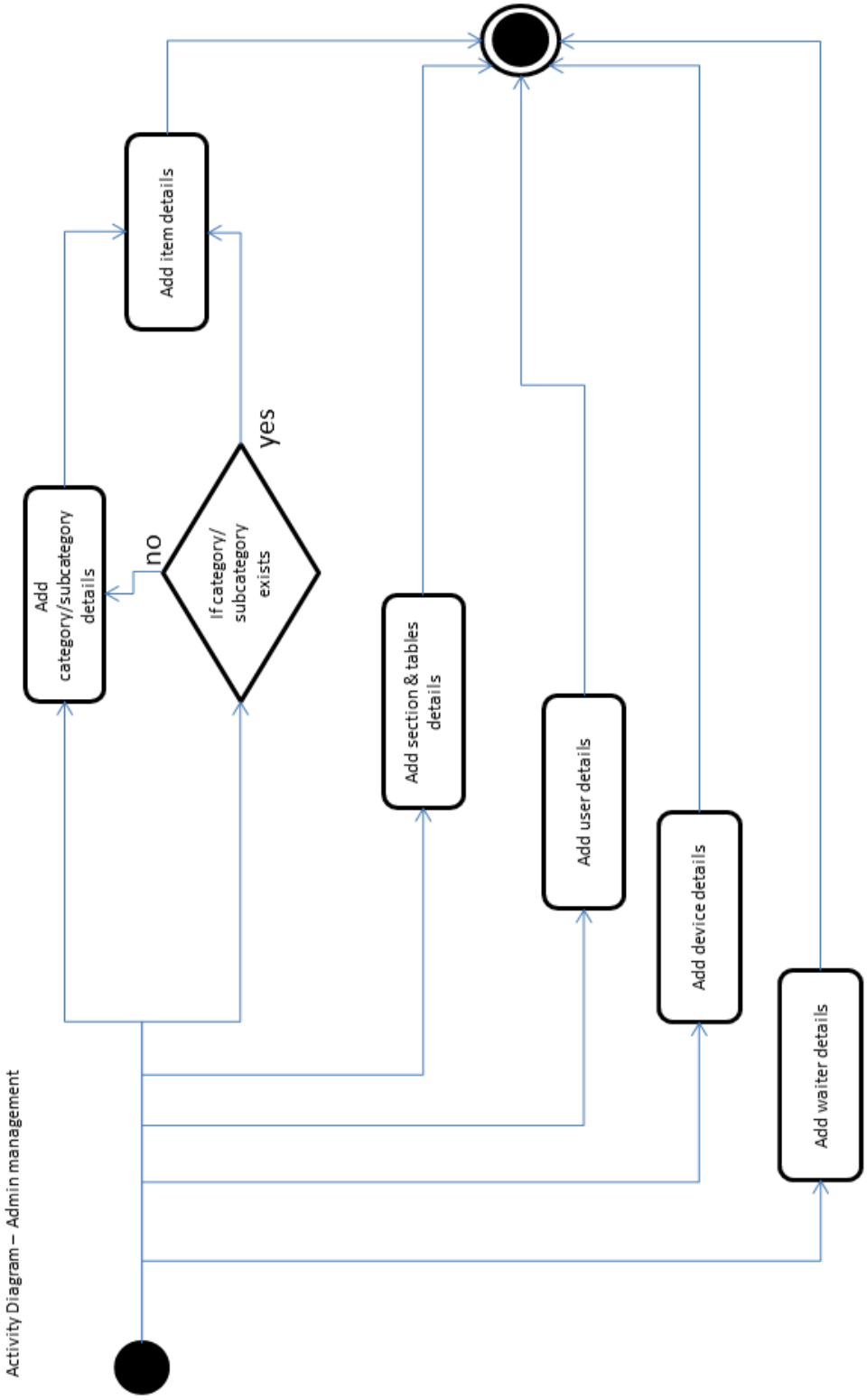

Figure 4.7 Activity Diagram (1)

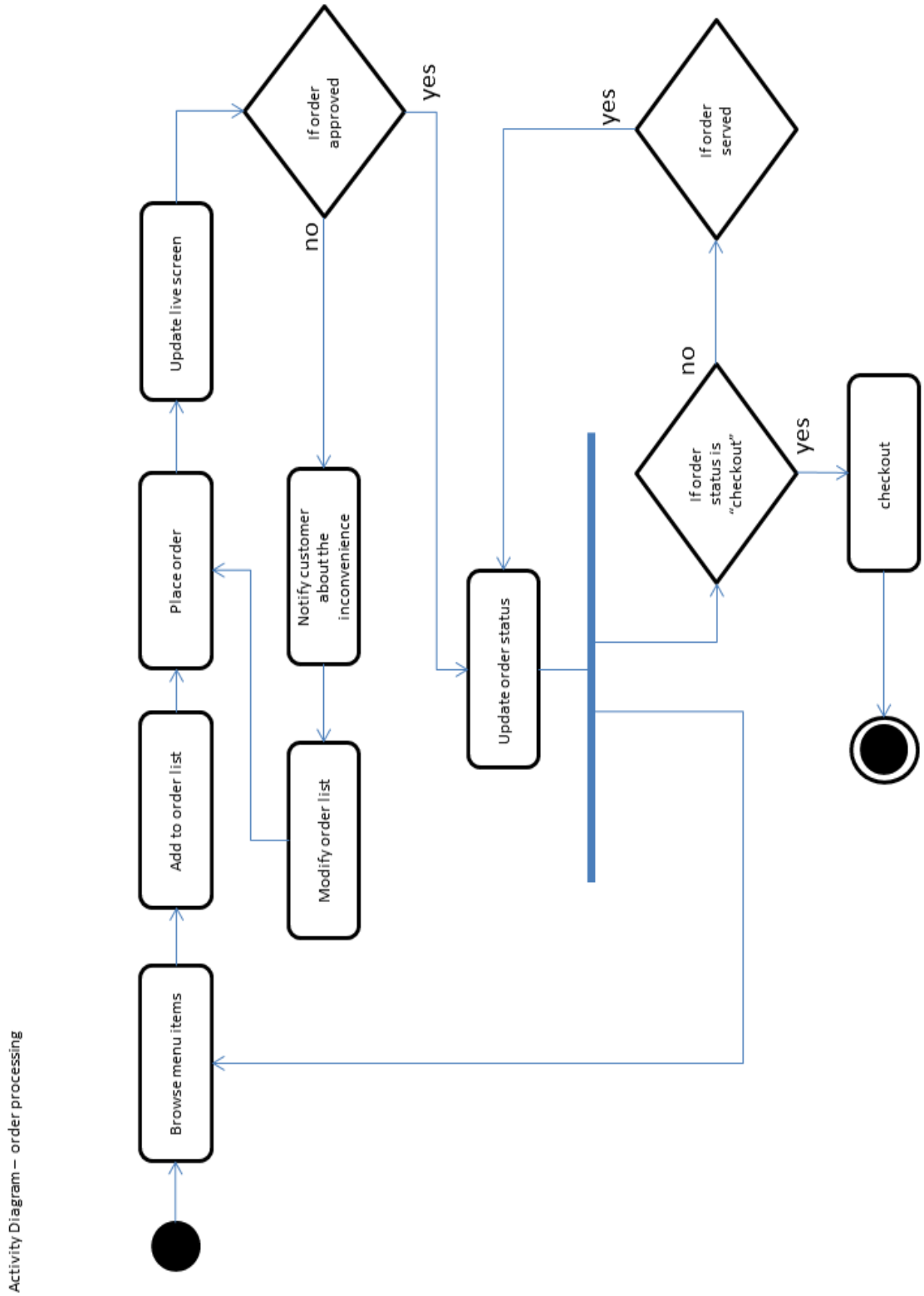

Figure 4.8 Activity Diagram (2)

4.4 SEQUENCE DIAGRAM

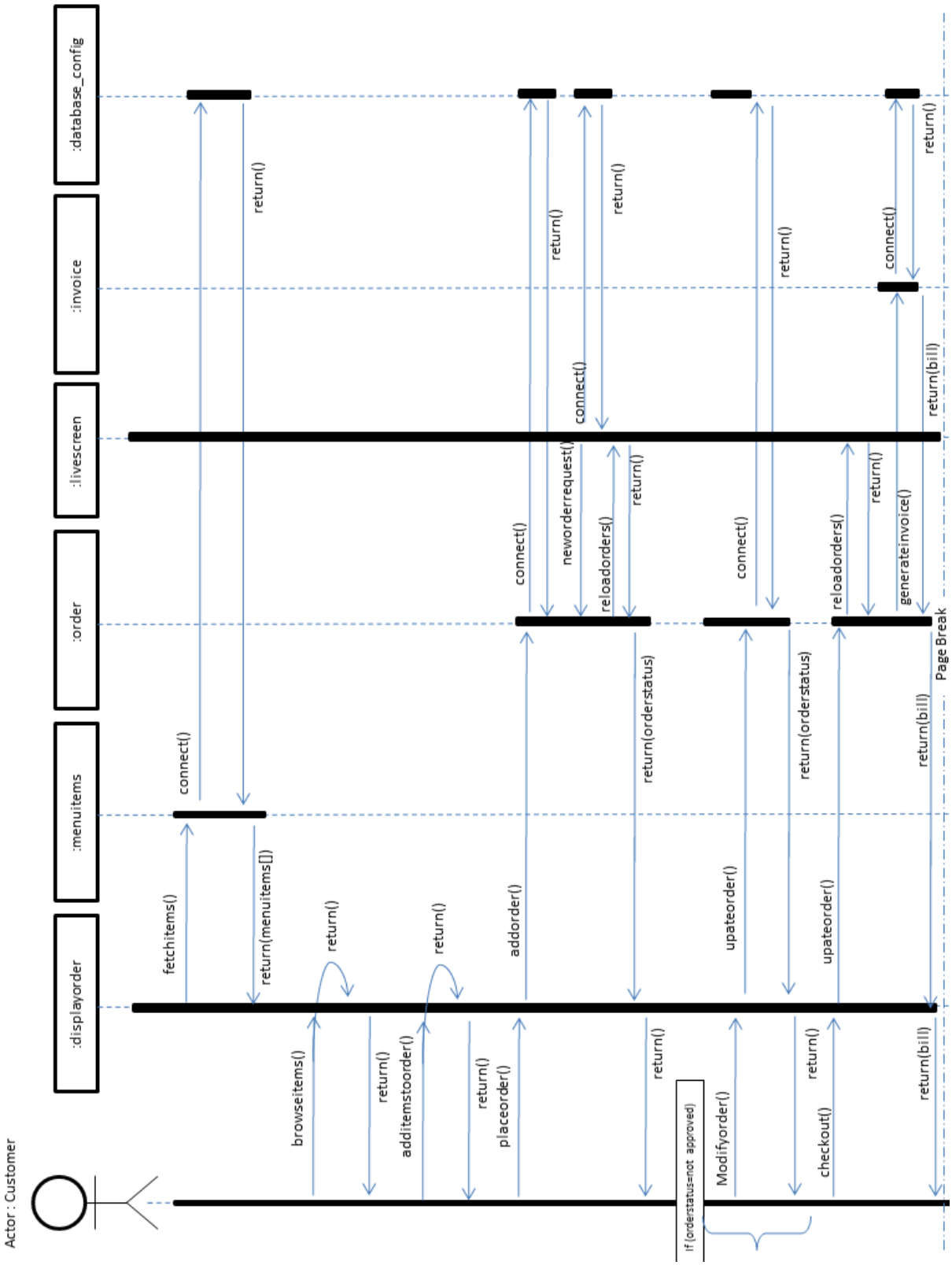

Figure 4.9 Sequence Diagram (1)

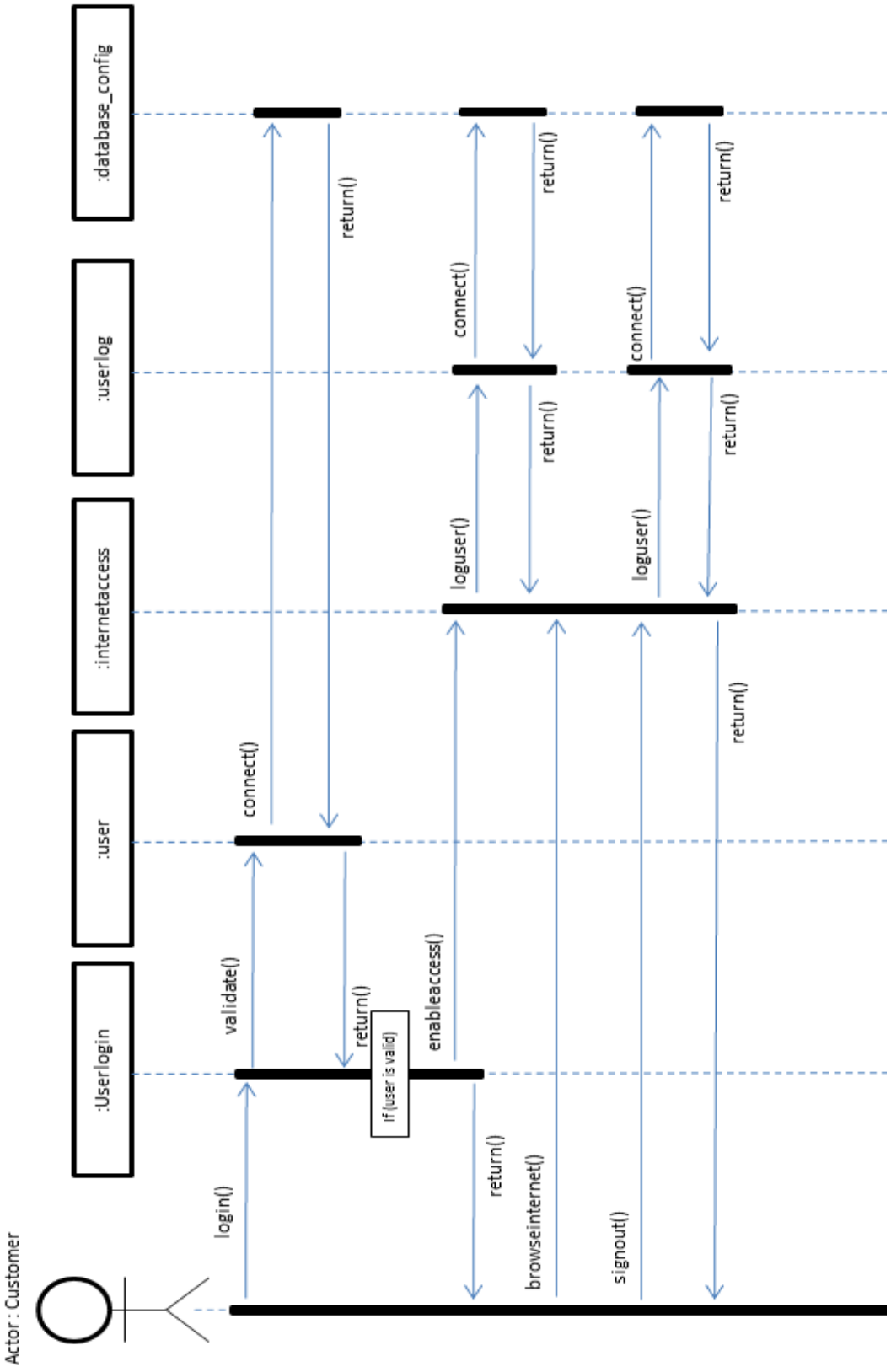

Figure 4.10 Sequence Diagram (2)

