

DECLARATION

I hereby that the project work entitled “**ART GALLERY**”, submitted in fulfillment for the requirement for the award of the B. Sc Computer Science is a report of the original work done by me during the period of study in **St. Pius X College Rajapuram** as center under the guidance of **Mr.Bibin P A** and supervision of **Mr.Praveen, Mrs.Maneesha** project leaders of **TEQUEVIA TECHNOLOGIES KANHANGAD.**

Rajapuram

Date:

Vaishnav

DECLARATION

I certified that this report titled “**ART GALLERY**” is a record of the project work done by **Vaishnav** under my supervision and guidance towards partial fulfillment of the requirement for award of the Degree of B.Sc Computer Science of the Kannur University.

Place: Rajapuram

Date:

Bibin P A

(Asst.professor-St Pius X College Rajapuram)

ACKNOWLEDGEMENT

I express our humble thanks to almighty, for the kind grace showed us to complete the project successful. I would like to acknowledge the assistance and contribution of those who aided in successful completion of this project.

I have pleasure to acknowledge our deep sense of gratitude to **Dr. Biju Joseph**- Principal St Pius X College Rajapuram, **Mr. Thomas Scaria** Head of the Department, **Mr. Bibin P A** – asst. professor, **Mr. Thomas Chacko** –asst.professor, **Mrs. Jincymol Joseph** – asst.professor, **Mrs. Seenu Anna James** Guest Lecturer and all other teachers for the whole or some guidance and support in completing our project.

I also thank **Mr. Praveen**, **Mrs.Maneesha** for their support, advice and timely help for the success of this project. I want to thank my friends and others including our parents of their help and co-operation for the success of this project. My love and gratitude go once again to all those who supported as for the success of our project.

CONTENTS

1. INTRODUCTION

1.1) INTRODUCTION.....	07
1.2) PROJECT OVERVIEW.....	07
1.3) NEED FOR THE SYSTEM.....	08
1.4) MODEL FOR DEVELOPMENT.....	08

2. SYSTEM ANALYSIS

2.1) INTRODUCTION.....	10
2.2) METHOD FOR ANALYSIS.....	10
2.3) EXISTING SYSTEM.....	11
2.4) PROPOSED SYSTEM.....	11
2.5) REQUIREMENTS.....	12
2.6) FUNCTIONAL DESCRIPTION.....	13.
2.7) FEASIBILITY STUDY.....	20
2.8) ACTOR IDENTIFICATION.....	21

3. SYSTEM DESIGN

3.1) INTRODUCTION.....	23
3.2) TABLE DESIGN.....	24
3.3) DFD (Data Flow Diagram).....	29
3.4) INTERFACE DESIGN.....	37
3.4.1) INPUT INTERFACE.....	37

3.4.2) OUTPUT INTERFACE.....	37
3.5) SOFTWARE DESCRIPTION.....	38
4. TESTING	
4.1) TESTING AND EVALUATION.....	44
4.2) TESTING STATERGIES.....	47
4.3) TESTING TECHNIQUES.....	48
5. IMPLEMENTATION.....	50
6. SYSTEM MAINTENANCE.....	52
7. CONCLUSION.....	54
8. APPENDIX.....	56
9. BIBLIOGRAPHY.....	63

INTRODUCTION

1.1 INTRODUCTION

The use of computer and Internet has been expanding significantly over the years. Presently the internet has been used widely as a means of communication, as a potential source of information.

In general, project is used to enhance the functionality of the existing system; it will make the current system more users friendly. It is important that a good project can fulfill all requirements users.

This is a web based software system. At the center of the system is the administrators which control the entire activities of the project named “Art Gallery” deals with the designing of a software for registration, product registration, bidding product ,order bidding product,order normal product etc..

