**BAKERY MANAGEMENT SYSTEM**

**ABSTRACT**

An Bakery shop that allows users to check for various bakery products available at the online store and purchase online. The project consists of list of bakery products displayed in various categories. The user may browse through these items as per categories. If the user likes a product he may add it to his shopping cart. Once user wishes to checkout he must register on the site first. He can then login using same id password next time. Now he may pay through a credit card or cash on delivery. Once the user makes a successful transaction he gets a copy of the shopping receipt on his email id. Here we use user friendly interface to make the entire frontend. The middle tier or code behind model is designed for fast processing. And SQL serves as a backend to store bakery products lists data. Thus, the online Bakery shopping project brings an entire bakery shop online and makes it easy for both buyer and seller.

**1. INTRODUCTION**

The main goal of this project is an managing the bakery shop in systematically, we don’t need to bother or thinking about this project. Every one can easily to use this application,it’s an user friendly application, in this project cover overall process in bakery shop. The project is aimed to develop by **JAVA** as Front end and **MS SQL SERVER** as Back end. The back end is used to store the information in this system.

**1.1 SYSTEM SPECIFICATION**

**1.1.1 HARDWARE SPECFICATION:**

* Processor : P 4 700 GHz.
* RAM : 4GB RAM
* Hard Disk Drive : 40 GB HDD

**1.1.2 SOFTWARE SPECIFICATION:**

* + Operating System : Windows XP/7/8/10
  + Front End : JAVA
  + Back End : MY SQL

1. **SYSTEM STUDY**

**2.1 EXISTING SYSTEM:**

A manual system sometime may hard to find some difficulties. Which could be show some wrong data and very hard to managing the stores. So we can provide the some automatic system wise bakery management. Which have avoid some drawbacks.

**2.1.1 DRAWBACKS:**

The existing system has the following drawbacks.

* Very hard to calculate stock details
* Waste of paper works
* Billing details are manually calculated

**2.2 PROPOSED SYSTEM:**

This system is easy to managing the bakery shop. We no need to memories any product or purchase details. All can be managed by systematically. It could be very easy and attractive design. Every user can easy to use this system so we no need to study separate study for this project.

**2.2.1 FEATURES:**

* Very low cost and high efficiency
* Purchase and sales details are managed systematically.
* Save time

1. **SYSTEM DESIGN AND DEVELOPMENT**

**3.1 FILE DESIGN**

The selection of the file system design approach is done according to the needs of the developers what are the needed requirements and specifications for the new design. It allowed us to identify where our proposal fitted in with relation to current and past file system development. Our experience with file system development is limited so the research served to identify the different techniques that can be used. The variety of file systems encountered show what an active area of research file system development is. The file systems may be from one of the two fundamental categories. In one category, the file system is developed in user space and runs as a user process. Another file system may be developed in the kernel space and runs as a privileged process. Another one is the mixed approach in which we can take the advantages of both aforesaid approaches. Each development option has its own pros and cons. In this article, these design approaches are discussed.

**3.2 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
* It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
* will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**3.3 OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**3.4 DATABASE DESIGN**

Today's businesses depend on their databases to provide information essential for day-to-day operations, especially in case of electronic commerce businesses who has a definite advantage with up-to-date database access. Good design forms the foundation of any database, and experienced hands are required in the automation process to design for optimum and stable performance.

Software Solutions have been constantly working on these platforms and have attained a level of expertise. We apply proven methodologies to design, develop, integrate and implement database systems to attain its optimum level of performance and maximize security to meet the client's business model.

### Business needs addressed:

* Determine the basic objects about which the information is stored
* Determine the relationships between these groups of information and the objects
* Effectively manage data and create intelligent information
* Remote database administration or on site administrative support
* Database creation, management, and maintenance
* Information retrieval efficiency, remove data redundancy and ensure data security

**3.5 SYSTEM DEVELOPMENT**

**3.5.1 DESCRIPTION OF MODULES**

1. Customer Registration
2. Purchase Orders
3. Sales Orders
4. Billing Report
5. **Customer Registration**

This modules contains add the new customer details. It Includes are User id, name, date of birth, address, mobile number, e-mail id, password and so on.

1. **Purchase Orders:**

This modules contains are user purchase the item details maintained. It includes are item name, purchasing date, no of quantity, rate details and so on.

1. **Sales Order**

This module contains maintain the sales details, with the help of sales details we can generate the billing report. Once the order was purchased by customer the amount will be added to the billing report.

1. **Report:**

This Module Contains two sub Modules. They are a. User Report- user can view their past purchase history and b. Admin Report- Admin can view the user history and purchase history.

1. **SYSTEM TESTING AND IMPLEMENTATION**

**SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

**The objective of testing is as follows:**

* + Testing is the process of executing a program with the intent of finding an error.
  + A successful test is that one of the cover of undiscovered error.