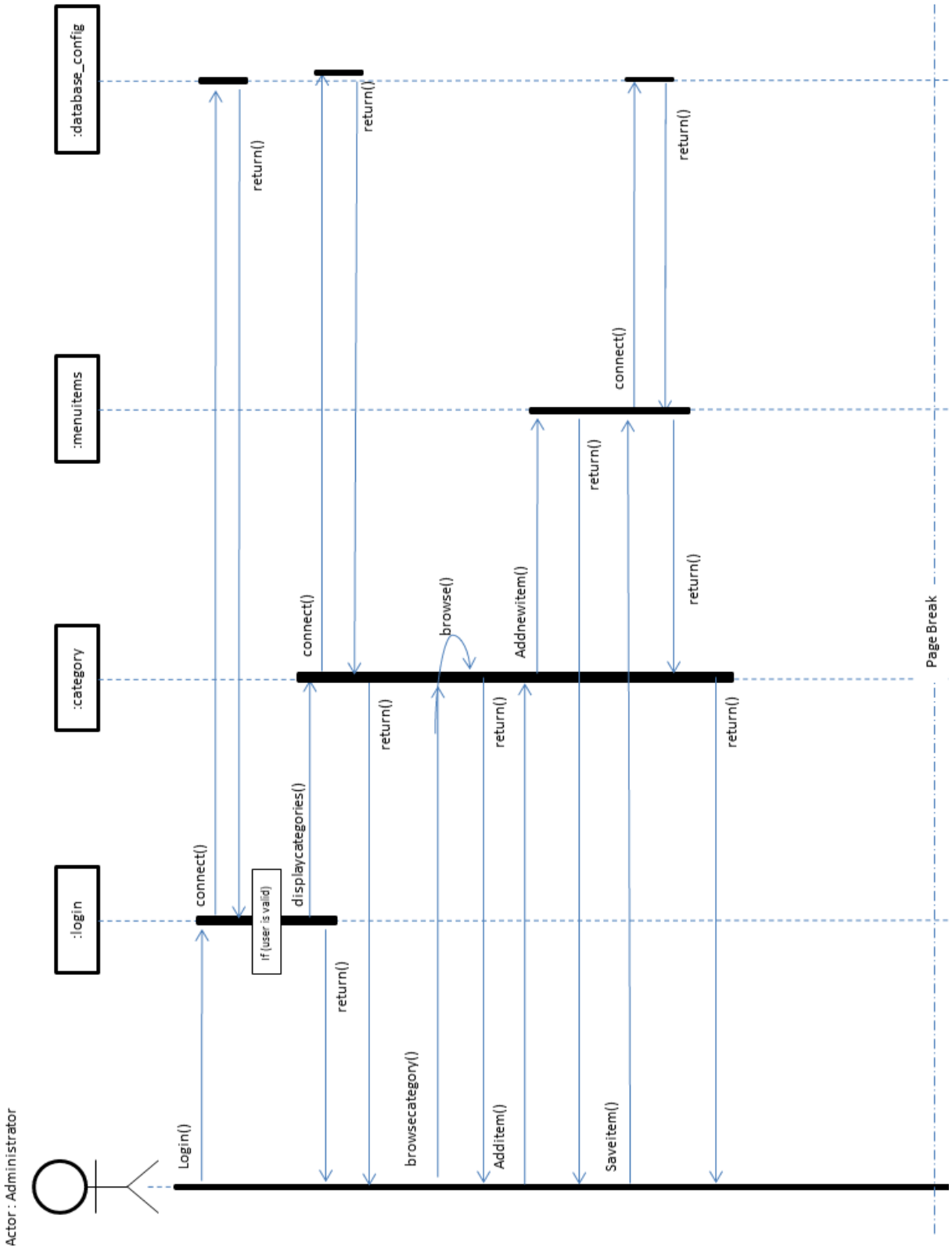

Figure 4.11 Sequence Diagram (3)

4.5 STATE DIAGRAM

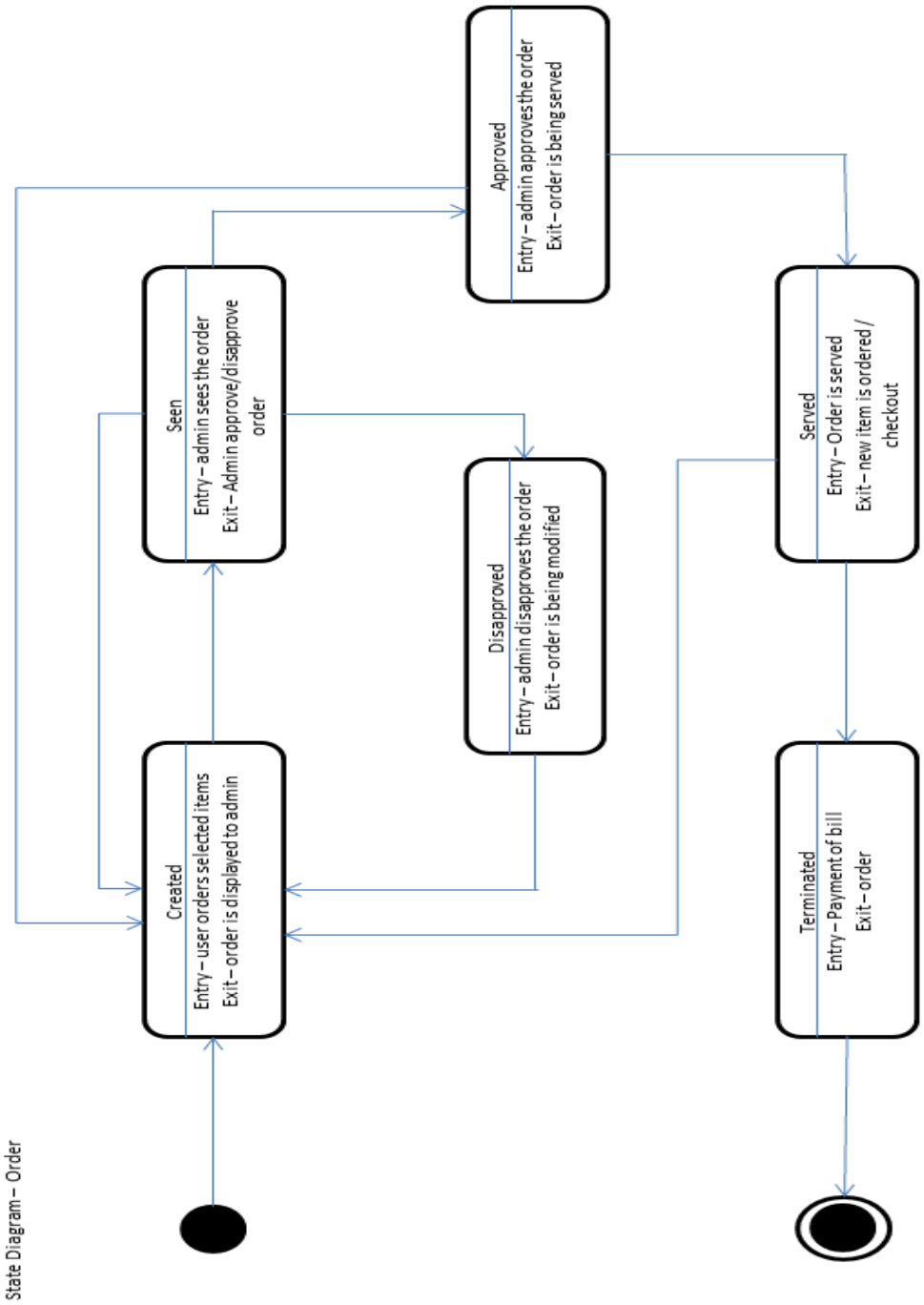

Figure 4.10.1 State Diagram

4.6 E-R DIAGRAM

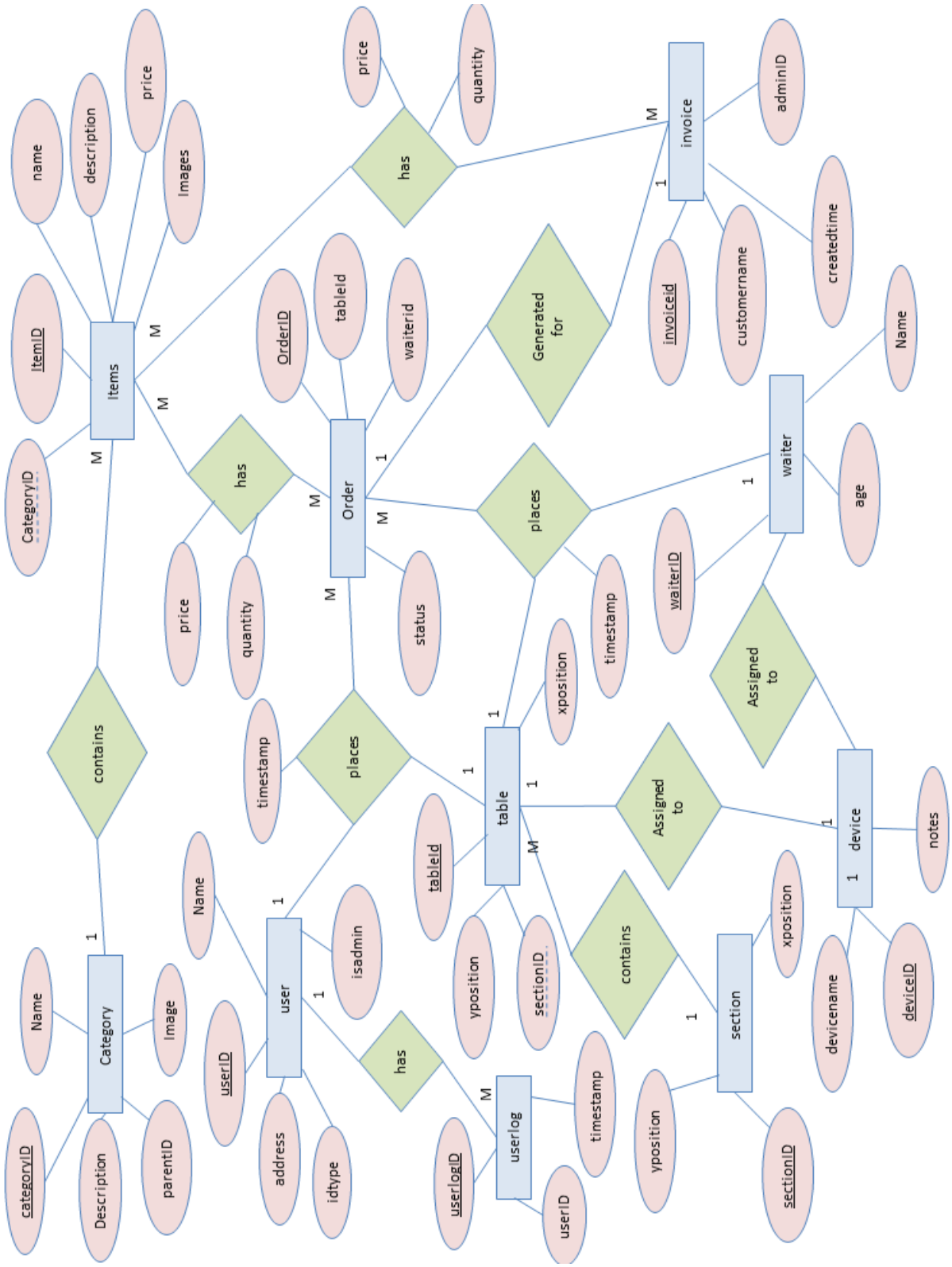

Figure 4.11 E-R Diagram

4.7 DATA DICTIONARY

1. Categories

Field	Type	Null	Key	Default	Extra
id	int(3)	NO	PRI	NULL	
name	varchar(200)	YES		NULL	
title	varchar(200)	YES		NULL	
parent_id	varchar(3)	YES		NULL	
image	varchar(50)	YES		NULL	

2. Device

Field	Type	Null	Key	Default	Extra
DeviceID	int(3)	NO	PRI	NULL	
DeviceName	varchar(50)	NO		NULL	
WaiterId	int(3)	YES	MUL	NULL	
TableId	int(3)	YES	MUL	NULL	
Notes	varchar(500)	YES		NULL	

3. Devicelog

Field	Type	Null	Key	Default	Extra
DeviceLogId	int(3)	NO	PRI	NULL	
DeviceId	int(3)	YES	MUL	NULL	
TimeStamp	timestamp	YES		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
WaiterId	int(3)	YES		NULL	
TableId	int(3)	YES		NULL	

4. Invoice

Field	Type	Null	Key	Default	Extra
InvoiceId	int(3)	NO	PRI	NULL	
OrderId	int(3)	YES	MUL	NULL	
CustomerName	varchar(50)	YES		NULL	
CustomerAddress	varchar(50)	YES		NULL	
CreatedTime	datetime	YES		NULL	
AdminUserDetailId	int(3)	YES		NULL	

5. Invoiceitem

Field	Type	Null	Key	Default	Extra
InvoiceItemId	int(3)	NO	PRI	NULL	
InvoiceId	int(3)	YES	MUL	NULL	
ItemName	varchar(50)	YES		NULL	
Quantity	int(3)	YES		NULL	
Amount	decimal(6,2)	YES		NULL	

6. Items

Field	Type	Null	Key	Default	Extra
item_id	int(3)	NO	PRI	NULL	
item_name	varchar(100)	YES		NULL	
Category_id	int(3)	YES	MUL	NULL	
description	varchar(1000)	YES		NULL	
ingredients	varchar(1100)	YES		NULL	
preperation	varchar(2000)	YES		NULL	
price	float	YES		NULL	
images	varchar(1000)	YES		NULL	
is_active	tinyint(1)	YES		NULL	
approx_duration	varchar(100)	YES		NULL	
notes	varchar(500)	YES		NULL	

7. Order

Field	Type	Null	Key	Default	Extra
OrderId	int(3)	NO	PRI	NULL	
UserDetailId	int(3)	YES	MUL	NULL	
TableId	int(3)	YES	MUL	NULL	
WaiterId	int(3)	YES	MUL	NULL	
CreatedTime	datetime	YES		NULL	
AcceptedTime	datetime	YES		NULL	
ServedTime	datetime	YES		NULL	
BilledTime	datetime	YES		NULL	
Status	int(3)	YES		NULL	
AdminUserDetailId	int(3)	YES		NULL	
Notes	varchar(50)	YES		NULL	

8. Orderitem

Field	Type	Null	Key	Default	Extra
OrderItemId	int(3)	NO	PRI	NULL	
OrderId	int(3)	YES	MUL	NULL	
Item_Id	int(3)	YES	MUL	NULL	
Quantity	int(3)	YES		NULL	
Price	decimal(6,2)	YES		NULL	
isServed	bit(1)	YES		NULL	

9. Restable

Field	Type	Null	Key	Default	Extra
TableId	int(3)	NO	PRI	NULL	
TableName	varchar(50)	YES		NULL	
PosX	int(3)	YES		NULL	
PosY	int(3)	YES		NULL	
Width	int(3)	YES		NULL	
Height	int(3)	YES		NULL	
SectionId	int(3)	YES	MUL	NULL	

10. Restaurantdetail

Field	Type	Null	Key	Default	Extra
RestaurantDetailId	int(3)	NO	PRI	NULL	
RestaurantName	varchar(50)	YES		NULL	
Address	varchar(100)	YES		NULL	
PhoneNo	decimal(12,0)	YES		NULL	
FaxNo	decimal(12,0)	YES		NULL	

11. Section

Field	Type	Null	Key	Default	Extra
SectionId	int(3)	NO	PRI	NULL	
SectionName	varchar(50)	YES		NULL	
PosX	int(3)	YES		NULL	
PosY	int(3)	YES		NULL	
Width	int(3)	YES		NULL	
Height	int(3)	YES		NULL	

12. Themes

Field	Type	Null	Key	Default	Extra
themeid	int(1)	NO		NULL	
themename	varchar(20)	NO		NULL	
day	varchar(3)	NO		NULL	
isset	bit(1)	YES		NULL	

13. Userdetail

Field	Type	Null	Key	Default	Extra
UserDetailId	int(3)	NO	PRI	NULL	
Fname	varchar(50)	YES		NULL	
Lname	varchar(50)	YES		NULL	
Email	varchar(50)	YES		NULL	
Password	varchar(15)	YES		NULL	
Address	varchar(50)	YES		NULL	
Phone	bigint(10)	YES		NULL	
IdentityType	varchar(50)	YES		NULL	
IdentityCode	varchar(50)	YES		NULL	
Documents	varchar(50)	YES		NULL	
IsAdmin	tinyint(1)	YES		NULL	

14. Userlog

Field	Type	Null	Key	Default	Extra
themeid	int(1)	NO		NULL	
themename	varchar(20)	NO		NULL	
day	varchar(3)	NO		NULL	
isset	bit(1)	YES		NULL	

15. Waiter

Field	Type	Null	Key	Default	Extra
UserDetailId	int(3)	NO	PRI	NULL	
Fname	varchar(50)	YES		NULL	
Lname	varchar(50)	YES		NULL	
Email	varchar(50)	YES		NULL	
Password	varchar(15)	YES		NULL	
Address	varchar(50)	YES		NULL	
Phone	bigint(10)	YES		NULL	
IdentityType	varchar(50)	YES		NULL	
IdentityCode	varchar(50)	YES		NULL	
Documents	varchar(50)	YES		NULL	
IsAdmin	tinyint(1)	YES		NULL	

5.1. SYSTEM ARCHITECTURE

The system architecture is basically a PHP-Android based client-server model of computing. Below is the brief description of the client-server model of our application.