1.2 PROJECT OVERVIEW

This ART GALLERY system helps Artists to Sell their art without selling to an intermediators (shops etc..). Admin will control/guide overall view of communication between the users (Artist and Customers). Admin and users will login themselves by the system on the basis of username and password. The Artist uploads the details about their art with all descriptions. The customer can view art and they can make a bid and make pay the amount and order the product they need..

The complaints by customer, which can be viewed by the admin. Admin will reply for the complaints that received from the customers and take action against the complaints. By this web application, the Artist can get more profit by eliminating the intermediary other than selling it in shop or at in the exhibitions.

- Admin
- Artists
- Customers

1.3 NEED FOR THE SYSTEM

The major objectives of the computerization in any field are to make man's job easier. The "Art Gallery" reduces the difficulty in selling art works in any situations, and also it helps to grab the quality art work at our bid price and also helps the artist both mentally and financially. We identify drawback of the existing system and overcome them through the proposed system. The objectives of the software are

- **user friendly**
- **Highly flexible**
- **Time saving**
- **Secured**

1.4 MODEL

Spiral model is used to develop this application with object oriented concepts. The spiral model is another iterative model that has been proposed. As the name suggests, the activities in this model can be organized like a spiral that has many cycle. Each cycle in this spiral begin with the identification for objectives for that cycle, the different activities that exist. The next step is to develop strategies that resolve the risk one effective use of the iterative model is often seen in specifications and therefore have a lot of control on which specifications go in the system and which stays out in a customized software development, where the client has to provide and approve the specifications. This process model becomes externally popular in the current business scenario. It is preferable to see return continuously of the investment made.

SYSTEM ANALYSIS

2.1 INTRODUCTION

System analysis is the study of sets of interacting entities, including computer system. This field is closely related requirements analysis. It is simply defined as a structured technique for handling large complex problems to lead to an efficient allocation of resources to meet well defined goals and objectives.

The project entitled “Art Gallery” deals with the designing of a web based application in which user can register, product registration, bid for product, order product etc.. Software for the public to easily grab the need products easily with an affordable price as well as to help their nearby farmers financially with minimum effort.

The Art Gallery consist of different types of users such as,

- Admin
- Artists
- Customers

2.2 METHOD OF ANALYSIS

The first step of designing a system is to identify the drawbacks in the existing systems. The working of the existing system is to examine and study for this purpose. One must know what information is to be gathered, where to find it and how to make it use. Data are gathered and checked for completeness and accuracy. Analysis of data involve identification of the component of the system and their inter relationship and identify the strength and weakness of the system are part of their process. The tools for gathering information are,

- Interviews
- Phone conversation

2.3 EXISTING SYSTEM

In the existing system normally, the customers buy the products by visiting the market or by any other art shops. More over there is no such a good platform for Artist to sell their work of art. So that the artist are facing more difficulties on selling their art to the right customers

Disadvantages of the existing system:

- There is no such a good platform for selling the art or if it is, then the artist were not getting their expected/satisfied amounts. .
- Maybe be the customers cannot find the needed art at one place, they are willing to go for a search from one place to another.

2.4 PROPOSED SYSTEM

In the proposed system, art gallery web application is a platform is provided for both the parties to sell and buy the products. The system will allow the customers to bid and buy products that they need. The Artist can post their art and customer can bid for the art and if they win on the bid they can order and get art directly by interacting with the Artist. This system is very helpful for both the Artist and the customers.

Advantages of proposed system:

- Easy to register and login.
- Less time consuming.
- More beneficial for the Artists.
- Gives access to customers for selecting the product they need.

2.5 SOFTWARE & HARDWARE REOUIREMENTS

SOFTWARE REQUIREMENTS

Front End : HTML, CSS, JS

Back End : PYTHON (Flask)

Server : WSGI

Database : MYSQL

Operating System : Windows 8 or above

HARDWARE REQUIREMENTS

Processor : Intel core 2.0Hz

Memory Size : 2GB RAM

Mouse : Compatible Mouse

Keyboard : Normal Keyboard

Monitor : Standard Monitor

Internet Connection : Any Internet Connection

2.6 FUNCTIONAL DESCRIPTION

The project is mainly divided into 4 modules. They are

- Admin
- Artists
- Customers

➤ **Admin** module consists of the following,

Login : Admin needs to login in order to avail their respective service

View artist: To view the details of artist.