### TESTING ISSUES

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**TESTING METHODOLOGIES**

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfills this objective a series of test step unit, integration. validation and system tests were planned and executed the test steps are:

**System Testing**

Testing is an important phase in project development. System testing makes a logical assumption that if all parts of the system are correct, and the goal will be achieved successfully. The software must meet the user specification and it must satisfy according to the needs of the users.

Testing is the process of executing a project within the intend of finding errors. A good test case is one that has a high probability of finding an undiscovered error.

**Unit Testing**

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

**In Project**, Each module such customer registration module, request module, employee details module, stock module, vehicle module and area detail modules are tested individually for example, Customer details module can contain the more forms to maintain the information so all forms could be tested like entered information store appropriately in database access page or not. If correctly accessed means the testing of registration module successfully completed. Likewise all modules are tested successfully.

**Integration Testing**

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

**In Project,** Integration of two modules can be tested together such as customer registration details and customer login module for verification purposes providing proper accessibility to users. The communication of Registration and Login module can test and executed successfully.

**Validation Testing**

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

**In this project,** Admin login details form Enter without username and password in textbox enter the submit button then Login failed message otherwise checks the both textbox value that is true means valid page displayed. Enter Password Displaying password character \*.if it displays the characters security is not availed so testing of software is failed.

**Output Testing**

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

**In project** All the forms are tested as it gives the necessary output to the user’s search such as view response details.

1. **CONCLUSION**

This project main objective goal of a shop manager can verify the product and sales details, we can easily found the stock details of the store which could be very helpful to managing the bakery shop.

**FUTURE ENHANCEMENT**

In this application we just focusing to maintain the purchase and sales, our future enhancement is that to purchasing the product item via online, which may very helpful for users. Customer no need come to shop and buying the product, customer just login and order their favourite food items. And also we can provide mobile application as well. Its have lot of module to use this application.

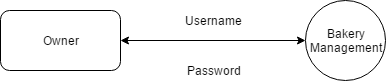
**BIBLIOGRAPHY**

* **CM96**  
  Peter Coad and Mark Mayfield, Java Design: Building Better Apps and Applets, Yourdon Press, 1996.
* **CH97**  
  Gary Cornell and Cay S. Horstmann, Core Java, second ed., SunSoft Press, 1997
* **Gea99b**  
  David M. Geary Graphic Java 2: Mastering the JFC, vol. II, Swing, third ed., Sun Microsystems Press, 1999.
  + **Gea99c**  
    David M. Geary Graphic Java 2: Mastering the JFC, vol. III, Advanced Swing, third ed., Sun Microsystems Press, 1999(?).

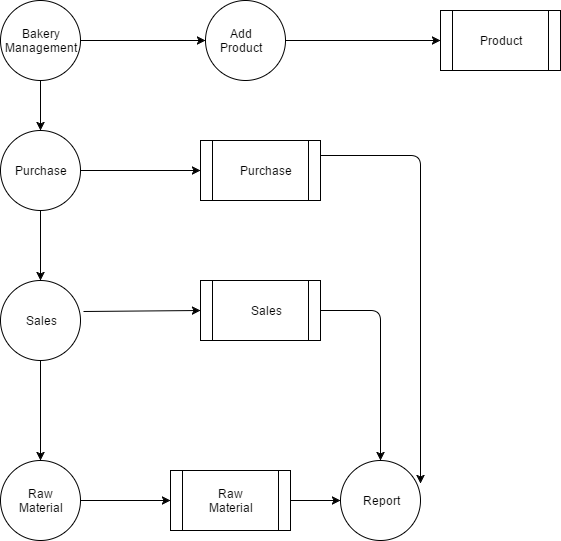
**APPENDICES**

1. **DATA FLOW DIAGRAM**

LEVEL 0:



LEVEL 1:

****

1. **TABLE STRUCTURE**

**TABLE NAME : ADMIN**

**PRIMARY\_KEY : ID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Id | Int |  | Admin id |
| Username | Varchar | 30 | Admin username |
| password | Varchar | 30 | Admin password |

**TABLE NAME : CUSTOMER**

**PRIMARY KEY : cid**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Cid | Int | 10 | Customer id |
| Name | Varchar | 30 | Customer name |
| Address | Varchar | 30 | Address |
| Contactno | Varchar | 10 | Contact number |
| Gender | Varchar | 10 | Gender |
| Email | Varchar | 10 | eamil |

**TABLE NAME : PRODUCT**

**PRIMARY KEY : product\_id**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Product\_id | Int | 10 | Product id |
| Company | Varchar | 10 | Company name |
| Model | Varchar | 30 | Model name |
| Price | Int | 10 | price |