The application is a distributed application that partitions tasks between the providers of a resource or service, called servers (PHP Application), and service requesters, called clients (Android Tablets). They communicate over a wireless network on separate hardware.

The illustration below depicts a distributed application that uses the places an order from the android client. The android client requests the PHP server to fetch and display the menu items. On placing an order from the android client, the request is send to the PHP server, which updates the managers/receptionist live screen. Further the request is approved/disapproved by the receptionist/manager and it updates the client's order status.

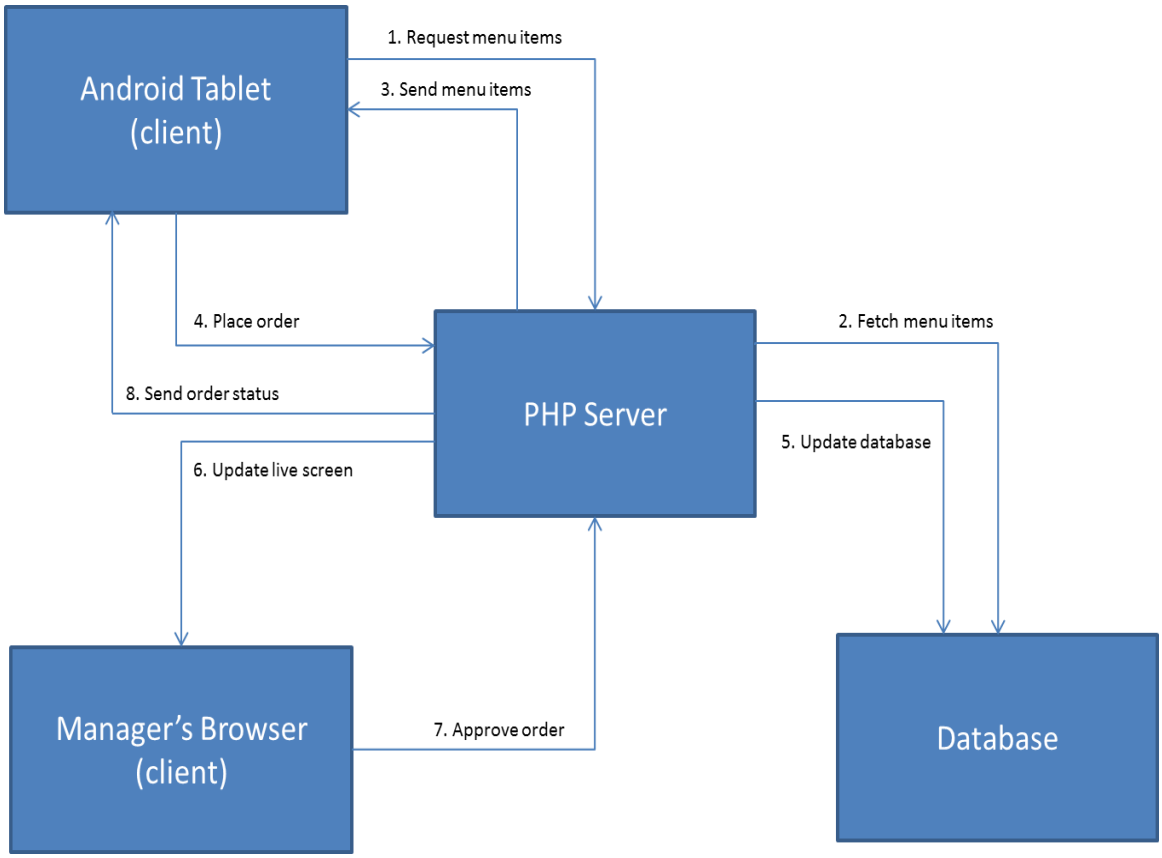

Fig 5.1.1 Distributed application of placing order

5.2. CODING STANDARDS

1. Naming Convention

Use **full English** descriptors that accurately describe the variable/field/class/interface
For example, use names like **menuItems**, **orderedItems**, or **MenuRequest**.

Use **terminology applicable** to the **domain**
If the users of the system refer to their clients as Customer, then use the term Customer for the class, not client.

Use **mixed case** to make names readable

Use **abbreviations sparingly**, but if you do so then use then intelligently and document it
For example, to use a short form for the word "number", choose one of **nbr**, **no** or **num**.

Avoid long names (<15 characters is a good tradeoff)

Avoid names that are similar or differ only in case

Use **Hungarian Notation** in casing for Class names, Method names, variables and method parameters

```
public class HelloWorld
{
        ...
}

void sayHello(String name)
{
        ...
}

int totalCount = 0;
void sayHello(String name)
{
        String fullMessage = "Hello " + name;
        ...
}
```

Use **Meaningful, descriptive** words to name **variables**. Do **not** use abbreviations.

Good:

```
string address
int salary
```

Not Good:

```
string nam
string addr
int sal
```

Do not use single character variable names like i, n, s etc. Use names like index, temp
One exception in this case would be variables used for iterations in loops:

```
for ( int i = 0; i < count; i++ )
{
        ...
}
```

If the variable is used only as a counter for iteration and is not used anywhere else in the loop, many people still like to use a single char variable (i) instead of inventing a different suitable name.

Do **not** use underscores (_) for local variable names.

Do not use variable names that **resemble** keywords.

Prefix boolean variables, properties and methods with "is" or similar prefixes.
Ex: private **boolean isFinished**

File name should match with class name.

Use Pascal Case for file names.

Use appropriate **prefix** for the UI elements so that you can identify them from the rest of the variables.

In some of the components, we have used certain pre defined postfix to reflect the meaning of the component.

Eg.

1. OperationListPanel
2. ClientListPanel
3. InformativePanel

Etc are all panels (As suggested from their terminating character sequence)

Indentation and Spacing

Use **TAB** for indentation. Do not use **SPACES**. Define the Tab size as 5.

Comments should be in the same level as the code (use the same level of indentation).

Good:

```
// Format a message and display

string fullMessage = "Hello " + name;
DateTime currentTime = DateTime.Now;
string message = fullMessage + ", the time is : " +
currentTime.ToShortTimeString();
MessageBox.Show ( message );
```

Not Good:

```
// Format a message and display
        string fullMessage = "Hello " + name;
        DateTime currentTime = DateTime.Now;
        string message = fullMessage + ", the time is : " +
        currentTime.ToShortTimeString();
        MessageBox.Show ( message );
```

Curly braces ({}) should be in the **same level** as the code outside the braces.

```
if ( ... )
{
        // Do something
        // ...

        return false;
}
```

There should be one and only one single blank line between each method inside the class.

The **curly braces should be on a separate line** and not in the same line as if, for etc.

Good:

```
if ( ... )
{
        // Do something
}
```

Not Good:

```
if ( ... ) {
        // Do something
}
```

Use **a single space before and after each operator** and brackets.

Good:

```
if ( showResult == true )
{
        for ( int i = 0; i < 10; i++ )
        {
                //
        }
}
```

Not Good:

```
if(showResult==true)
{
        for(int     i= 0;i<10;i++)
        {
                //
        }
}
```

Naming Constants

In Java, constants, values that do not change, are typically implemented as *static final* fields of classes. The convention is to use full English words, all in upper case, with underscores between the words

Example
MINIMUM_BALANCE
MAX_VALUE
DEFAULT_START_DATE[1]

Field Visibility

Fields should not be declared public for reasons of encapsulation. All fields should be declared private and accessor methods should be used to access / modify the field value. This results in less coupling between classes as the protected / public / package access of field can result in direct access of the field from other classes

Declaration

One declaration per line is recommended since it encourages commenting. In other words,

```
int level; // indentation level
int size;  // size of table
```

is preferred over

```
int level, size;
```

Do not put different types on the same line. Example:

```
int foo,  fooarray[]; //WRONG!
```

Note: The examples above use one space between the type and the identifier. Another acceptable alternative is to use tabs, e.g.:

```
int     level;          // indentation level
int     size;           // size of table
Object  currentEntry;   // currently selected table entry
```

[1] Java Coding Standards

Initialization

Try to initialize local variables where they're declared. The only reason not to initialize a variable where it's declared is if the initial value depends on some computation occurring first.

Class and Interface Declarations

When coding Java classes and interfaces, the following formatting rules should be followed:

No space between a method name and the parenthesis "(" starting its parameter list
Open brace "{" appears at the end of the same line as the declaration statement
Closing brace "}" starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the "}" should appear immediately after the "{"

```
class Sample extends Object {
   int ivar1;
   int ivar2;

   Sample(int i, int j) {
      ivar1 = i;
      ivar2 = j;
         }

   int emptyMethod() {}

   ...
}
```

5.3. SCREENSHOTS

Fig. 5.1 Login

1. Provide manager's username.
2. Provide manager's password.
3. Click **login** button to log in.

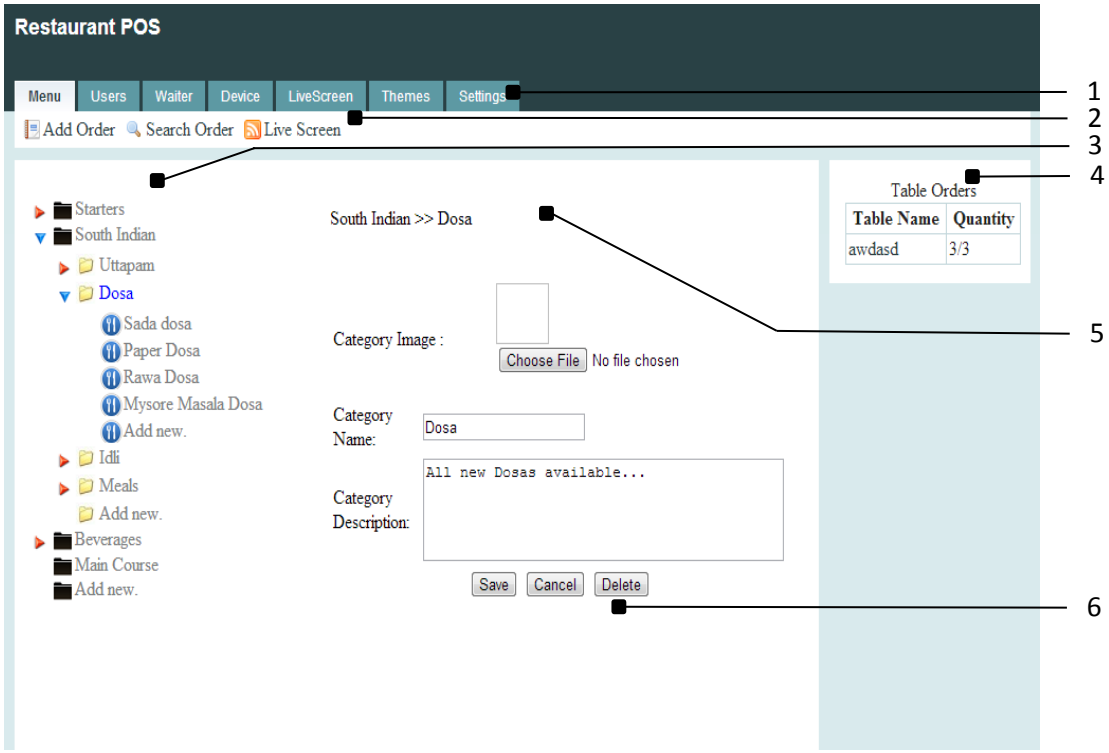

Fig 5.2 Category Details

1. These are different tabs for different functionality to be performed by the manager. Currently selected tab is “Menu” shown by white background.
2. This shows the basic operation that will be used by the manager at any point of time.
3. Since the currently selected tab is “Menu” Point 3 shows the categories and their subcategory, and their items within it. Current Selected Category is “South Indian” Current Selected Subcategory is “Dosa”.
4. This table shows the constantly updating table order details. It Shows table name, order quantity served/total quantity ordered.
5. This section of the page is used to add category/subcategory or it shows the details of selected category/subcategory. Currently these are the details of the subcategory “Dosa”
6. “Save” Button is used to save the modified details.

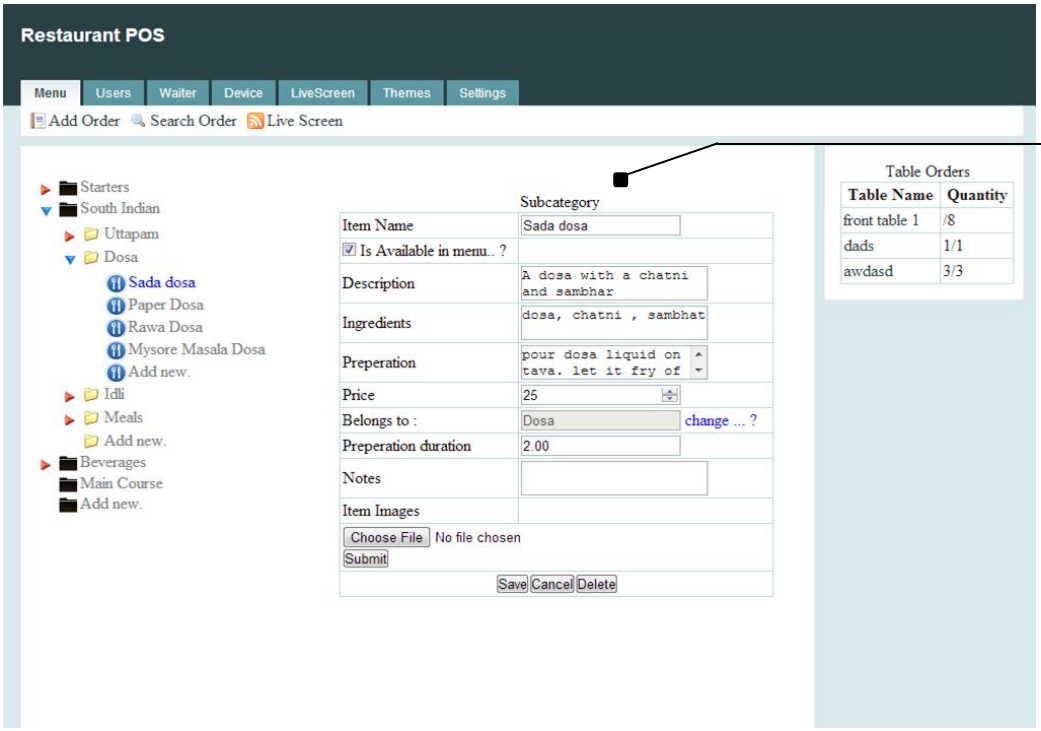

Fig. 5.3 Item Details

1. These are the details of the selected Items. Manager can modify the details and save it. And can even delete it. Currently selected item is Sada Dosa, so it displays the details of sada dosa.

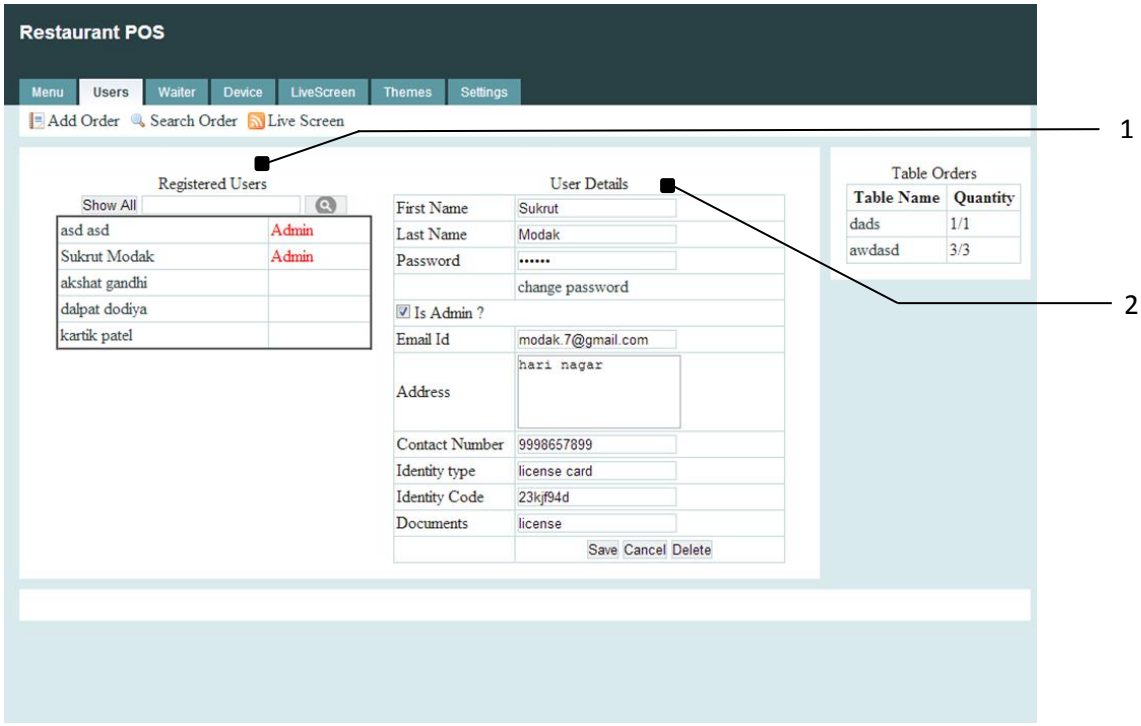

Fig. 5.4 User Details

1. The currently selected tab is “Users”. So point 1 displays all the users registered to the system. Users can be admin or normal customer.
2. These are the details of the user selected in the list shown on left side. Manager can save and delete the user.

