

Development of a feature-rich, **O**NLINE **R**ESTAURANT **S**YSTEM

A Project Report for Summer/Winter Industrial Training

Submitted by

Mayuraksha Sikdar

Subha Prasad Dey

Debajyoti Ghosh

in partial fulfillment for the award of the degree of

B. Tech

in

Computer Science & Engineering

Narula Institute of Technology



At

Ogma TechLab Pvt. Ltd.



Ogma TechLab Pvt. Ltd.



BONAFIDE CERTIFICATE

Certified that this project work was carried out under my supervision

“Development of a feature-rich, Online Restaurant System” is the bonafide work of

Name of the student: Mayuraksha Sikdar

Signature:

Name of the student: Subha Prasad Dey

Signature:

Name of the student: Debajyoti Ghosh

Signature:

SIGNATURE

Name : Saumitra Das
PROJECT MENTOR

SIGNATURE

Ogma TechLab Original Seal

Acknowledgement

I take this opportunity to express my deep gratitude and sincerest thank to my project mentor, **SAUMITRA DAS** for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

I will like to give a special mention to my colleagues. Last but not the least I am grateful to all the faculty members of Ogma TechLab Pvt. Ltd. or their support.

1. Abstract	5
2. Introduction and Objectives of the Project	5
3. Project Category (RDBMS/OOPS/Networking/Multimedia/Artificial Intelligence/Expert Systems etc.)	5
4. Tools/Platform, Hardware and Software Requirement specifications	5
5. Goals of Implementation	6
6. SDLC Process Applied	6
7. Data Model	8
8. Functional Requirements (Use Case Diagram)	9
9. System Requirements	12
10. Feasibility Study	14
11. Project Planning	16
12. Project Scheduling	17
13. Software Engineering Paradigm applied	17
a. Data Flow Diagram (DFD)	17
14. Database design	20
15. User Interface Design	22
16. Coding	33
17. Testing	33
18. System Security measures	38
19. Database/Data security	38
20. Creation of User profiles and access rights	38
21. Cost Estimation of the Project along with Cost Estimation Model	39
22. Future scope and further enhancement of the Project	40
23. Bibliography	41
24. Appendix	42

1. Abstract

This project aims to develop a web application is for Online Restaurant System. This is an online restaurant where multiple users can view and information of the foods.

This project, once implemented, will enable people to get access to an online restaurant. The main object of this application is to make it interactive and its ease of use. It would make searching, viewing and search of a food easier. It contains a sophisticated search engine for users to search for products specific to their needs. The search engine provides an easy and convenient way to search for foods, where a user can search for a product interactively and the search engine would refine the different foods available based on the user's input.

2. Introduction and Objectives of the Project

Online Restaurant System is the system for manage the restaurant business. The main point of developing this system is to help restaurant administrator manage the restaurant business and help customer for online ordering and reserve table. The project is developing because; many restaurants have a lot difficult to manage the business such as customer ordering and reservation table. By using manual customer ordering is difficult to waiter keep the correct customer information and maybe loss the customer information. So, online restaurant management system will develop to help the restaurant administrator to manage restaurant management and for customer make their online ordering and reservation table. Other than that, this project is to upgrade the manual system and make the business easily to access and systematic

3. Project Category:

Web Application.

4.Tools/Platform, Hardware and Software Requirement specifications.

Tools

Dreamweaver CS6

Dia tool

MS-Office

Platform

Microsoft Windows 7/8/10

Hardware Requirement Specification

Client Machine		Server Machine	
HDD	200 MB	HDD	320 GB
Processor	Pentium 4 or newer processor that supports SSE2	Processor	Dual Core or newer processor
Memory	512 MB	Memory	2 GB

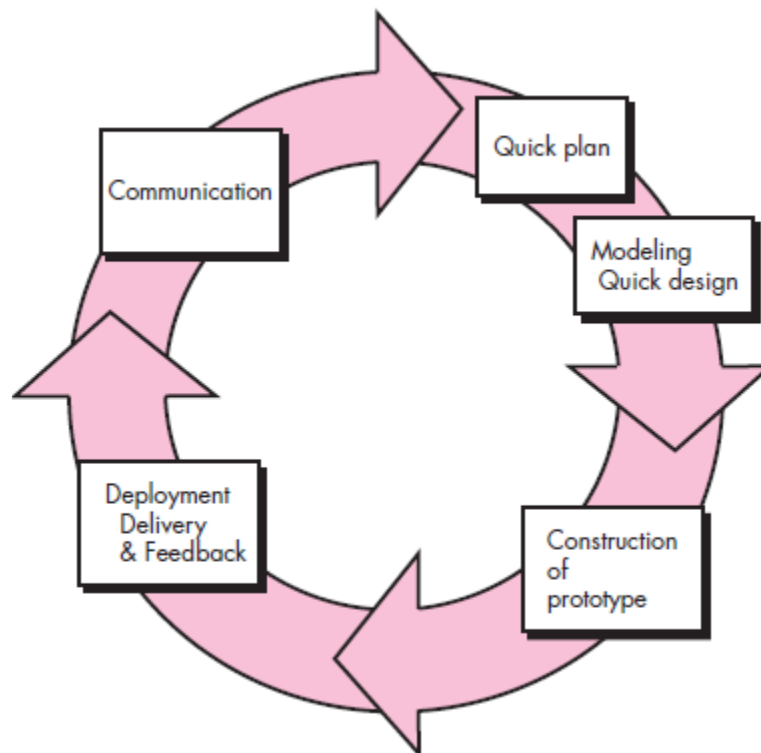
Software Requirement Specification

Client Machine		Server Machine	
Browser	Any standard browser with JavaScript interpreter	Software	Apache
Client side markup / scripting languages	HTML, JavaScript	Database Management System Software	MySQL
		Specification	MySQL-4.1

5. Goals of Implementation

The implementation aims at seamless document sharing across the institution.

6. SDLC Process Applied



Often, a customer defines a set of general objectives for software but does not identify detailed input, processing, or output requirements. In other cases, the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human/machine interaction should take. In these, and many other situations, a prototyping paradigm may offer the best approach.