**TABLE NAME : PURCHASE**

**PRIMARY KEY : PID**

**FOREIGN KEY : PROID,MID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Pid | Int | 10 | Product id |
| Mid | Int | 10 | Model id |
| Mname | Varchar | 10 | Model name |
| Quantity | Int | 10 | quantity |

**TABLE NAME : SALES**

**PRIMARY KEY : SID**

**FOREIGN KEY : PROID,MID,UID**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATA TYPE** | **SIZE** | **DESCRIPTION** |
| Pid | Int | 10 | Product id |
| Uid | Int | 10 | User id |
| Mid | Int | 10 | Model id |
| Mname | Varchar | 10 | Model name |
| Quantity | Int | 10 | quantity |

1. **SAMPLE CODING**

<!DOCTYPE html>

<html>

<head>

<title>Bakery Management</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" >

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.1/css/all.css">

<link rel="stylesheet" href="/home/login.css" >

</head>

<!--Coded with love by Mutiullah Samim-->

<body>

<div class="container h-100">

<div class="d-flex justify-content-center h-100">

<div class="user\_card">

<div class="d-flex justify-content-center">

<div class="brand\_logo\_container">

<img src="/home/banana\_chocolate\_cake.jpg" class="brand\_logo" alt="Logo">

</div>

</div>

<div class="d-flex justify-content-center form\_container">

<form>

<div class="input-group mb-3">

<div class="input-group-append">

<span class="input-group-text"><i class="fas fa-user"></i></span>

</div>

<input id="username" type="text" name="" class="form-control input\_user" value="" placeholder="username">

</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text"><i class="fas fa-key"></i></span>

</div>

<input id="password" type="password" name="" class="form-control input\_pass" value="" placeholder="password">

</div>

<div class="form-group">

<div class="custom-control custom-checkbox">

<input type="checkbox" class="custom-control-input" id="customControlInline">

</div>

</div>

</form>

</div>

<div class="d-flex justify-content-center mt-3 login\_container">

<button id="login" type="button" name="button" class="btn login\_btn">Login</button>

</div>

</div>

</div>

</div>

</div>

</body>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="/home/login.js"></script>

</html><html>

<head>

<link href="/bootstrap/bootstrap.css" rel="stylesheet" />

<link href="/bootstrap/datatables.css" rel="stylesheet" />

<link href="https://netdna.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">

<link rel="stylesheet" type="text/css" href="/home/home.css">

</head>

<body>

<div class="row">

<div class="side-menu">

<nav class="navbar navbar-default" role="navigation">

<!-- Brand and toggle get grouped for better mobile display -->

<div class="navbar-header">

<div class="brand-wrapper">

<!-- Brand -->

<div class="brand-name-wrapper">

<a class="navbar-brand" href="#">

Brand

</a>

</div>

</div>

</div>

<!-- Main Menu -->

<div class="side-menu-container">

<ul class="nav navbar-nav">

<li><a href="#htab1" data-toggle="tab"><span class="glyphicon glyphicon-send"></span> ADD

CUSTOMER</a></li>

<li><a href="#htab2" data-toggle="tab"><span class="glyphicon glyphicon-send"></span> ADD

PURCHASE</a></li>

<li><a href="#htab3" data-toggle="tab"><span class="glyphicon glyphicon-send"></span> SALES</a>

<li><a href="#htab4" data-toggle="tab"><span class="glyphicon glyphicon-send"></span> STOCK

DETAILS</a>

<li><a href="/" ><span class="glyphicon glyphicon-send"></span> LOGOUT</a>

</li>

</ul>

</div>

<!-- /.navbar-collapse -->

</nav>

</div>

<!-- Main Content -->

<div class="container-fluid">

<div class="side-body">

<div class="tab-content" style="margin: 30px">

<div role="tabpanel" class="tab-pane fade in active" id="htab1">

<h1>ADD CUSTOMER</h1>

<div class="col-md-3">

<div class="form-group">

<label>Name:</label>

<input type="text" class="form-control" id="cus\_name">

</div>

<div class="form-group">

<label>Mobile:</label>

<input type="text" class="form-control" id="cus\_mobile">

</div>

<div style="display: none" class="form-group">

<label>Age:</label>

<input type="text" class="form-control" id="cus\_age">

</div>

<div class="form-group">

<label>Address:</label>

<input type="textarea" class="form-control" id="cus\_address">

</div>

<div class="form-group">

<label>Pincode:</label>

<input type="textarea" class="form-control" id="cus\_pincode">

</div>

<button id="add\_customer" type="submit" class="btn btn-primary">ADD CUSTOMER</button>