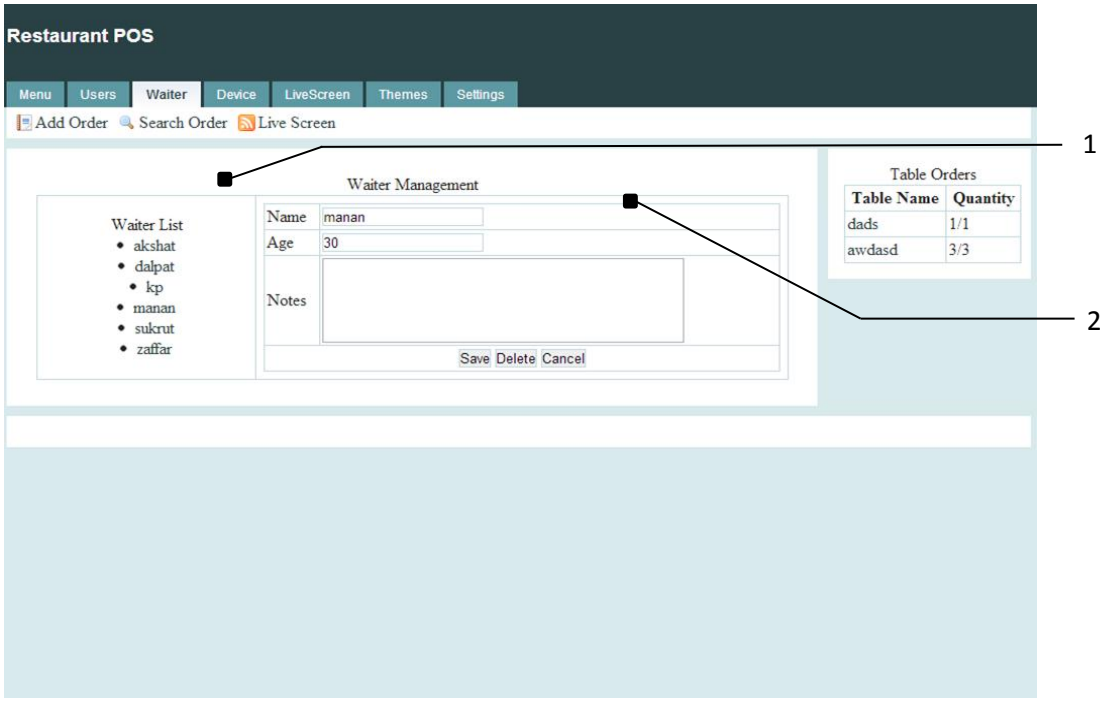

Fig. 5.5 Waiter Details

1. The currently selected tab is “Waiters”. So point 1 displays all the waiters registered to the system. These registered waiters will be used to assign the tablets.
2. These are the details of the waiter selected in the list shown on left side. Manager can save and delete the waiter.

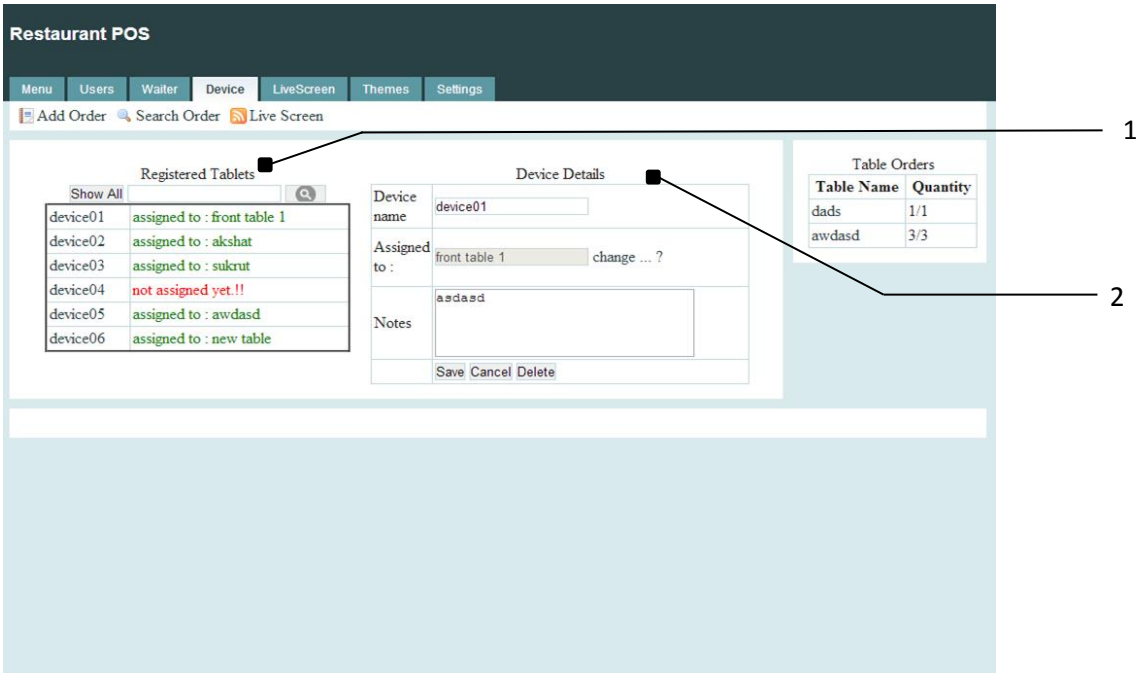

Fig. 5.6 Device Details

Selected tab is “Device”

1. Displays the list of registered devices which will be used to place the order. It also displays the waiter or table assigned to the device.
2. This portion shows the details of the selected device. Manager can save details or add new device.

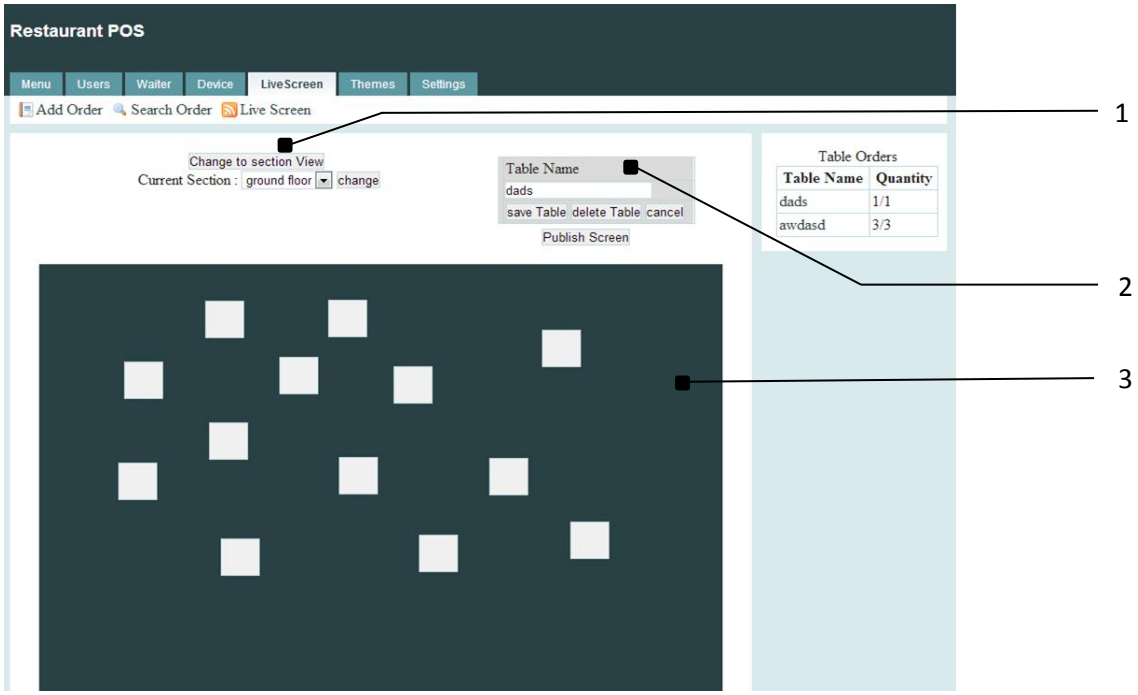

Fig. 5.7 Live Screen Generation

1. Selected tab is “Live Screen” which will display the Live Screen Generation Parameters. Point 1 shows the button “Change to Section View” which will display the orientation of the sections of the restaurant. It also shows the current section selected, and manager can change the section and point 3 will display the orientation of tables in the selected section.
2. This part shows the details of the selected tab. Manager can modify the table name.
3. This is the portion which will display the orientation of the tables/section. User can drag and change the position of the tables and section.

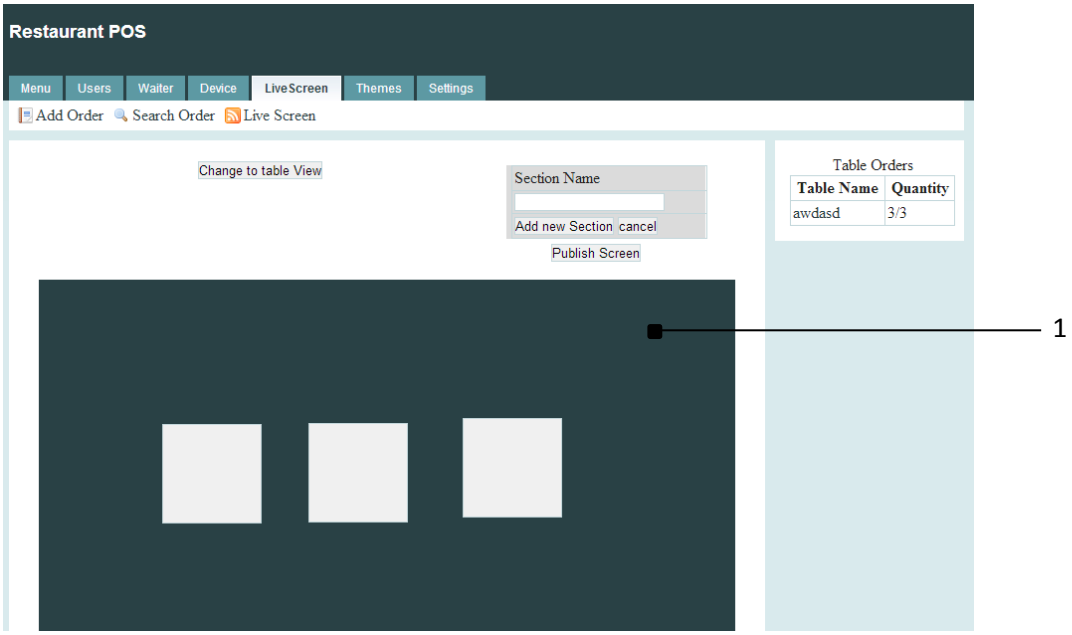

Fig. 5.8 Live Screen Generation

1. This is the section view of the physical layout of the different sections of the restaurant.

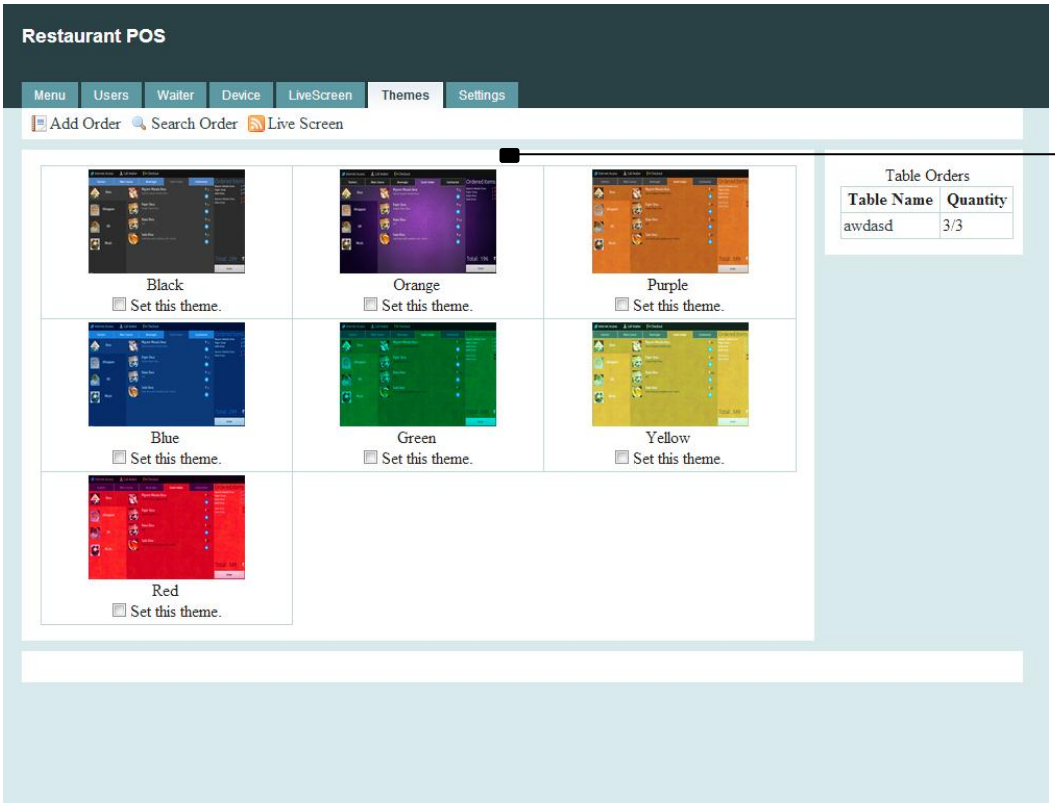

Fig. 5.8 Themes Selection

1. List of themes which can be set by the manager for android application.

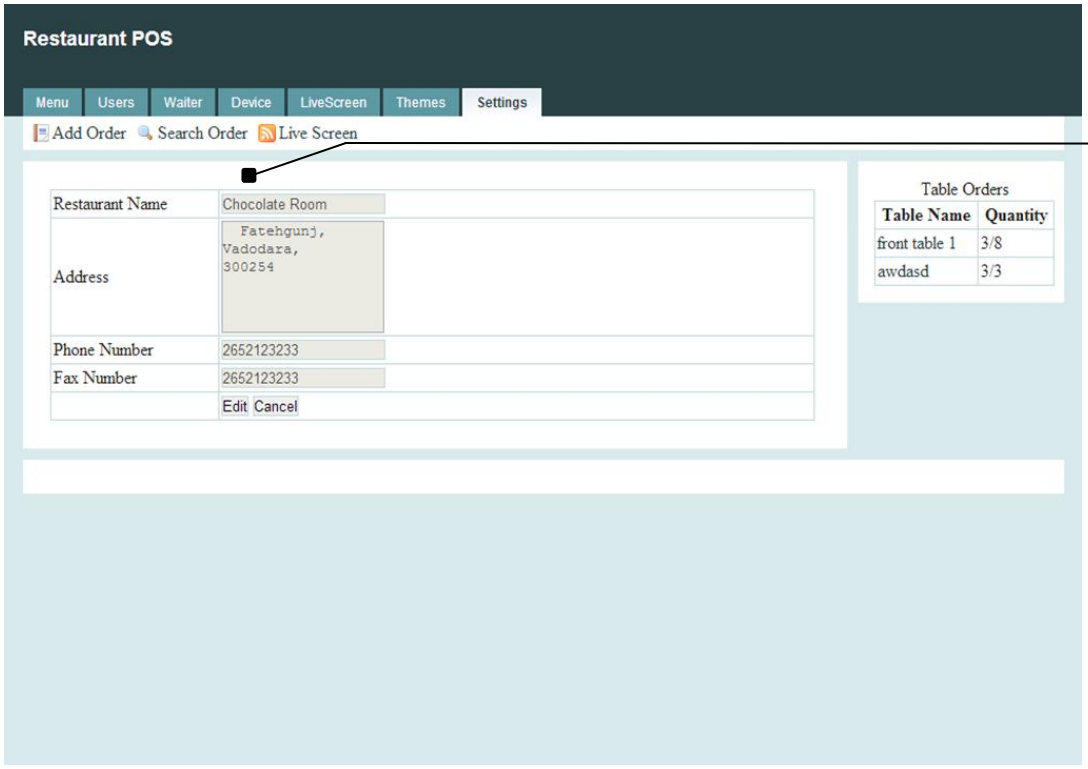

Fig.5.9 Restaurant Settings

1. This page contains the basic settings of the restaurant, like restarant name, address which will be used for invoice generation.