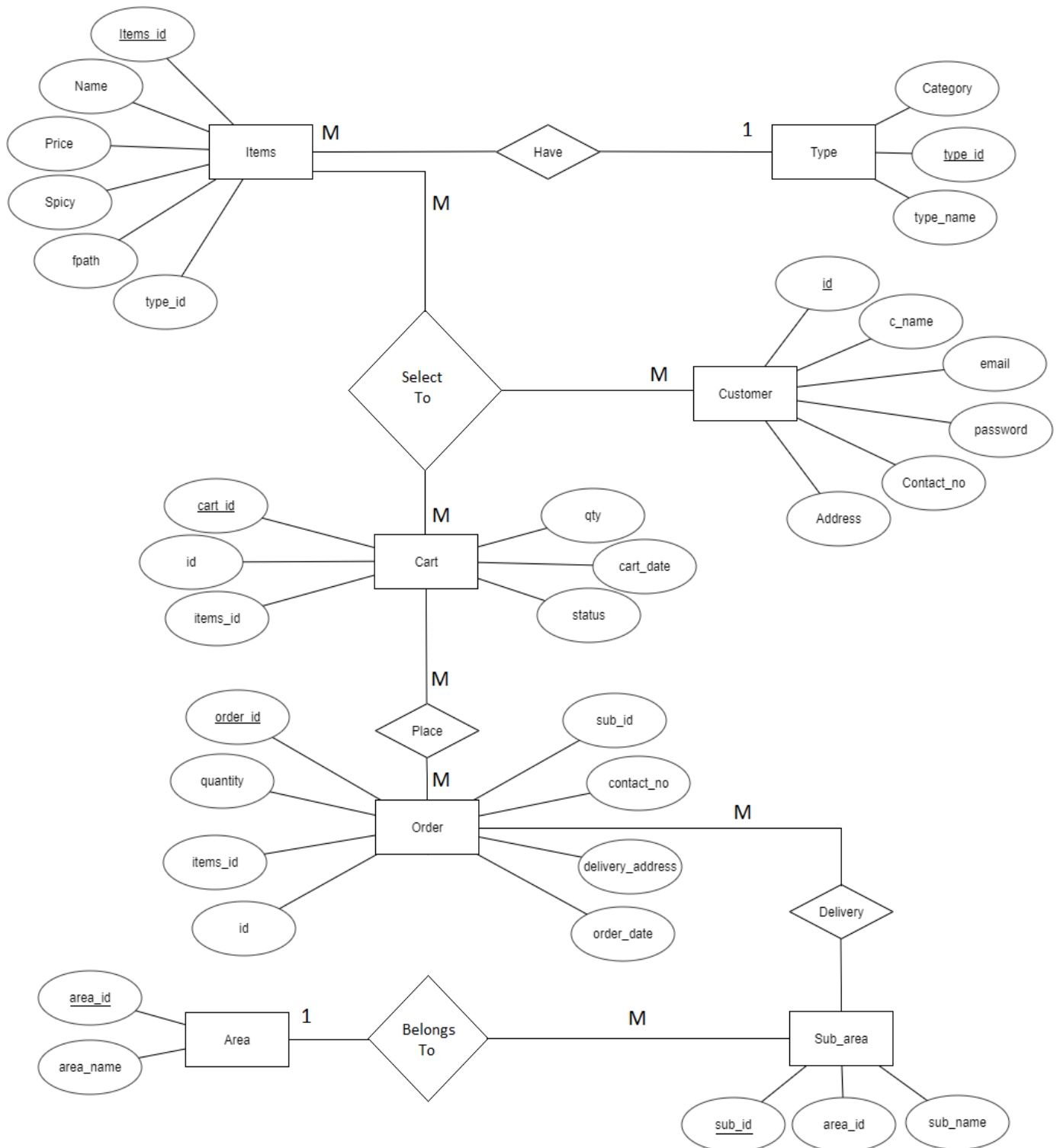
The prototyping paradigm begins with **requirements gathering**. Developer and customer meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory. A "**quick design**" then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user (e.g., input approaches and output formats). The quick design leads to the construction of a prototype. The prototype is evaluated by the customer/user and used to refine requirements for the software to be developed. Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

Ideally, the prototype serves as a mechanism for identifying software requirements. If a working prototype is built, the developer attempts to use existing program fragments or

applies tools (e.g., report generators, window managers) that enable working programs to be generated quickly.

7. Data Model

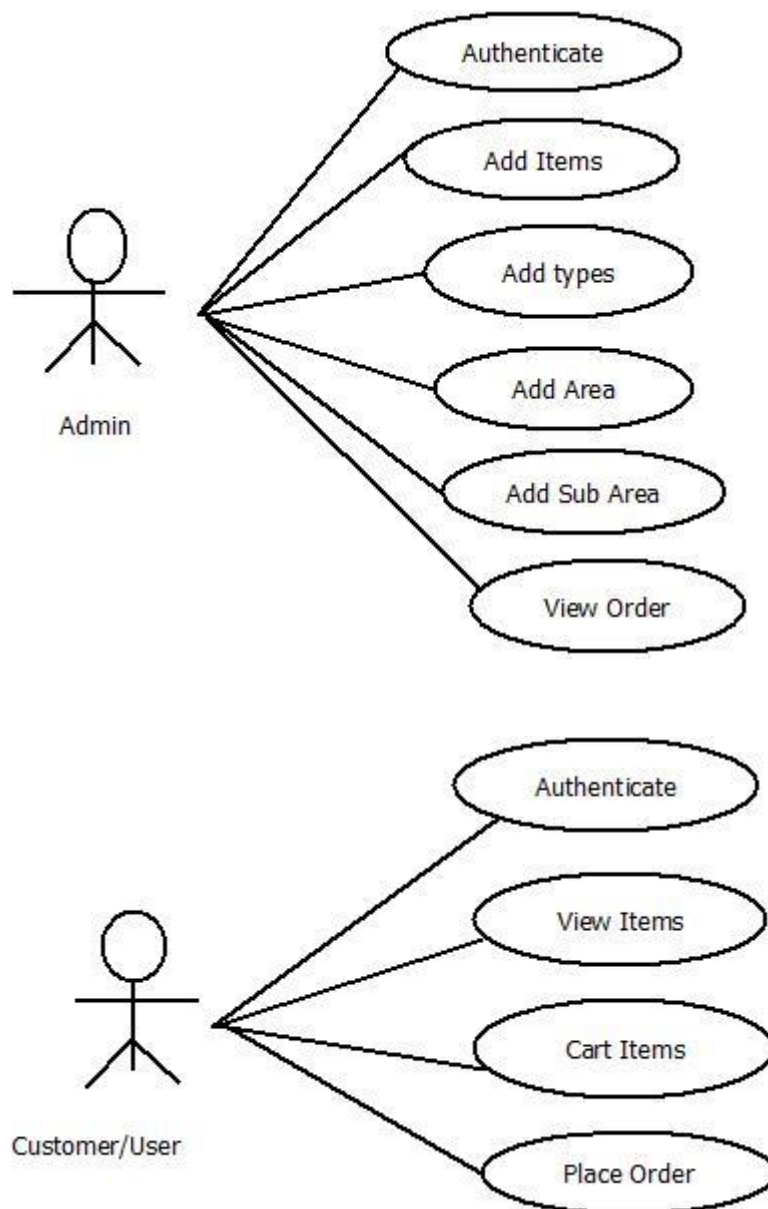
ER Diagram



8. Functional Requirements

Functional Requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Nonfunctional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately.