</div>

<div class="col-md-5">

<table id="stock\_table" class="table table-striped table-bordered table-sm" cellspacing="0"

width="100%">

<thead>

<tr>

<th class="th-sm">SNO

</th>

<th class="th-sm">NAME

</th>

<th class="th-sm">MOBILE

</th>

<th style="display: none" class="th-sm">AGE

</th>

<th class="th-sm">ADDRESS

</th>

<th class="th-sm">PINCODE

</th>

</tr>

</thead>

<tbody id="cus\_list\_body">

</tfoot>

</table>

</div>

</div>

<div role="tabpanel" class="tab-pane fade" id="htab2">

<h1>ADD PURCHASE</h1>

<div class="col-md-3">

<div class="form-group">

<label>PID:</label>

<input id="pid" type="text" class="form-control">

</div>

<div class="form-group">

<label>PNAME:</label>

<input id="pname" type="text" class="form-control">

</div>

<div class="form-group">

<label>PRICE:</label>

<input id="pprice" type="text" class="form-control">

</div>

<div class="form-group">

<label>QUANTITY:</label>

<input id="pquantity" type="textarea" class="form-control">

</div>

<div class="form-group">

<label>BRAND:</label>

<input id="pbrand" type="textarea" class="form-control">

</div>

<div class="form-group">

<label>EXPIERY DATE:</label>

<input id="pexpiery" type="textarea" class="form-control">

</div>

<button id="add\_purchase" type="submit" class="btn btn-primary">ADD PRODUCT</button>

</div>

<div class="col-md-5">

<table id="purchase\_table" class="table table-striped table-bordered table-sm" cellspacing="0"

width="100%">

<thead>

<tr>

<th class="th-sm">PID

</th>

<th class="th-sm">PNAME

</th>

<th class="th-sm">PRICE

</th>

<th class="th-sm">QUANTITY

</th>

<th class="th-sm">BRAND

</th>

<th class="th-sm">EXPIERY DATE

</th>

</tr>

</thead>

<tbody id="pur\_list\_body">

</tfoot>

</table>

</div>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab3">

<h1>SALES</h1>

<div class="col-md-3">

<div class="form-group">

<label>PRODUCT ID:</label>

<input id="spid" type="text" class="form-control">

</div>

<div class="form-group">

<label>CUSTOMER MOBILE NUMBER:</label>

<input id="smobile" type="text" class="form-control">

</div>

<div class="form-group">

<label>PNAME:</label>

<input id="sname" type="text" class="form-control" disabled>

</div>

<div class="form-group">

<label>PRICE:</label>

<input id="sprice" type="text" class="form-control" disabled>

</div>

<div class="form-group">

<label>QUANTITY:</label>

<input id="squantity" type="textarea" class="form-control">

</div>

<div class="form-group">

<label>AMOUNT:</label>

<input id="samount" type="textarea" class="form-control" DISABLED>

</div>

<div class="form-group">

<label>BRAND:</label>

<input id="sbrand" type="textarea" class="form-control" disabled>

</div>

<div class="form-group">

<label>EXPIERY DATE:</label>

<input id="sexpiery" type="textarea" class="form-control" disabled>

</div>

<button id="add\_sales" type="submit" class="btn btn-primary">ADD SALES</button>

</div>

<div class="col-md-5">

<table id="addcustomer\_table1" class="table table-striped table-bordered table-sm"

cellspacing="0" width="100%">

<thead>

<tr>

<th class="th-sm">PID

</th>

<th class="th-sm">PNAME

</th>

<th class="th-sm">PRICE

</th>

<th class="th-sm">QUANTITY

</th>

<th class="th-sm">AMOUNT

</th>

<th class="th-sm">BRAND

</th>

<th class="th-sm">EXPIERY DATE

</th>

</tr>

</thead>

<tbody id="body\_sal">

</tfoot>

</table>

</div>

</div>

<div role="tabpanel" class="tab-pane fade in" id="htab4">

<h1>STOCK DETAILS</h1>

<div class="col-md-12">

<table id="addcustomer\_table" class="table table-striped table-bordered table-sm"

cellspacing="0" width="100%">

<thead>

<tr>

<th class="th-sm">PID

</th>

<th class="th-sm">PNAME

</th>

<th class="th-sm">QUANTITY

</th>

<th class="th-sm">BRAND

</th>

<th class="th-sm">EXPIERY DATE

</th>

</tr>

</thead>

<tbody id="stock\_body">

</tfoot>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<script src="/bootstrap/jquery.min.js"></script>

<script src="/home/home.js"></script>

<script src="/bootstrap/datatable.min.js"></script>

<script src="/bootstrap/datatable.bootstrap.min.js"></script>

<script src="/bootstrap/bootstrap.min.js"></script>

</html>

1. **SAMPLE INPUT**
2. **SAMPLE OUTPUT**