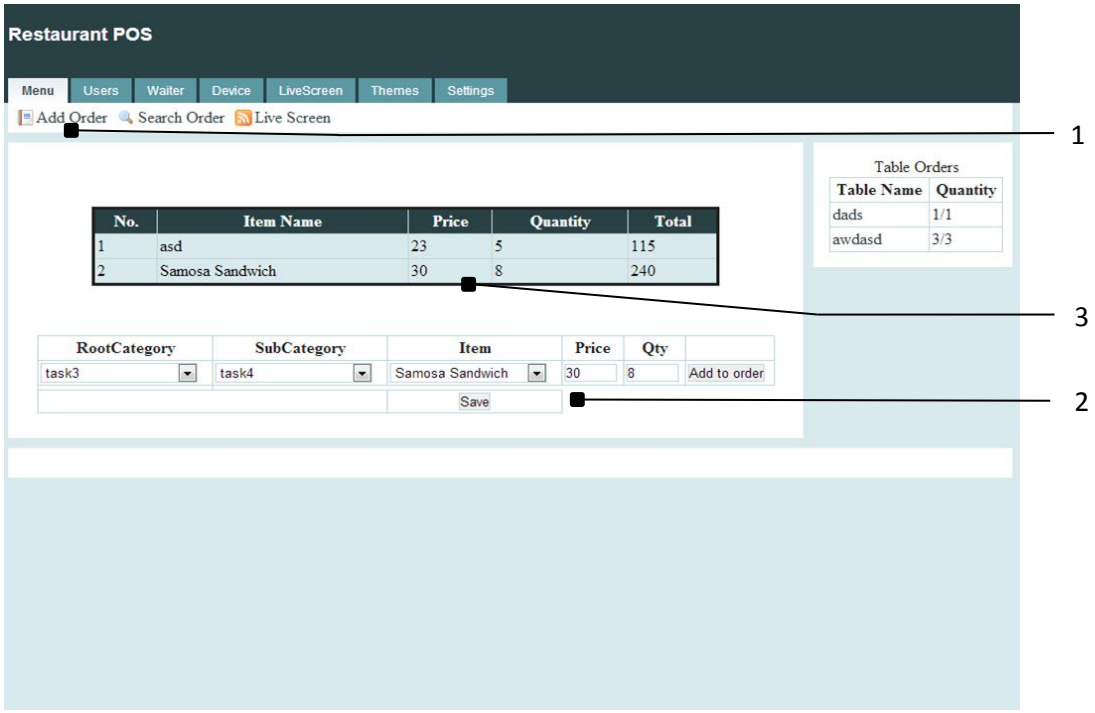

Fig. 5.10 Add Order Page

1. The "Add order" button displays the UI to add orders in case of any parcel orders.
2. This portion shows the categories, subcategories, and items list, manager selects the particular root category, then the subcategory and finally the item. And enter the required quantity and click on "Add to Order" button.
3. The order gets added into this table showing the summary of the items ordered with price, qty, and total.

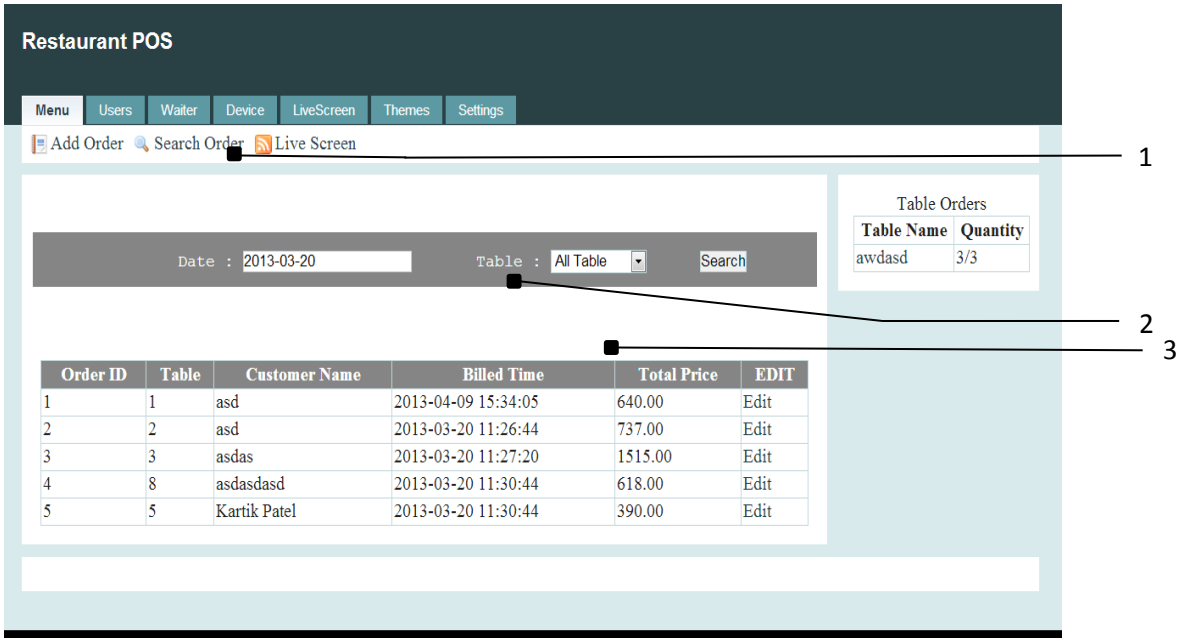

Fig.5.11 Order Search Page.

1. The Search Order Button displays the UI to search all past orders.
2. The Date field allows the manager to select a past date and Table field allows to select Particular table if required and search button displays the search in below table
3. The table displays the result of search according to search criteria.

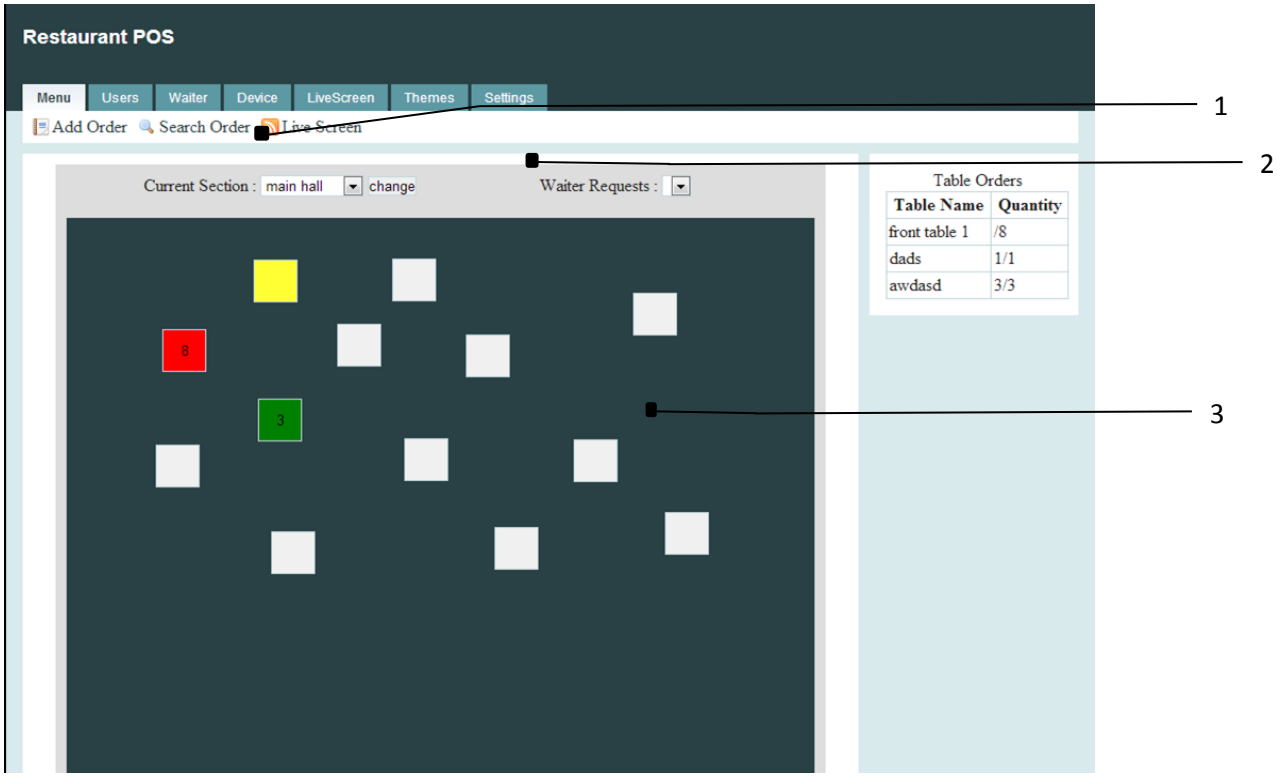

Fig.5.12 Live screen for all the orders

1. The Current section drop down allows user to select section for which the table will be displayed.
2. The Waiter request field notifies the manager about the waiter request by a customer. The field turns to red color when waiter request comes.
3. The division is use to display the layout of selected section. When a new order comes the respective table blinks with red color, when the order is seen and approved the respective table color changes to yellow, after the order is served the respective table color changes to green and when the customer request for checkout the respective table blinks with yellow color.

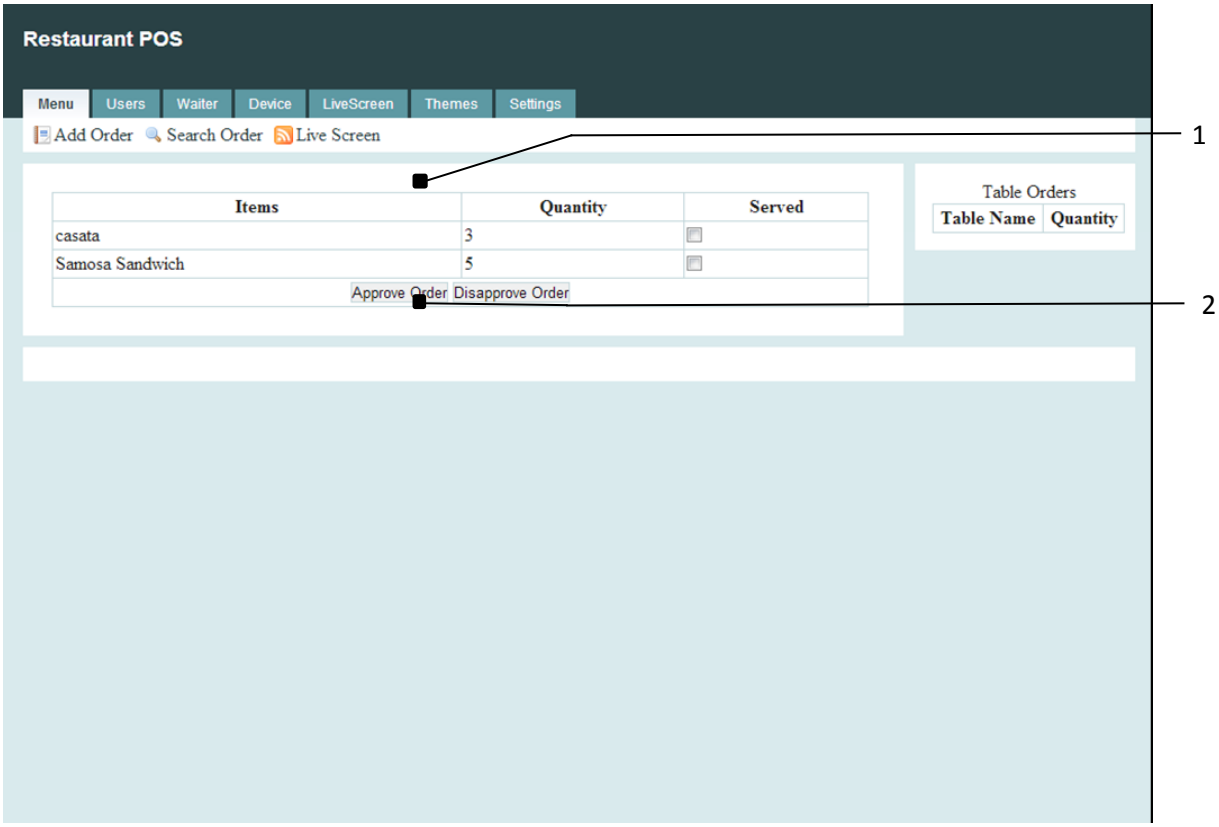

Fig. 5.13 Order details for particular order

1. The Table shows the items ordered with quantity and a field which denotes whether the item is served or not.
2. The Approve/disapprove buttons update the status of the order.

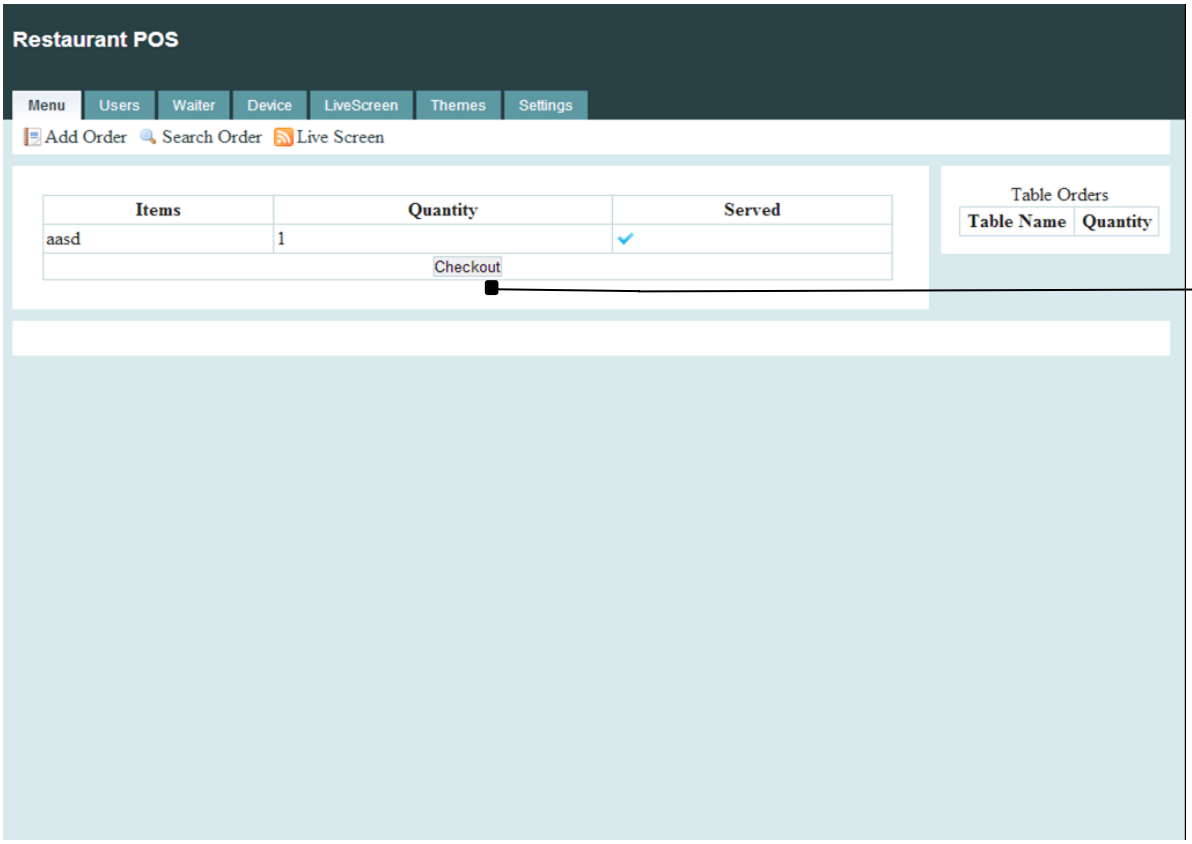

Fig. 5.14 Order item details

1. The Table shows the items ordered with quantity, if all the items are served and request for checkout comes, the manager can click on checkout which opens the bill page.