Use Case Diagram



Use Case Descriptions

Use Case Name:	Authentication
Priority	Essential
Trigger	Menu selection
Precondition	User is connected to the Internet and on the Restaurant home page
Basic Path	<ol style="list-style-type: none"> 1. User enters username and password. 2. The username and password is matched with the record in the database. 3. If the authentication parameters are correct the user is directed to the user's main page, otherwise an error message is displayed.
Alternate Path	NA
Post Condition	The user is on the User Home Page
Exception Path	If there is a connection failure the server returns to the wait state
Use Case Name:	Update Details
Priority	Essential
Trigger	Menu selection
Precondition	User is connected to the Internet and on the main page
Basic Path	<ol style="list-style-type: none"> 1. User enters values for all the fields. 2. User clicks on the submit button. 3. The server side program processes the request and returns the User registration number.
Alternate Path	NA
Post Condition	The User registers himself/herself with the Restaurant
Exception Path	If there is a connection failure the server returns to the wait state

Use Case Name:	Search Food
Priority	Essential
Trigger	Menu selection
Precondition	Admin is connected to the Internet and on the admin's main page
Basic Path	<ol style="list-style-type: none"> 1. Admin enter registration number of a user 2. The admin data appears 3. The Admin write his food name and then clicks on the search button
Alternate Path	NA
Post Condition	The admin can see all foods.
Exception Path	If there is a connection failure the server returns to the wait state
Use Case Name:	Add Food Details
Priority	Essential
Trigger	Menu selection
Precondition	Admin is connected to the Internet and on the admin's main page
Basic Path	<ol style="list-style-type: none"> 1. The Admin gives food Details 2. The details of the foods is available when a link from the link is clicked
Alternate Path	NA
Post Condition	The admin searches by food name
Exception Path	If there is a connection failure the server returns to the wait state

Use Case Name:	Update food Details
Priority	Essential
Trigger	Menu selection

Precondition	User is connected to the Internet and on the main page
Basic Path	4. Admin enters values for all the fields. 5. Admin clicks on the submit button. 6. The server side program processes the request and returns the User registration number.
Alternate Path	NA
Post Condition	The Admin update his/her foods with the Food Store cell
Exception Path	If there is a connection failure the server returns to the wait state

9. Non Functional Requirements

In addition to the obvious features and functions that you will provide in your system, there are other requirements that don't actually DO anything, but are important characteristics nevertheless. These are called "non-functional requirements" or sometimes "Quality Attributes." For example, attributes such as performance, security, usability, compatibility. aren't a "feature" of the system, but are a required characteristic. You can't write a specific line of code to implement them; rather they are "emergent" properties that arise from the entire solution. The specification needs to describe any such attributes the customer requires. You must decide the kind of requirements that apply to your project and include those that are appropriate.

Each requirement is simply stated in English. Each requirement must be objective and quantifiable; there must be some measurable way to assess whether the requirement has been met.

Often deciding on quality attributes requires making tradeoffs, e.g., between performance and maintainability. In the APPENDIX you must include an engineering analysis of any significant decisions regarding tradeoffs between competing attributes.

Here are some examples of non-functional requirements:

Performance requirements

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance. For better performance the application will restrict the document size to 5 MB.

Operating constraints

List any run-time constraints. This could include system resources, people, needed software, The application must run without any manual intervention.

Platform constraints

Discuss the target platform. Be as specific or general as the user requires. If the user doesn't care, there are still platform constraints.

Since the application will be developed in PHP it is platform independent.

Accuracy and Precision

Requirements about the accuracy and precision of the data. (Do you know the difference?) Beware of 100% requirements; they often cost too much.

Modifiability

Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person- months).

Minimal

Portability

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

Minimal

Reliability

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

Security

One or more requirements about protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, time, ...) to break into the system. Do not discuss solutions (e.g. passwords) in a requirements document.

Only secured users can access the application.

No one can go to any independent page without logging in.

Usability

Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

Legal

There may be legal issues involving privacy of information, intellectual property rights, export of restricted technologies, etc.

10. Feasibility Study

You should provide a feasibility report in the following format:

- **Product:** A general statement of the product; give a brief description of what the proposed system will do, highlighting where the proposed system meets the specified business requirements of the organization.
- **Technical Feasibility:** Will the proposed system perform to the required specification? Outline technical systems options you propose to use, which will give a technical solution satisfying the requirements and constraints of the system, as outlined in the terms of reference.
- **Social Feasibility:** Consideration of whether the proposed system would prove acceptable to the people who would be affected by its introduction. Describe the effect on users from the introduction of the new system; consider whether there will be a need for retraining the workforce. Will there be a need for relocation of some of the workforce? Will some jobs become deskilled? Will the current workforce be able to perform effectively any new tasks introduced by the proposed system? Describe how you propose to ensure user co-operation before changes are introduced.

- **Economic Feasibility:** Consider the cost/benefits of the proposed system. Detail the costs that will be incurred by the organization adopting the new system; consider development costs and running costs. Detail benefits that the new system will bring, direct economic benefits such as reduced costs, and indirect benefits, such as improved management information and better customer service. Illustrate the cost/benefit of the new system by applying a suitable cost/benefit analysis method such as the payback method.
- **Market Research:** A comprehensive market research identifying a need for the product. Detail all market research you carried out, listing sources of information. Justify any conclusions you have drawn from your research. Identify the potential customer base for your product, together with evidence of customer need for the product. Describe how you propose to compete with similar products on the market.
- **Alternative Solution:** Consideration of alternative solutions should be documented. At least two alternative business or technical systems options should be considered. Detail the differences between these options and the proposed system. Justify your choice of the proposed system and the reasons for rejecting the alternative options.

At this point, all of the planning for the project has been done and if the feasibility study has shown that the project is likely to succeed within its constraints, then it only remains for us to start the requirements analysis and thus proceed with the project.