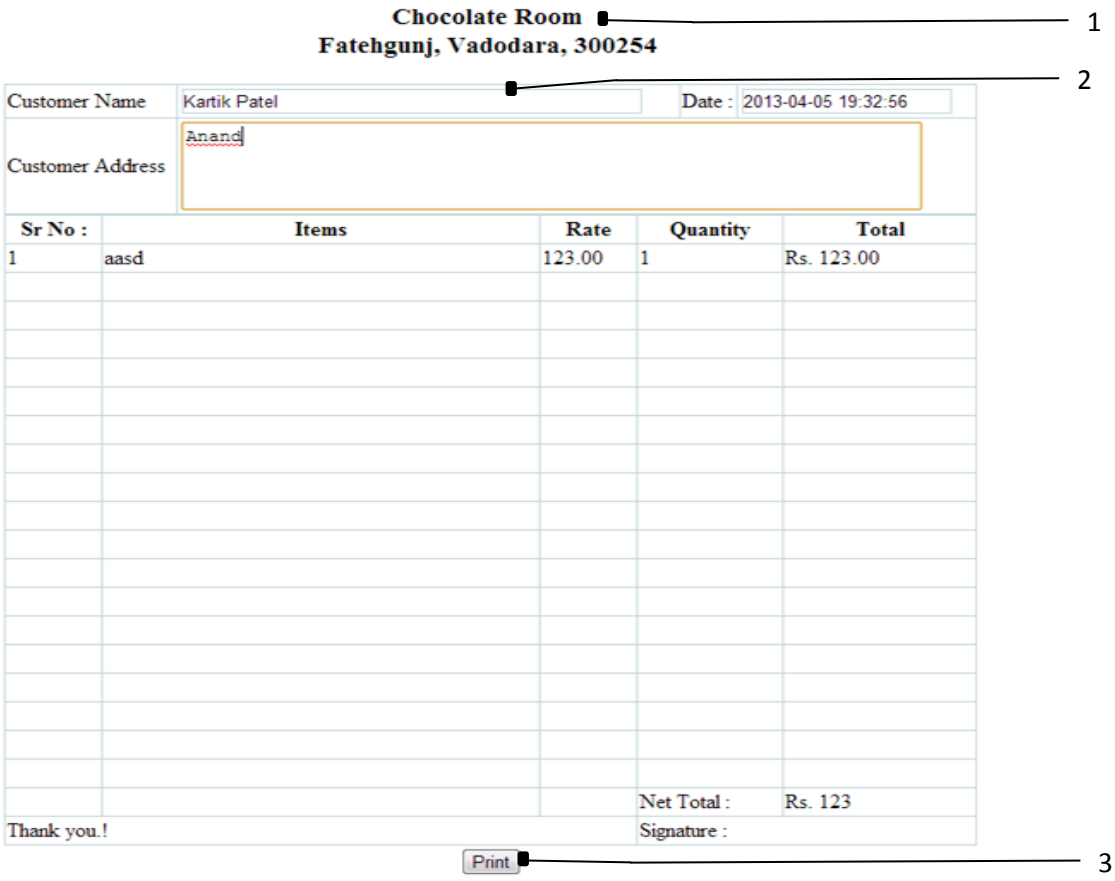

Fig.5.15 Bill Generation

1. This is the name of the company, which comes from the database.
2. The Manager/receptionist enters the customer name and address to generate the invoice.
3. Print Button is used to take printout of the invoice.

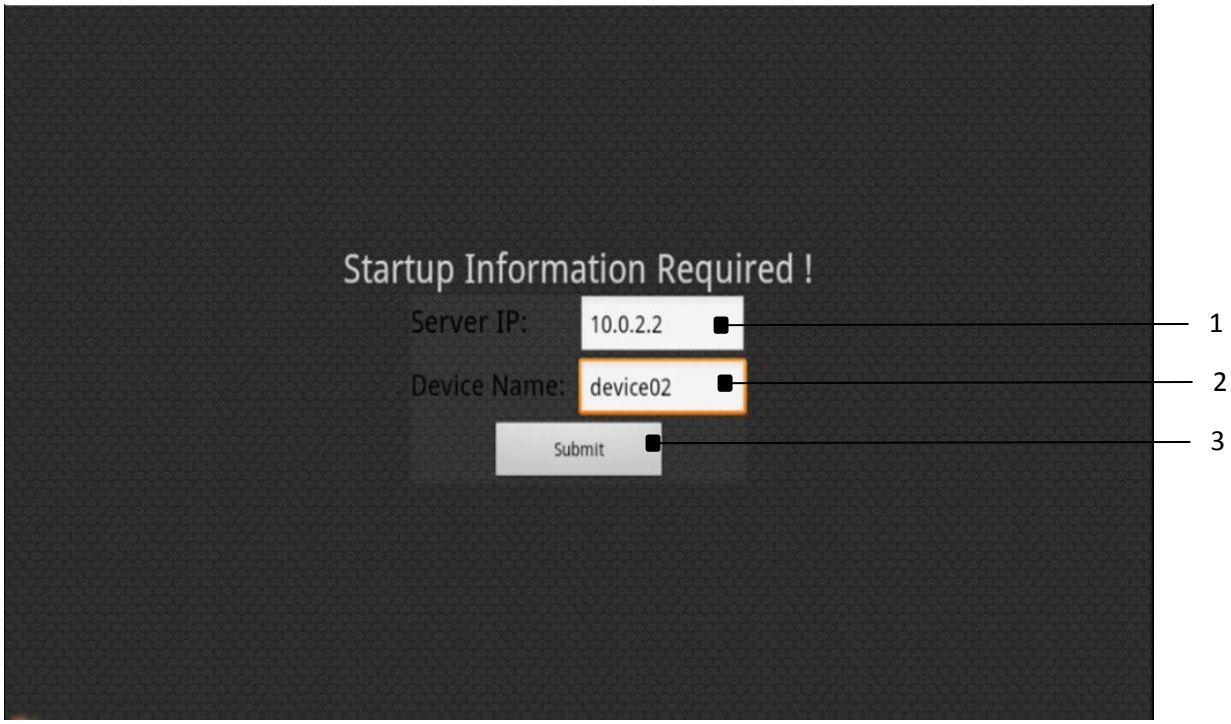

Fig. 5.16 Android Startup page

1. Manger has to provide server ip address to connect it to the server.
2. The unique device name to identify this device.
3. Submit Button to save the details in the device.

Fig. 5.17 Android Ordering Screen (Theme-1)

1. Quick Links to Internet access, Call waiter, and checkout.
2. Main Category tabs. Currently selected tab is SouthIndian.
3. Subcategory List. Currently selected subcategory is Dosa.
4. Items list under selected subcategory. Each item in the list contains the Item name, description, price and an add button. By clicking on the add button the item gets added into the order list
5. List of ordered and not yet ordered items.
6. Displays the total amount, and the order button to order the currently unordered items.

Fig. 5.18 Android Ordering Screen (Theme-2)

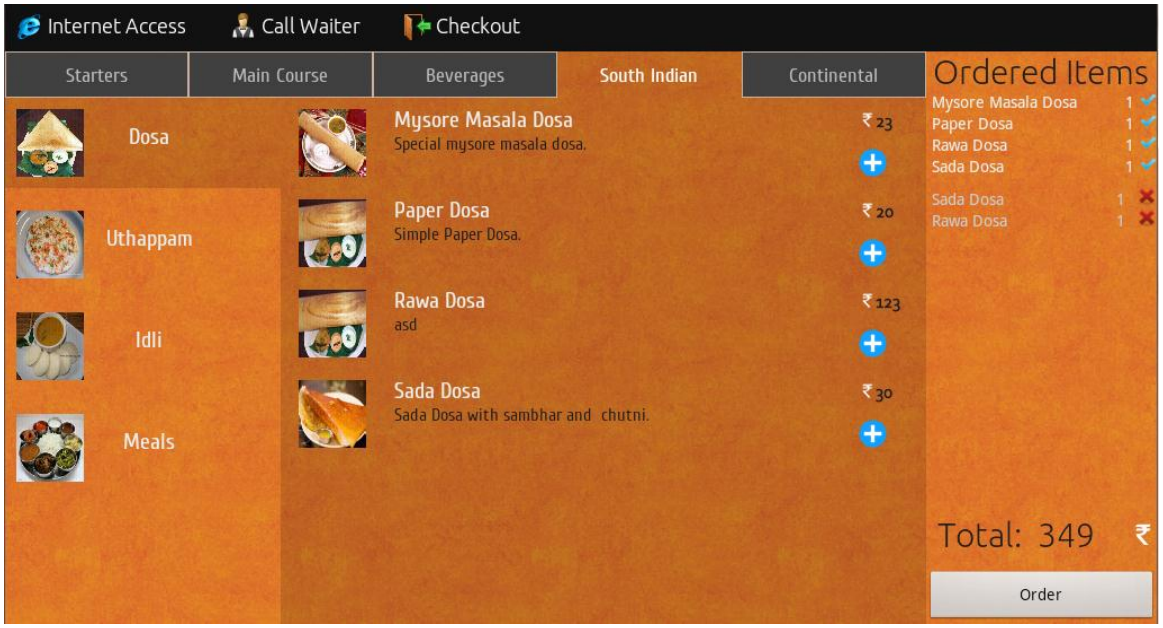

Fig. 5.19 Android Ordering Screen (Theme-3)

1. The same screen with different theme

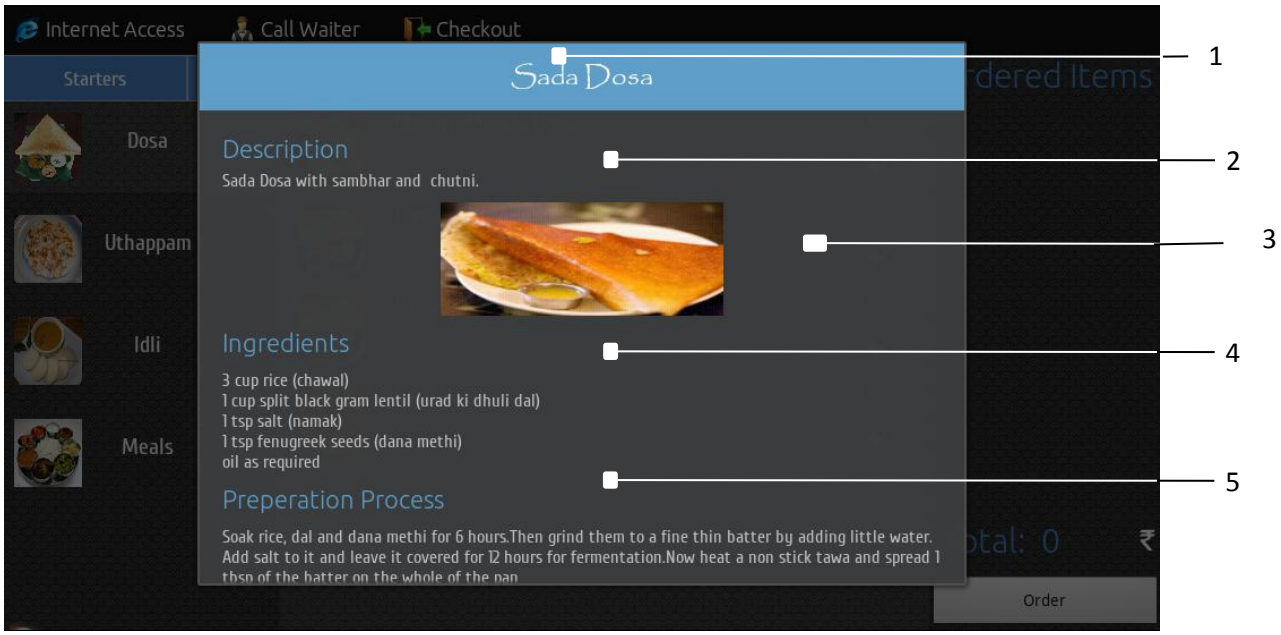

Fig. 5.20 Item Description Screen (Theme-1)

1. Item name
2. Description
3. Item Images.
4. Item Ingredients
5. Preparation process

Fig. 5.21 Item Description Screen (Theme-2)

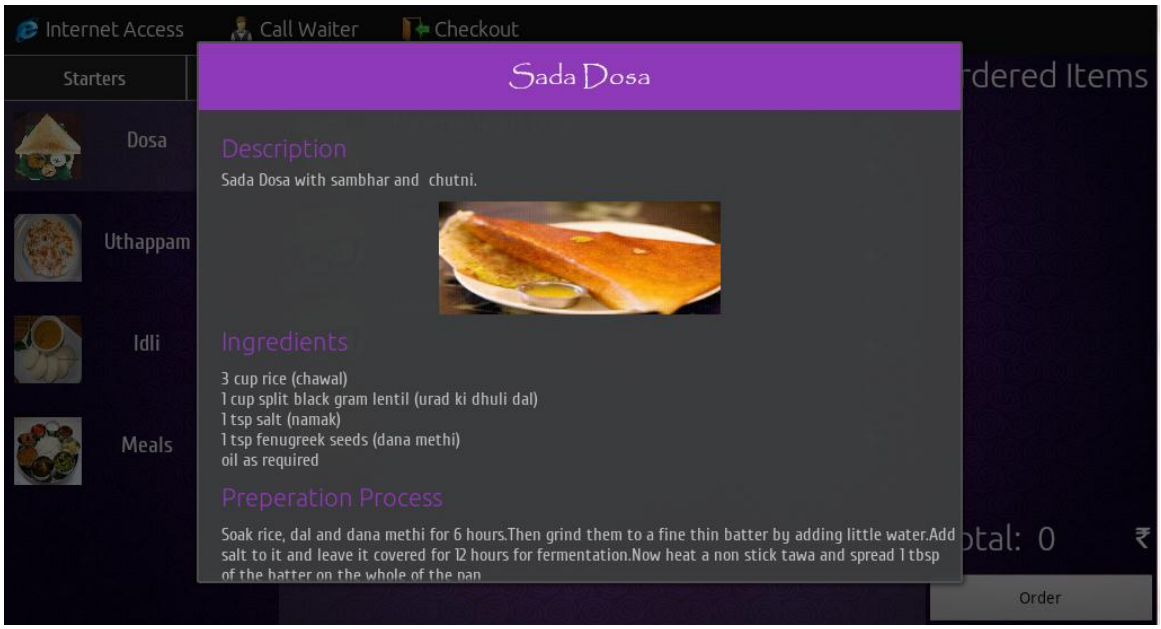

Fig. 5.22 Item Description Screen (Theme-3)

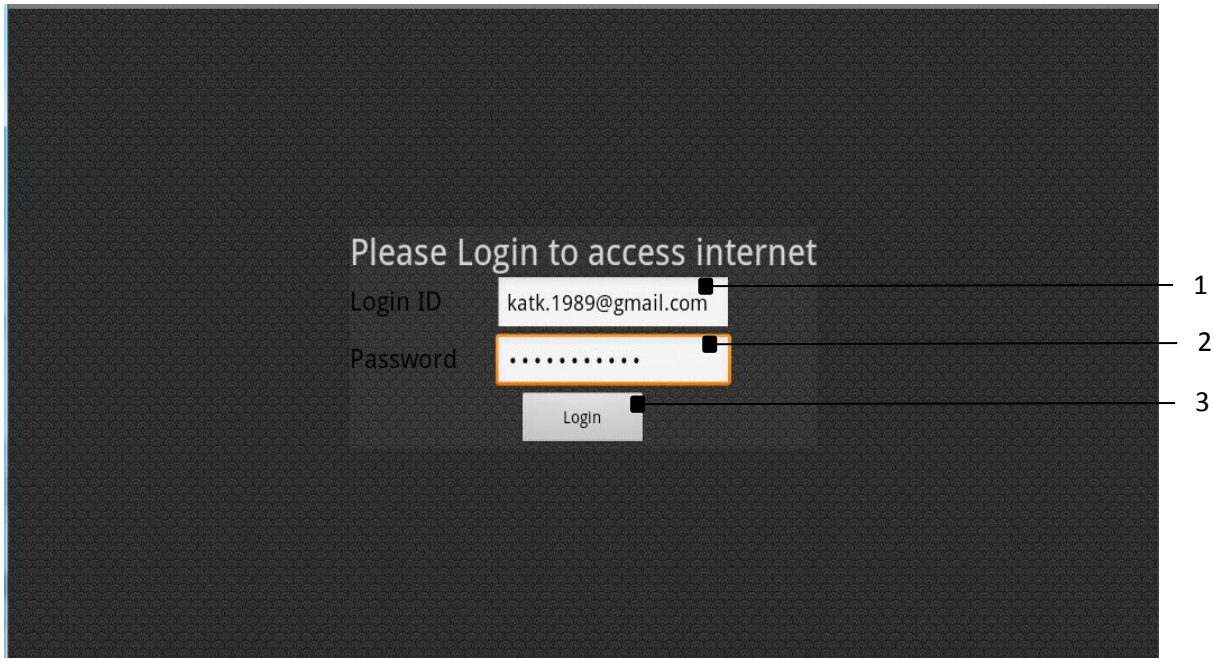

Fig. 5.23 Login screen for internet access

1. Customer's email id to be used as login id.
2. Customer's password.
3. Login button to log in.

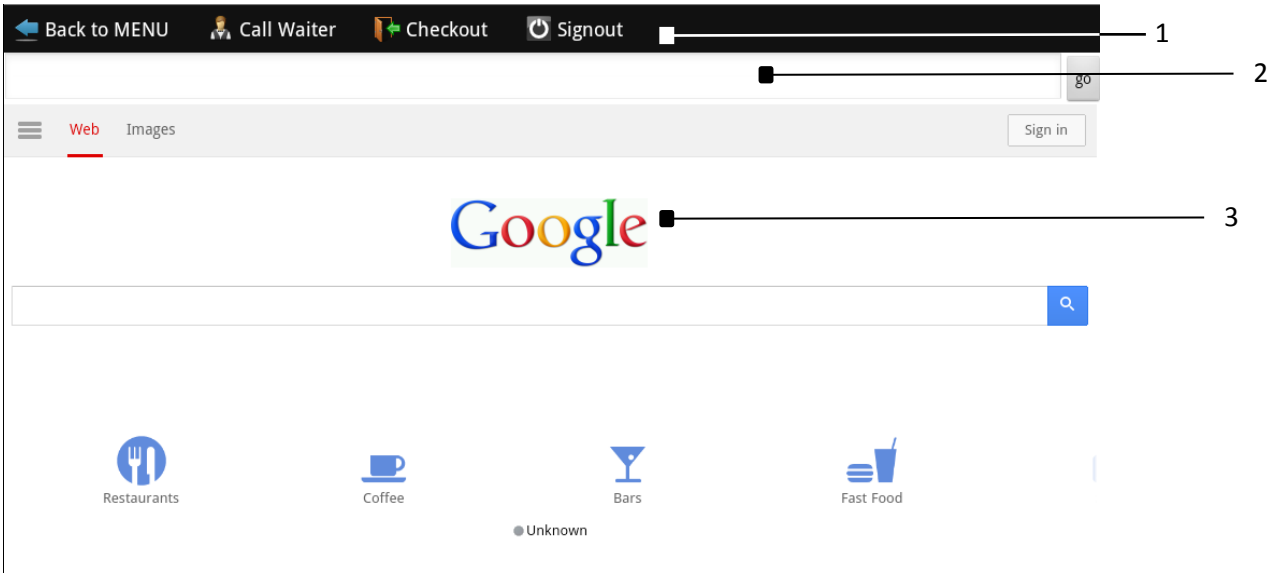

Fig. 5.24 Internet access

1. Quick links to go back to menu screen, sign out from internet etc.
2. User enters the url and clicks the go button.
3. The content of the url is displayed here.

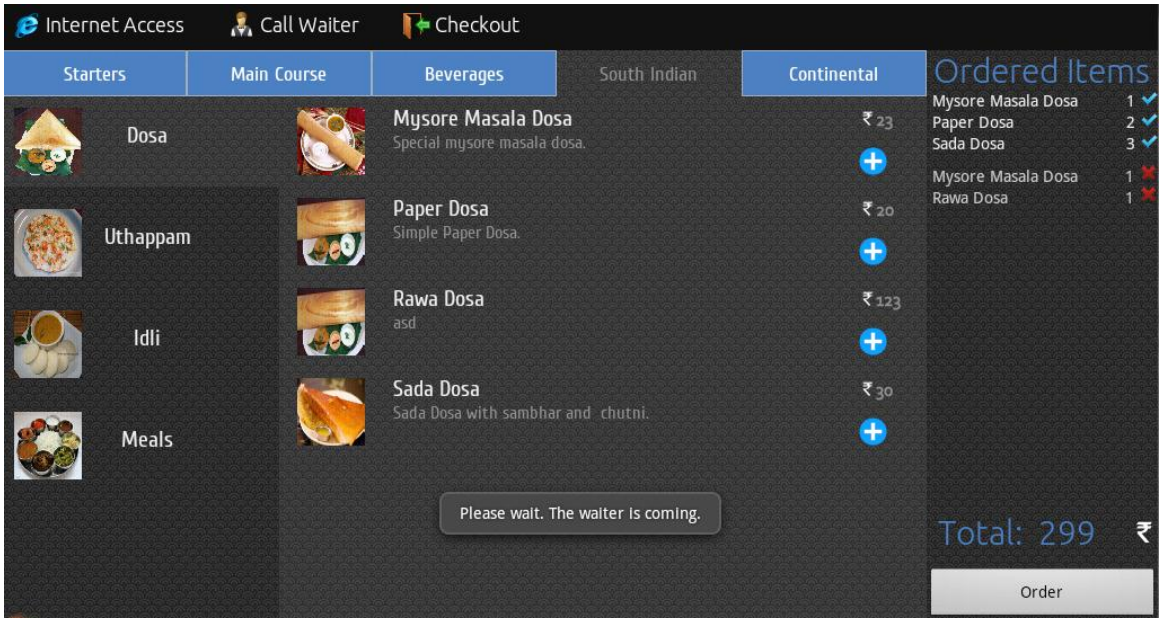

Fig. 5.25 Android Startup page

1. On clicking the Call waiter button, the request is send to the manager and a message a displayed saying “Please wait. The waiter is coming”.

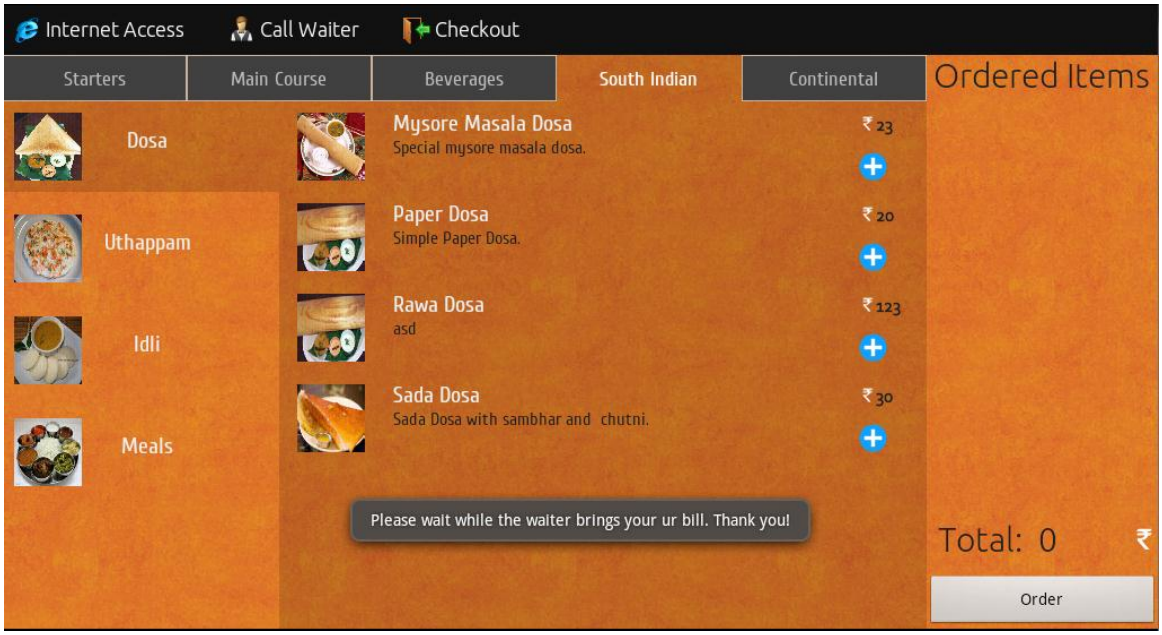

Fig. 5.26 Android Startup page

1. On clicking the Checkout button, the request is send to the manager and a message a displayed saying “Please wait. While the waiter brings you your bill. Thank you!”.

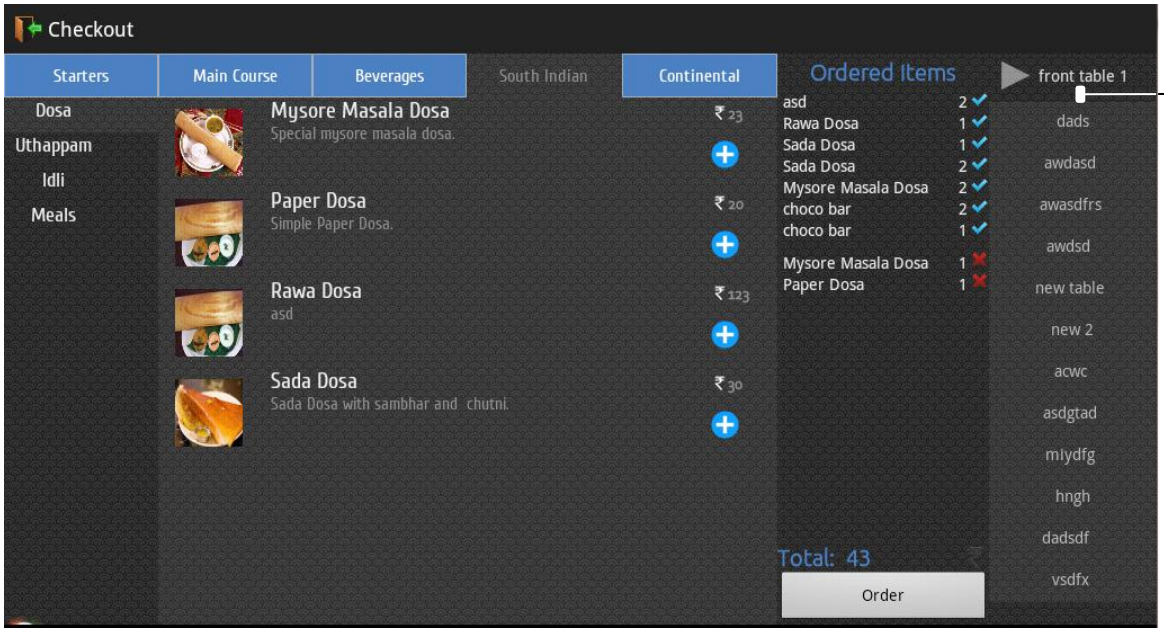

Fig. 5.27 Waiter's Screen

1. In the waiter mode the tables are displayed on the right corner. And the order for the selected table will be displayed.

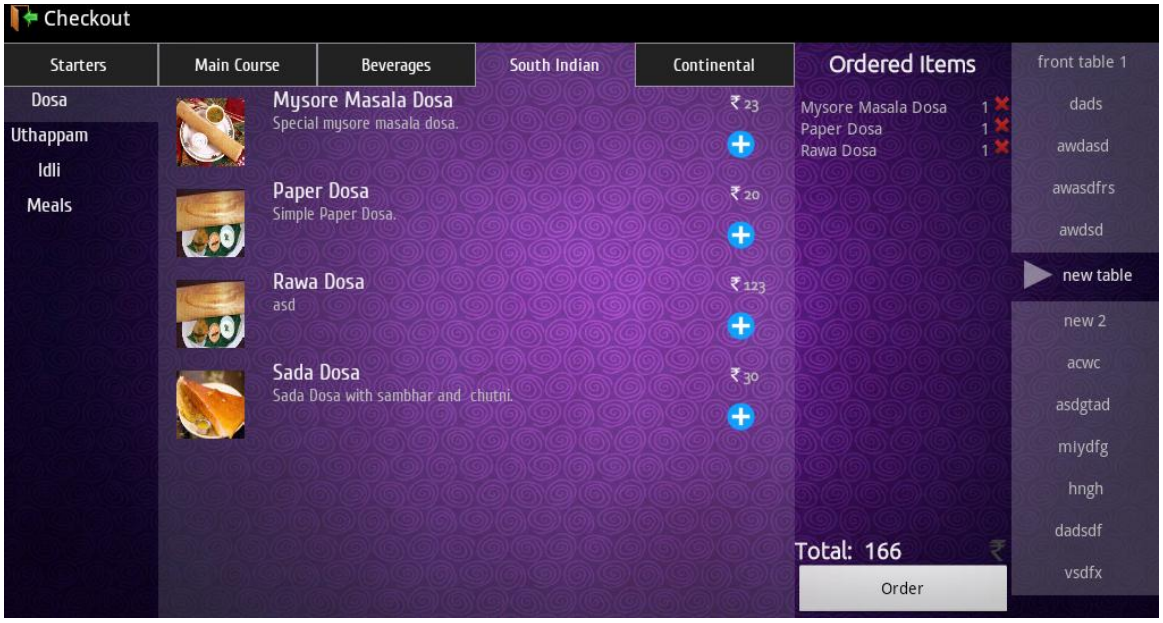

Fig. 5.28 Waiter Screen (Theme-2)

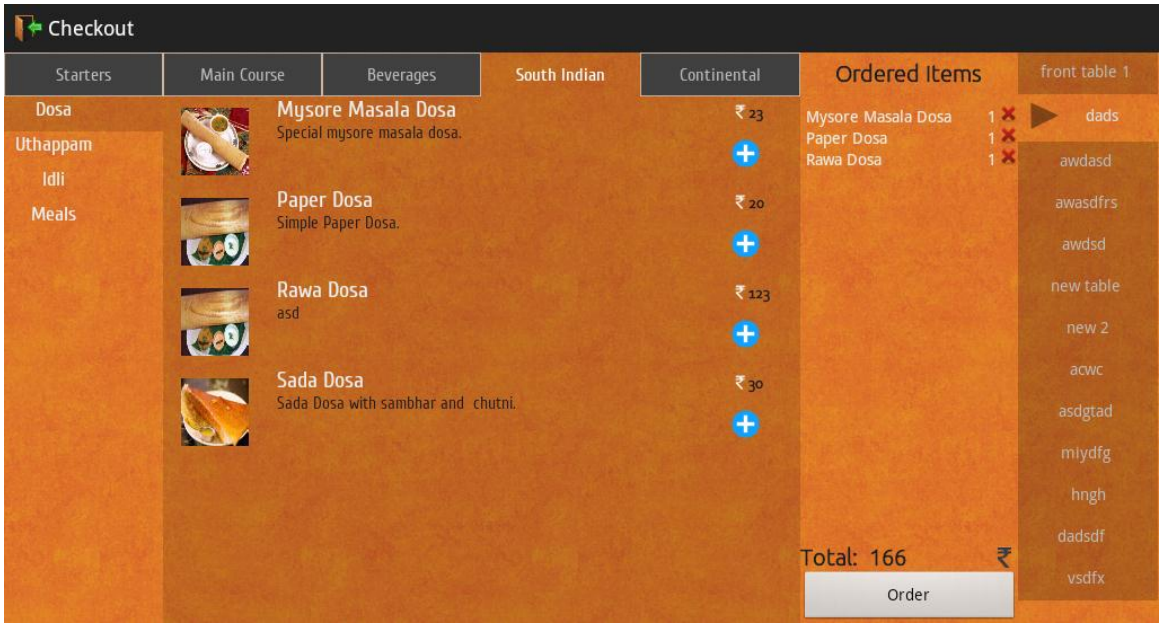

Fig. 5.29 Waiter Screen (Theme-2)

Software testing has a dual function; it is used to establish the presence of defects in the program and it is also used to help judge whether or not the program is usable in practice. Software testing is used for validation and verification, which ensures that the software conforms to its specification and meets the need of the software customer. The testing is done by our Team members that act as novice users and test the application with all possible ways to find the bugs and error as well as check validation.