Feasibility study

You should provide a feasibility report in the following format:

- **Product:** A general statement of the product; give a brief description of what the proposed system will do, highlighting where the proposed system meets the specified business requirements of the organization.
- **Technical Feasibility:** Will the proposed system perform to the required specification? Outline technical systems options you propose to use, which will give a technical solution satisfying the requirements and constraints of the system, as outlined in the terms of reference.

- **Social Feasibility:** Consideration of whether the proposed system would prove acceptable to the people who would be affected by its introduction. Describe the effect on users from the introduction of the new system; consider whether there will be a need for retraining the workforce. Will there be a need for relocation of some of the workforce? Will some jobs become deskilled? Will the current workforce be able to perform effectively any new tasks introduced by the proposed system? Describe how you propose to ensure user co-operation before changes are introduced.
- **Economic Feasibility:** Consider the cost/benefits of the proposed system. Detail the costs that will be incurred by the organization adopting the new system; consider development costs and running costs. Detail benefits that the new system will bring, direct economic benefits such as reduced costs, and indirect benefits, such as improved management information and better customer service. Illustrate the cost/benefit of the new system by applying a suitable cost/benefit analysis method such as the payback method.
- **Market Research:** A comprehensive market research identifying a need for the product. Detail all market research you carried out, listing sources of information. Justify any conclusions you have drawn from your research. Identify the potential customer base for your product, together with evidence of customer need for the product. Describe how you propose to compete with similar products on the market.

Alternative Solution: Consideration of alternative solutions should be documented. At least two alternative business or technical systems options should be considered. Detail the differences between these options and the proposed system. Justify your choice of the proposed system and the reasons for rejecting the alternative options. At this point, all of the planning for the project has been done and if the feasibility study has shown that the project is likely to succeed within its constraints, then it only remains for us to start the requirements analysis and thus proceed with the project

11. Project Planning

Project planning is concerned with identifying the following for every project:

- Activities
- Milestones

- Deliverables.

A plan must be drawn up to guide the development towards the project goal. A plan is drawn up at the start of a project. This plan should be used as the driver for the project. The initial plan is not static, and must be modified as the project progresses.

Planning is required for development activities from specification through to delivery of the system.

12. Project Scheduling

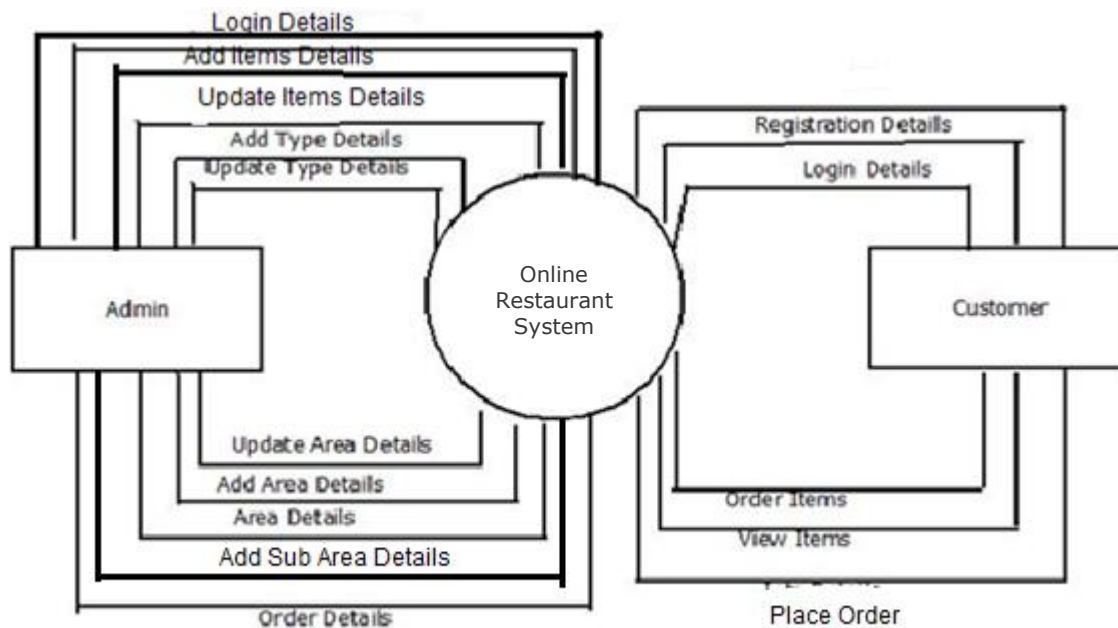
GANTT chart

Task	Person(s) Responsible	Week 1	Week 2	Week 3	Week 4
Communication					
Quick Plan					
Modeling Quick Design					
Construction of Prototype					
Deployment, Delivery and Feedback					

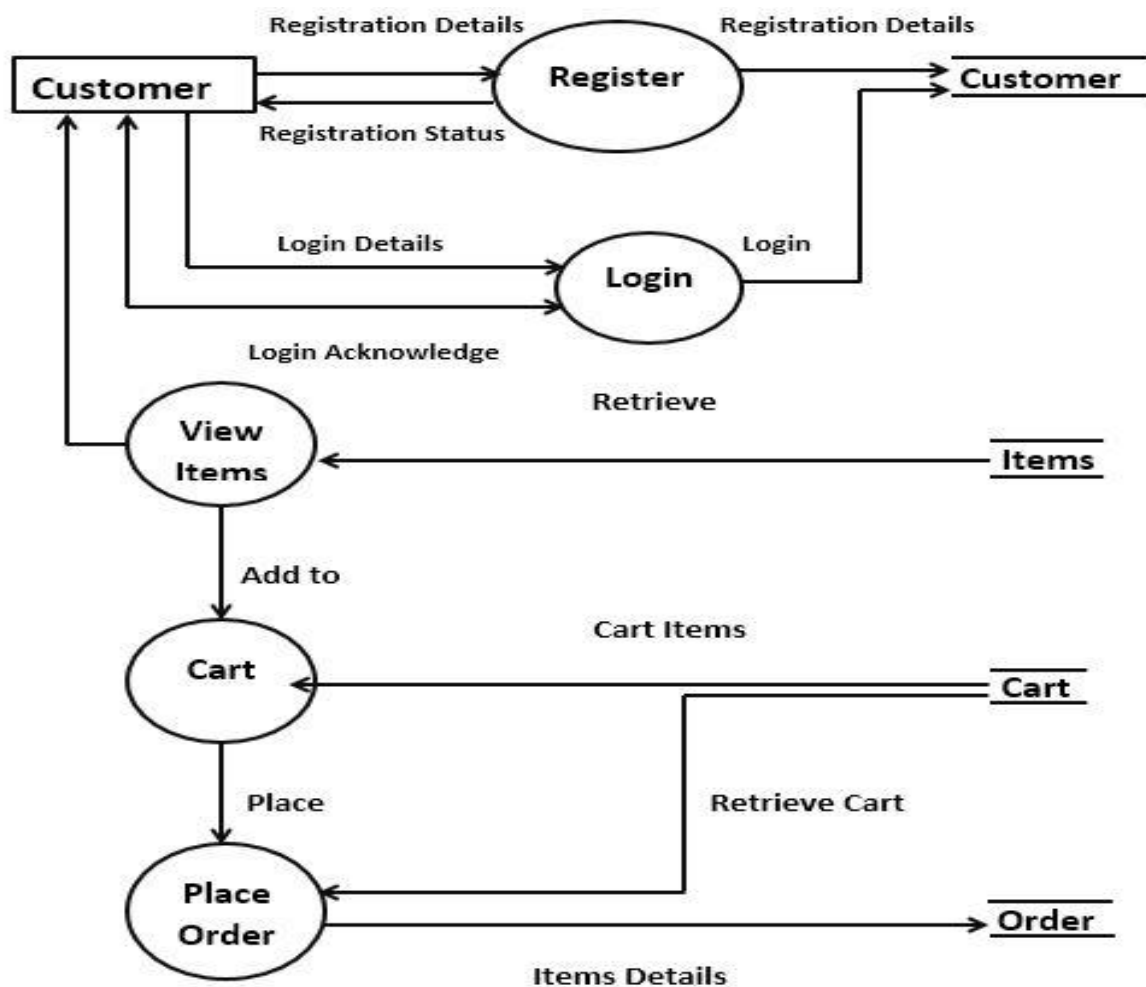
13. Software Engineering Paradigm Applied

Data Flow Diagrams

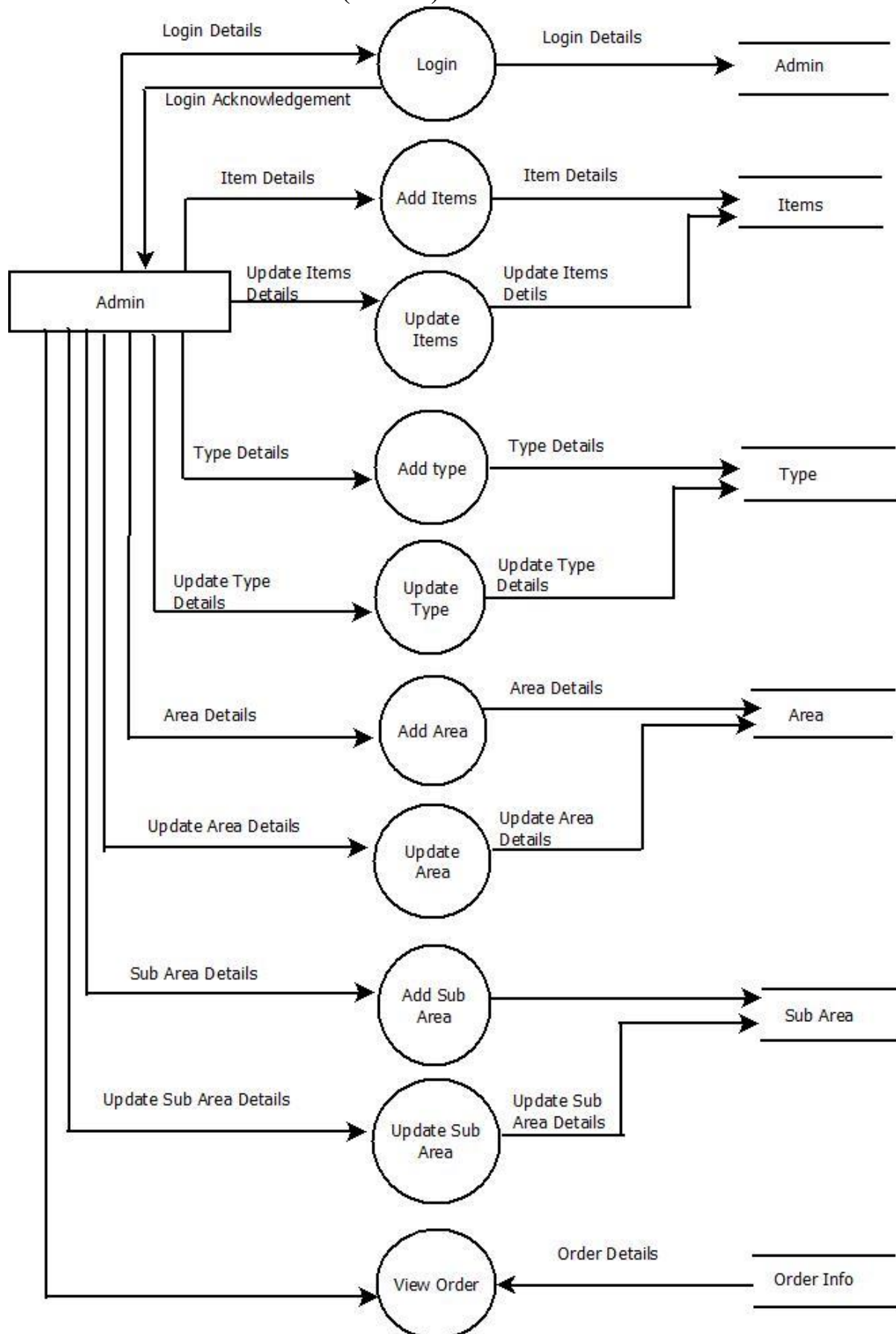
Level 0



‘1’ LEVEL DIAGRAM FOR (CUSTOMER)



'1' LEVEL DIAGRAM FOR (ADMIN)



14. Schema/Database Design

Customer Table

Name of Attributes	Data type	Key
Id	INT	Primary
c_name	VARCHAR	
email	VARCHAR	
password	VARCHAR	
Contact_no	VARCHAR	
address	VARCHAR	

Order Table

Name of Attributes	Data type	Key
order_id	INT	Primary
quantity	INT	
Items_id	INT	Foreign (Items)
Id	INT	Foreign (Customer)
Sub_id	INT	Foreign (Sub_area)
Contact_no	VARCHAR	
delivery_address	VARCHAR	
order_date	Date	

Admin Table

Name of Attributes	Data type	Key
Aid	INT	Primary
name	VARCHAR	
username	VARCHAR	
pass	VARCHAR	