6.1. TESTING PLAN

- **The Testing process**

We tested the software process activities such as Design, Implementation and Requirement Engineering. As design errors are costlier to repair, once, the system has started to operate it is therefore quite obvious to repair them all at the initial stage of the system.

- **Requirements Traceability**

As most interested portion in the system is system meeting its requirements therefore testing should be planned so that all requirements are individually tested, we checked the output of certain combination of inputs, which gives desirable output or not. Strictly going along our requirements specifications gave us the path to get desirable result for system users.

- **Tested Items**

Our test items were like, validation of each and every field when user enters the data. The user is not allowed to enter incorrect data and also he is not allowed to leave the text boxes blank. The mandatory fields which were the necessary fields were tested to contain data and not blank. Places where only numeric or alphabetic characters were required, testing was done to see that no data other than required was entered. Also at some places, the user must select an option out of many. He should not be allowed to leave the page without selecting any value from the provided options in such cases.

- **Testing schedule**

We have tested each module back to back so that errors and omissions can be found as early as possible. Once the system has been developed fully we tested it on other machines by deploying it on various machines.

- **General Plan for Testing**

The Network Administrating System is tested using Bottom-Up testing strategy.

1. Testing of each stored procedure or query (select / insert / update) with standard inputs.
2. Testing of each individual class of the assembly.
3. Testing of each individual user interface class.(GUI Components)
4. Module wise testing (Also sub module wise testing) while code development.(Testing of related Classes)
5. Integration testing of system after integration of individual modules.(Testing of all the classes so as to verify that they fulfill the purpose of the functionality)
6. Security testing.
7. User level testing.
8. Robustness testing. (To test that the system works for long durations under extreme conditions also)

6.2. TESTING STRATEGY

Testing strategies is a general approach to the testing process rather than a method of devising particular system or components tests. Different testing strategies may be adopted depending on the type of system to be tested and the development process used. So considering functional oriented nature of this software we adopted mixture of following strategies.

Testing is the process carried out on software to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications.

Testing is advantageous in several ways. Firstly, the defects found help in the process of making the software reliable. Secondly, even if the defects found are not corrected, testing gives an idea as to how reliable the software is. Thirdly, over time, the record of defects found reveals the most common kinds of defects, which can be used for developing appropriate preventive measures such as training, proper design and reviewing.

The testing sub-process includes the following activities in a phase dependent manner:

- Create Test Plans.
- Create Test Specifications.
- Review Test Plans and Test Specifications.
- Conduct tests according to the Test Specifications, and log the defects.
- Fix defects, if any.
- When defects are fixed continue from activity.

The development process repeats this testing sub-process a number of times for the following phases.

- Unit Testing.
- Integration Testing.
- System Testing.
- Acceptance Testing.

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct. System Testing ensures that the system meets its stated design specifications. Acceptance Testing is testing by the users to ascertain whether the system developed is a correct implementation of the Software Requirements Specification.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

6.3. TESTING METHODS

TESTING STAGES:

- **Unit testing**

  Unit testing focuses verification effort on the smallest unit of software design the software component or module. In this type of testing the individual modules were tested and verified whether accurate output was made available or not. The modules were tested individually as they could result into faster responses than when they were integrated.

- **Integration Testing**

  When the unit testing was over, all the modules were integrated one by one and tested as a whole. It might be possible that all modules may work individually, but they may not work when put together. Data can be lost across the interface, one module can have an adverse affect on the other or sub functions of another, when combined may not produce desired major function, individually acceptable imprecision may be magnified to unacceptable level; global data structure can present problem. So any system has to be tested this way so that the final output is the desired one. Also the common functions throughout the system were taken and formed into a class so that they could be accessed from the same place without creating any ambiguities.

- **Validation Testing**

  After the integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected, validation testing begins. Validation testing can be defined in many ways but a simple definition is that a validation succeeds when the software functions in a manner that can be reasonably accepted by the Client.

- **System testing**

  Any software is only one element of a larger computer based system. Ultimately software is incorporated with other system elements like hardware, people, information and a series of system integration and validation tests are conducted. System testing is

actually a series of different test whose primary purpose is to fully exercise the computer based system. A necessary check to be performed was that of the new system being accepted by the older window's application. The data is not being misplaced when both the systems exist.

- **Storage Testing**

  The database of the system has to be stored on the hard disk. So the storage capacity of the hard disk should be enough to store all the data required for the efficient running of the system. This must be tested by artificially dumping large amounts of data into the database.

6.3.1 ACCEPTANCE TESTING

Before delivering the product to the client, it will be tested for stability and correctness of results. By conducting the fore mention test cases, we should be able to conclude on the reliability of the product before the client tests it for acceptance. First tests will be conducted on a sample portion of the database. Later tests will also include testing on the entire database. The client will test the program for criteria mentioned in the next section before accepting the product.

Specific **acceptance criteria** (the client expects the following):

- Conformity with the predefined requirements.
- Correct data returned from the MySQL database searches
- Functionality in multiple environments. Since this is a client-server product, it must be compatible to various settings on different systems.

6.3.2 FEATURE LEVEL TESTING

Task-Oriented Functional Tests

The mandatory fields must not be left empty.

Email-id should contain '@ 'and '. 'Characters.

Date should be in DD/MM/YYYY format.

User name and Password must match with registered details.

Searching facilities is performed by specified criteria.

Integration Tests

Components or modules that should be tested independently are following

1) Browser Compatibility.

2) Hardware compatibility.

3) Interface Testing.

4) User Input Testing.

System-Level Tests

In system level tests we will use black-box testing since all independent modules or functionalities are already tested before. System level test will start after unit and integration testing gets completed.

Unstructured Tests

As user input to the system is not drastically varying, it will not require Unstructured Tests. Input related modules would be tested with set of test data, which covers whole data set of user input.

Volume Tests

In other scenario we will test system against large amount of transactions that will help in testing system to function properly at high stress levels.

Performance Tests

Systems have certain features that highly affect their takeaway value. These features need to be tested against its performance.

- User Response Time - Should be moderate. It should be in range of 1-5 seconds.(Maybe between 5 to 10 in case of informative operations)
- File Searching Time - Should be in range of 4-5 minutes.(For searching on the entire network)
- System Load Time - Should be in range of 1-5 seconds.

6.3.3 CONFIGURATION AND COMPATIBILITY TESTING

Application Tests

- Tests were conducted to find errors and accuracy when requests were made for different applications. (Example file transfer for files of different formats etc)
- Test response time to users by the MySQL database.
- These tests were focused on operating large amount of data by MySQL database, and also data fetch time for response to user.

6.4 TEST CASES

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
Username is blank.	Username	Blank	Error Message	Enter Username	Passed
Correct Username is entered.	Username	e1	Valid	Valid	Passed
Incorrect User name is entered.	Username	m1	Error Message	Enter Correct Username	Passed
Length of Username	Username	More than 15 character	Error Message	Username can't be more than 15 characters	Passed
Password is blank.	Password	Blank	Error Message	Enter Password	Passed
Correct Password is entered.	Password	1	Valid	Valid	Passed
Incorrect Password is entered.	Password	dgdsfsd	Error Message	Enter Correct Password	Passed
Length of Password	Password	More than 15 characters or less than 8 characters	Error Message	Password should be of length 8 to 15 characters	Passed

Table 6.1 Test case for Manager/Receptionist Login

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
No Images Uploaded for menu items	Images selected	Blank	Error Message	Select at least 1 Image	Passed
Item Name is blank	Item name	Blank	Error Message	Enter Item Name	Passed
Price is Blank	Price	Blank	Error Message	Enter Price	Passed
Enter Alphabet in Price	Price	12o	Error Message	Price Should be Numeric	Passed
Expansion Operation on the Category and Item tree	Selection of a Category / Subcategory	Single Click on Category/ Subcategory	Expansion of Menu in tree format to show the Subcategories and Items respectively	Proper expansion with nesting and indentation	Passed
Compression Operation on Category and Item tree	Selection of an already visited Category / Subcategory	Single Click on Category/ Subcategory	Compression of tree	Compression of tree	Passed
No Images Uploaded for Subcategory	Images selected	Blank	Error Message	Select One Image	Passed
Item Name is blank	Item name	Blank	Error Message	Enter Item Name	Passed

Table 6.2 Test case for Menu Items

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
Blank Device Name	Device Name	Blank	Error Message	Enter Device Name	Passed
Blank Waiter Name	Waiter Name	Blank	Error Message	Enter Waiter Name	Passed
Duplicate Device Name	Device Name	Single click	The entire file path to be displayed in the text field	The entire file path gets displayed in the text field	Passed
Selecting Device Assigned to	Assigned to	Selecting Waiter	The Dropdown list should be filled with all Waiter Name	The Dropdown list filled with all Waiter Name	Passed
Selecting Device Assigned to	Assigned to	Selecting Table	The Dropdown list should be filled with all Table Name	The Dropdown list filled with all Table Name	Passed

Table 6.3 Test case for Device and Waiter Management

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
Blank Section Name	Section Name	Blank	Error Message	Enter Section Name	Passed
Blank Table Name	Table Name	Blank	Error Message	Enter Table Name	Passed
Displaying of Table Section wise	Selection of a Section from dropdown	Ground Floor	The Ground floor section view should be displayed	The Ground floor section view displayed with all the table in the section with their position	Passed

Table 6.4 Test case for Table and Section Management

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
Blank First Name	First Name	Blank	Error Message	Enter First Name	Passed
Blank Last Name	Last Name	Blank	Error Message	Enter Last Name	Passed
Blank Email Id	Email Id	Blank	Error Message	Enter Email	Passed
Validating Email Id	Email Id	xyz.com	Error message	Invalid Email Id	Passed
Blank Contact Number	Contact Number	Blank	Error Message	Enter Contact Number	Passed
Enter Alphabet in Contact Number	Contact Number	Blank	Error Message	Contact Number is invalid	Passed
Blank Id Type	Id Type	Blank	Error Message	Enter Id Type	Passed
Blank Id Code	Id Code	Blank	Error Message	Enter Id Code	Passed
Blank Password	Id Password	Blank	Error Message	Enter Password	Passed
Length of Password	Password	More than 15 characters or less than 8 characters	Error Message	Password should be of length 8 to 15 characters	Passed

Table 6.5 Test case for User Management

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
Blank Username	Username	Blank	Error Message	Enter Username	Passed
Blank Password	Password	Blank	Error Message	Enter Password	Passed
Length of Password	Password	More than 15 characters or less than 8 characters	Error Message	Password should be of length 8 to 15 characters	Passed

Table 6.6 Test case for Internet Access Activity

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
Blank Order	Order Items	Click on Order Button Without selecting a single item	Error Message	Select at least one item to order	Passed
Checkout without Order	Checkout Button	Click on Checkout without Ordering	Error Message	You can not checkout without placing an order	Passed

Table 6.7 Test case for Menu Activity

Test Case Description	Input name	Entered input	Required Output	Actual Output	Result
New Order	Order Details	Click on Order Button in an android device to place order	Live Screen should be updated	The respective Table node Blinks with Red color to indicate new Order	Passed
Order Approved /Seen	Table Node	Click on Respective Table Node	Live Screen should be updated	The respective Table node is assigned Yellow color to indicate Order is Seen and Approved	Passed
Order Served	Is Served	Select all Items as Served	Live Screen should be updated	The respective Table node is assigned Green color to indicate Order is Seen and Approved	Passed
Request for Checkout	Order Details	Click on Checkout	Live Screen should be	The respective	Passed

		Button in an android device to Checkout	updated	Table node Blinks with Yellow color to indicate Checkout Request	
Call Waiter	Table Name	Click on Call waiter Button in an android device to Call a waiter	Live Screen should be updated	The waiter list should blink with Red color to indicate Waiter Request	Passed

Table 6.8 Test case for Integration of Menu activity in android and Live Screen in Php

Test Case Description	**Input name**	**Entered input**	**Required Output**	**Actual Output**	**Result**
Blank Customer Name	Customer Name	Blank	Error Message	Enter Customer Name	Passed
Blank Address	Address	Blank	Error Message	Enter Address	Passed

Table 6.9 Test case for Generating Invoice

7.1. LIMITATIONS

Although the Restaurant POS System performs all required operations necessary for a restaurant but its functionality is limited to the industry defined requirements and all the major aspects are included in the package. However, it does not include the entire control over the orders of the restaurant. The system is limited by the set of functionalities provided and can be extended further in many directions as per specific requirements.

There are certain features which may extend the scope of the application. Some of the features are:

- Food Stock Management.
- Food Coupons management.
- Kitchen management.
- Custom Theme Generation by the manager/receptionist.

However the system totally depends on the network resources and can crash in case of a network crash. Network error handling may need the support of a network administrator.

The system provides elegant look and feel due to the power of highly attractive and powerful user interface components of android. However the system is developed in android Vr-2.3.3 which puts a limitation on the improving the look and feel of the application

7.2. FUTURE EXTENSIONS

Following are the future enhancements of our system which make it a stretchable prototype:

- **Kitchen Management**

  The system currently includes the interaction between the users and the manager but later on it can be extended to interact with the kitchen also. This would require some UI development which enables the kitchen chef to interact with the manager on every order he receives.

- **Food Stock Management**

  The system currently stores no record on the availability of food stock, which can be thought of as an extra feature to the system, and hence widening the product scope.

- **Food Coupons Management**

  The system can be further extended to incorporate a feature like food coupons processing for a particular order.

- **Custom Theme Management**

  The System contains 7 fixed themes which can set by the manager as per his wish. An addition to that can be, by providing the manager with a UI where he can customize the existing theme or he can also develop his own theme.

- **Different Payment facility**

  The System can be enhanced by including different payment methods, like adding credit card payment, online payment etc.

REFERENCES

1. Developers.android.com

-An excellent website for android application development, owned by the android developers. It consists of the complete api which was most helpful at the time of android development.

2. www.w3schools.com

-An excellent source of information on web technology and other supporting technologies(HTML,JavaScript,jQuery).

3. dev.mysql.com

-Details about the working and functions of MySQL database.

4. www.codeproject.com

-An excellent website giving small code snippets as the solution to doubts of the coders all over the world. The website contains a good collection of codes which are reliable and useful for study by new developers.

5. www.stackoverflow.com

- Gives instant solutions to queries related to any of the programming or analysis part.

6. www.php.net

- A very useful website developed by the PHP owners which describes each and every functionality of PHP.

EXPERIENCE

Today after end of Masters in Computer Application, since, we ventured into the idea of developing a **"RESTAURANT POS SYSTEM",** we are proud of all the sincere efforts and commitment that we have put into it as it's only our "hard work" that has eventually paid off to reach our goal.

One has to learn to work in a team. Throughout our career we would be interacting with lot many people on the professional front. These people will be having different beliefs, philosophies, principles, outlooks, visions and approaches.

One has to strike a balance to cope with the difference of opinion. Most difficult situation arises when more than one opinion comes out to be correct and the problem is which one to choose. At this point there has to be a consensus, proper discussion and reasoning to choose one of the available solutions.

There is another point to learn, regularly reporting to project guide at time to time, and from this we learned how to follow project plan milestone under guidance of our guide. Our **guide Mr. Upanish Doshi, Director of Samay Softwares** used to have project status discussion every week, during these periods we use to submit our report and discuss about the problems faced throughout the week, And also possible solution for those problems.

Overall, it was a nice learning experience. At times, it was tough. The knowledge and the information gained eventually surely made up for more than what we had to miss. There is a vast ocean of knowledge out there and we have taken our plunge. The learning curve was initially steep, but later it became less hazardous as we progressed.

All along we have been well and truly supported and encouraged by our teachers and friends and so they surely deserve our hearty thanks.

To conclude, Faculty of Science and Technology, M. S. University gave us a solid foundation on which we would be able to build a successful career ahead. This project has been a wonderful working experience and will become an everlasting memory.