Items Table

Name of Attributes	Data type	Key
Items_id	INT	Primary
Name	VARCHAR	
Spicy	VARCHAR	
Price	VARCHAR	
Fpath	VARCHAR	
type_id	INT	Foreign

Area Table

Name of Attributes	Data type	Key
area_id	INT	Primary
area_name	VARCHAR	

Sub_area Table

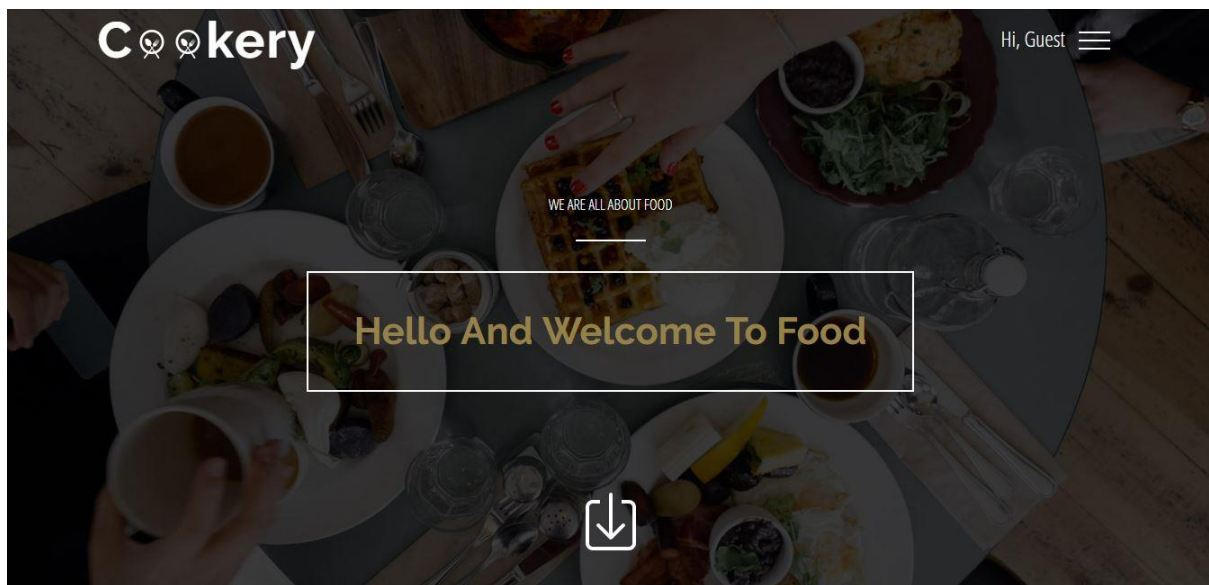
Name of Attributes	Data type	Key
sub_id	INT	Primary
Area_id	INT	Foreign
Sub_name	VARCHAR	

cart_table

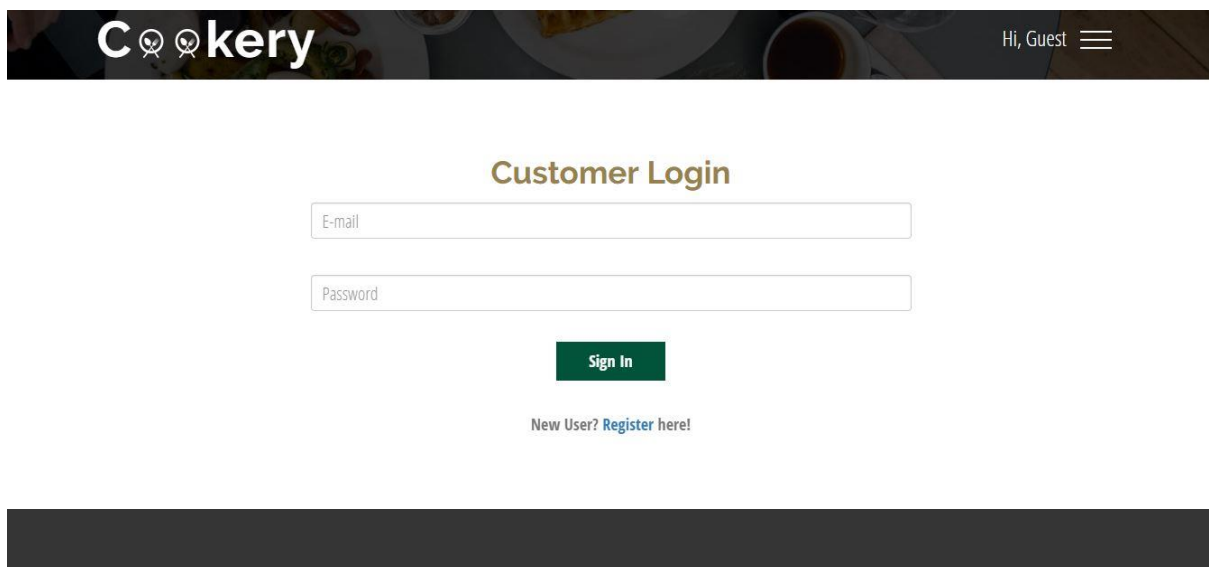
Name of Attributes	Data type	Key
cart_id	INT	Primary
id	INT	Foreign
items_id	INT	Foreign
qty	INT	
cart_date	DATE	
status	ENUM	

15. User Interface Design:

Home Page



Customer Login &Registration



Customer Registration

Menu Page

Menu

Starters Main Course[Non-Veg] Rice-Roti Main Course[Veg] Dessert

There are many variations



Chicken Cutlet [1pc]

Rs.30



Chicken Fingers [6pcs]

Rs.100



Chicken Satay [6pcs]

Rs.150

Food Details



Starters

There are many variations



Chicken Fingers [6pcs]

Spicy : Low

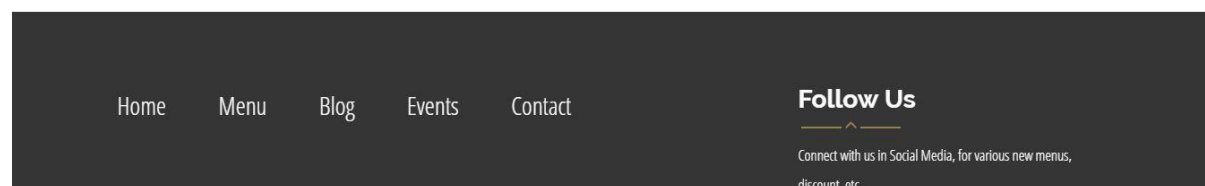
Price : 100

Quantity [Add to Cart](#)

Cart Item




Product	Name	Price	Quantity	Subtotal	
	Chicken Fingers [6pcs]	100	<input type="text" value="1"/>	100	
Continue Shopping				Total 100	Checkout



View Order



Product	Name	Price	Quantity	Total
	Chicken Satay [6pcs]	150	1	150

[← Continue Shopping](#)

Place Order



Delivery Details

Checkout



Order Details

Billing Address :

Sodepur
Area - Phoolbagan
Sub Area - Alochaya

Delivery Address :

Sodepur
Area - Phoolbagan
Sub Area - Alochaya

Order Details

Total Shopping	150.00
Delivery Charge	30.00
(CGST+SGST) 18%	7.50
Total	187.50

Place Order



Thank You !

Your order has been taken.
Please wait while we prepare your order.



Contact Us

Contact

Name	Phone number
<input type="text" value="Subha Prasad Dey"/>	<input type="text" value="7059931598"/>
Email	Address
<input type="text" value="antudey.2014.ad@gmail.com"/>	<input type="text" value="Sodepur"/>
MESSAGE	
<input type="text" value="Enter Message Here..."/>	
<input type="button" value="Send"/>	

Admin Login

Administration Login

Email address

Password


Admin Profile

Admin Panel

- Dashboard
- Order
- Category
- Items
- Area
- Deliver Locations

Welcome Subha Logout

Dashboard



Copyright © Indian Restaurant

Ordered Food

Admin Panel

- Dashboard
- Order
- Category
- Items
- Area
- Deliver Locations

Dashboard / Order Page

Order

Order ID	Quantity	Item ID	ID	Sub ID	Contact No.	Delivery Address	Order Date
123	2	111	1	1112	8100132483	sodepur	2018-08-17
1234	3	1111	10	11112	8100132400	kolkata	2018-08-15

Copyright © Indian Restaurant

All Category

Admin Panel

Dashboard

Order

Category

Items

Area

Deliver Locations

Welcome Subha

Logout

Dashboard / Category

All Category

Add New Category

Type	Category	Update	Delete
Veg	Rice-Roti		
Veg	Main Course		
Veg	Dessert		
Non-Veg	Starters		
Non-Veg	Main Course		

Copyright © Indian Restaurant

Add Category

Admin Panel

Dashboard

Order

Category

Items

Area

Deliver Locations

Welcome Subha

Logout

Dashboard / Category / Add New

Add New category

Enter Name of the Type

--Select Type--

Enter Name of the Category

Add New

Copyright © Indian Restaurant

All Items

Admin Panel

- Dashboard
- Order
- Category
- Items**
- Area
- Deliver Locations

Welcome Subha Logout

Dashboard / Items

All Items

Add New Item

Sr. No.	Name of the Item	Spicy	Price	Item Image	Type	Update	Delete
1	Kakori Kebabs [6pcs]	Low	110		Starters[Non-Veg]		
2	Stir Fried Chilli Chicken	Medium	120		Starters[Non-Veg]		
3	Chicken Satay [6pcs]	Low	150		Starters[Non-Veg]		
4	Chicken Wings [6pcs]	Low	120		Starters[Non-Veg]		
5	Tangri Kebabs [4pcs]	Low	140		Starters[Non-Veg]		
6	Fried Chicken [6pcs]	Low	150		Starters[Non-Veg]		
7	Malai Tikka	Low	150		Starters[Non-Veg]		

Add New Items

Admin Panel

- Dashboard
- Order
- Category
- Items**
- Area
- Deliver Locations

Welcome Subha Logout

Dashboard / Items / Add New

Add New Items

Select Category

-Select-

Item Name

Select spicy type

-Select Spicy-

Item Price

Select Item Image

Choose File

No file chosen

Add Item

Copyright © Indian Restaurant

Area

Admin Panel

Dashboard

Order

Category

Items

Area

Deliver Locations

Welcome Subha Logout

Dashboard / Area

Area

Add

All Areas

Area Name	Update	Delete
Madhyamgram		
New Barrackpur		
Phoolbagan		
Sealdah		

Back

Add New Area

Admin Panel

Dashboard

Order

Category

Items

Area

Deliver Locations

Welcome Subha Logout

Dashboard / Area / Add New

Add New Area

Enter The Name of the Area

Add Area

Back

Copyright © Indian Restaurant

Delivery Location

Admin Panel

Dashboard

Order

Category

Items

Area

Deliver Locations

Welcome SubhaLogout

Dashboard / Delivery Locations

Delivery Locations

Add New Location

Select Area

---Select Area---

Search

<Copyright © Indian Restaurant

Add Delivery Location

Admin Panel

Dashboard

Order

Category

Items

Area

Deliver Locations

Welcome SubhaLogout

Dashboard / Delivery Locations / Add New

Add New Location

---Select---

Enter The Name of the Location

Add Location

<Copyright © Indian Restaurant

16. Coding:

Home Page:

17. Testing

Team Interaction

The following describes the level of team interaction necessary to have a successful product.

- The Test Team will work closely with the Development Team to achieve a high quality design and user interface specifications based on customer requirements. The Test Team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.
- The Test Team will work closely with Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.
- Since the application interacts with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

Test Objective

The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. We will be testing a Binary Search Tree Application utilizing a pre-order traversal format. There will be eight key functions used to manage our application: load, store, clear, search, insert, delete, list in ascending order, and list in descending order. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a

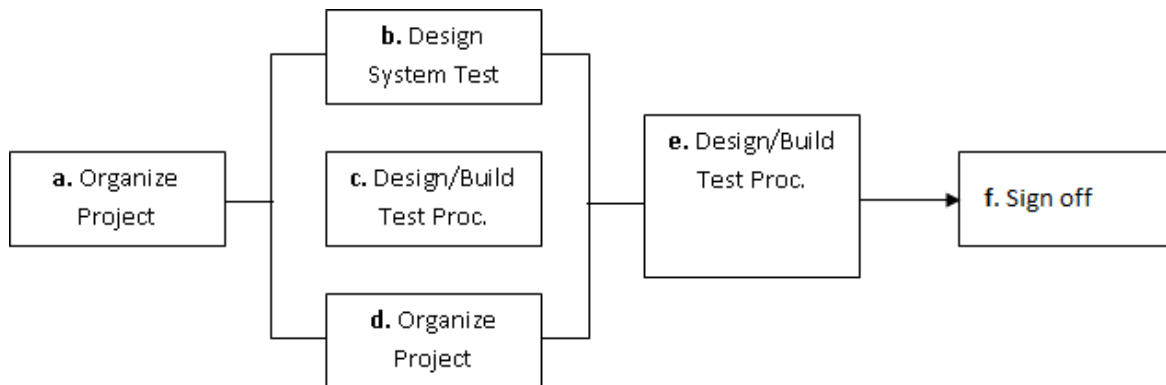
demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

Process Overview

The following represents the overall flow of the testing process:

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
2. Identify which particular test(s) will be used to test each module.
3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test.
5. Document the test case configuration, test data, and expected results.
6. Perform the test(s).
7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
8. Successful unit testing is required before the unit is eligible for component integration/system testing.
9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

Testing Process



The diagram above outlines the Test Process approach that will be followed.

- a. Organize Project** involves creating a System Test Plan, Schedule & Test Approach, and assigning responsibilities.
- b. Design/Build System Test** involves identifying Test Cycles, Test Cases, Entrance & Exit Criteria, Expected Results, etc. In general, test conditions/expected results will be identified by the Test Team in conjunction with the Development Team. The Test Team will then identify Test Cases and the Data required. The Test conditions are derived from the Program Specifications Document.
- c. Design/Build Test Procedures** includes setting up procedures such as Error Management systems and Status reporting.
- d. Build Test Environment** includes requesting/building hardware, software and data set-ups.
- e. Execute System Tests** – The tests identified in the Design/Build Test Procedures will be executed. All results will be documented, and Bug Report Forms filled out and given to the Development Team as necessary.
- f. Signoff** - Signoff happens when all pre-defined exit criteria have been achieved.

Testing Strategy

The following outlines the types of testing that will be done for unit, integration, and system testing. While it includes what will be tested, the specific use cases that determine how the testing is done will be detailed in the Test Design Document. The test cases that will be used for designing use cases is shown in Figure 2.1 and onwards.

Test Cases:

Tested By:	Jayanta Pal
Test Type	Unit Testing
Test Case Number	1
Test Case Name	User Identification
Test Case Description	The user should enter his/ her accurate userid and password so that he/she can able to go for the further options. The test case will check the application for the same since a user can only login with the correct userid , password.
Item(s) to be tested	
1	Verification of the userid and password with the record in the database.
Specifications	
Input	Expected Output/Result
1) Correct User id and password	1) Successful login
2) Incorrect Id or Password	2) Failure Message

Tested By:	Jayanta Pal
Test Type	Unit Testing
Test Case Number	2
Test Case Name	Add User Details
Test Case Description	The User must add details.
Item(s) to be tested	
1	Check whether all the mandatory fields are filled up.
Specifications	
Input	Expected Output/Result
1) Trying to submit without filling the mandatory fields.	1) A message asks the user to fill the form correctly
2) Check the valid data for all form fields.	2) Return an error message.

Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness.

White Box Testing

In white box testing, the UI is bypassed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

Black Box Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case we'll focus only on function validation and performance. And in both cases, we will use the black-box method of testing.

18. System Security measures (Implementation of security for the project developed)

- Only authorized users are allowed.

Without signing in users are not allowed to go an intermediate page by typing an URL. For all such efforts, users will be redirected to the home page

19. Database/Data security

- Database is present in remote machine.
- PHP's default securities are applied.

20. Creation of User profiles and access rights

- The admin must create users manually
- The admin can create more admins

21. Cost Estimation of the Project along with Cost Estimation Model

Analogous estimate of effort or cost

Used for Early Estimate or Individual Activity Estimate

Sample example shown below is for two major deliverables of a software project. You use a previous project as a benchmark for analogous estimation. Using your experience, you will estimate a multiplier.

Multipliers:

1. Prototyping: 0.75.
2. Testing: 0.5
3. Deployment: 0.5

Finally, if you want to convert to cost, you would use current rates for the resource.

WBS ID	Previous Similar Project Activity	Previous Effort	Current Project Estimate	Multiplier	Effort (Previous Effort * 0.75)	Cost (Rs. 500/hr.)
1	Prototyping	40 Work-Hours	Prototyping	0.75	30 Work-hours	Rs. 15000/-
2	Testing	20 Work-Hours	Testing	0.50	10 Work-Hours	Rs. 5000/-
Total					40 Work-Hours	Rs. 20000/-

Note: Effort is also called Size and unit of estimation is called either Work-Hour, person-hours.

22. Future scope and further enhancement of the Project

Online Restaurant System has lot of enhancement options. Like search option for *food, comments of customers on food items, more images of same food item, quick checkout option, similar food suggestion.*

23. Bibliography

1. Roger S. Pressman. Software Engineering: A Practitioner's Approach (Sixth Edition, International Edition). McGraw-Hill, 2005.
2. Ian Sommerville. Software Engineering (Seventh Edition). Addison-Wesley, 2004.
3. Frederick P. Brooks. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Addison-Wesley Pub Co; 1st edition (August 2, 1995).
4. PHP and MySQL Web Development by Luke Welling
5. Learning PHP, MySQL, JavaScript, CSS...by Robin Nixon
6. PHP and MySQL for Beginners by Mark A Lassoff
7. Database System Concepts by Abraham Silberschatz, Henry F. Korth, S. Sudarshan
8. Database System, written by Connolly & Begg
9. HTML BY Davinder Singh Minhas
10. HTML and CSS: Design and Build Websites, Author: Jon Duckett.

24. Appendix

DFD conventions, guidelines, and common errors

Although DFDs are simple to understand and draw, students and practitioners alike encounter similar types of problems while modelling software problems using DFDs. While learning from experience is a powerful thing, it is an expensive pedagogical technique in the business world. It is therefore helpful to understand the different types of mistakes that users usually make while constructing the DFD model of systems.

- **Many beginners commit the mistake of drawing more than one bubble in the context diagram. A context diagram should depict the system as a single bubble.**
- **Many beginners have external entities appearing at all levels of DFDs. All external entities interacting with the system should be represented only in the context diagram. The external entities should not appear at other levels of the DFD.**
- **Many beginners leave different levels of DFD unbalanced.**
- A common mistake committed by many beginners while developing a DFD model is attempting to represent control information in a DFD. It is important to realize that a DFD is the data flow representation of a system, and it does not represent control information. For an example mistake of this kind:
 - o Consider the following example. A book can be searched in the library catalog by inputting its name. If the book is available in the library, then the details of the book are displayed. If the book is not listed in the catalog, then an error message is generated. While generating the DFD model for this simple problem, many beginners commit the mistake of drawing an arrow (as shown in fig. 1) to indicate the error function is invoked after the search book. But, this is a control information and should not be shown on the DFD.