Manage artist: To delete or update the artist, if needed.

View customer: To view the details of customers.

Manage customer: To delete the customer, if needed.

View Normal Products: To view the normal products.

Manage Normal Products : To delete or update normal product if needed.

View bidding products : To view the bidding products.

Manage bidding products: To delete or update the products if needed.

Add notification: To delete the application.

View notification : To view announcement and to delete the announcement.

View and reply to complaints: To view the complaints from the users and reply it.

View feedback: To view the feedbacks from the users.

➤ **Artist** module consist of the following,

Registration: To use this web application user must register first by adding basic details such as name, e-mail, phone number, password, etc.

Login: Artist needs to login in order to avail their respective service.

Add normal product: To add the normal product with its detailed description.

Manage normal product: To view and manage the normal product.

Add bidding product: To add the bidding product with its detailed description.

Manage bidding product: To view and manage the bidding product.

View order : To view the order made by the customers.

Manage order : To manage the orders

View Feedback : To view the feedback add by the customers.

View Notification: To view the notification added by the admin.

➤ **Customer** module consists of the following,

Registration: To use this web application user must register first by adding basic details such as name, e-mail, phone number, password, etc.

Login: customer needs to login in order to avail their respective service.

View normal product: To view the normal product with its descriptions.

Order normal product : To order the normal product.

View bidding product : To view bidding products.

Order bidding product : To order the bidding product.

Add product to cart: To add the product to cart for buying

Manage Cart : To view products in cart and to remove products from cart.

Make payment: To make payment for the ordering product.

Add bidding: To add the product for bidding.

View Order : To view the ordered product.

View notification: To view the notification added by the admin.

Add complaints: To adding complaints about the product.

View complaints reply : View replies to the complaint from the admin.

Add Feedback: To add feedback about the web application.

2.7 FEASIBILITY STUDY

Here are three feasibility studies needed for the systems. They are

- Behavioral Feasibility Study
- Technical Feasibility study
- Economic Feasibility Study

2.7.1 ECONOMIC FEASIBILITY STUDY

Economic analysis is the most frequently used method for evaluating the effectiveness of a candidate system. It is more commonly known as cost benefit analysis, the procedure to determine the benefits and saving that are expected from a candidate system and compare them with costs. If the benefits outweigh cost then a decision is made to design and implement the system. Otherwise make alteration in the proposed system.

2.7.2 BEHAVIOURAL FEASIBILITY STUDY

People are inherently resistant to change and computers have been known to facilitate change. An estimate should be made about the reaction of the user staff towards the development of a computerized system. Computer installation has something to do with turnover, transfers and changes in jobs status. The introduction of a candidate system requires special effort to educate, and train the staff. The system was found to be technically, economically and behaviorally feasible. The system was developed user friendly, needless training and improves efficiency. Disregarding the initial expenses, the candidate system was assessed to be feasible in all ways.

2.7.3 TECHNICAL FEASIBILITY STUDY

The assessments of technical feasibility centers on the existing system and to what extent it can support the proposed addition. This was based on an outline design of system requirements in turns of inputs ,files ,programs, procedures and staff. It involves financial considerations to accommodate technical enhancements.

2.8 ACTOR IDENTIFICATION

ACTOR

An actor is someone or something that interact with the system. An actor is he/she who uses the system. An actor exchanges information with the system. Asking certain questions as detailed below can identify the actor of the system.

IDENTIFICATION OF ACTOR

We can identify actors through a list of questionnaires.

Who will use the functionality of the system:-Administrator

Who will lead support from the system and do their daily tasks:-clients

Who will lead maintain and administrate the system:-Administrator

With which other this system need to interact:-database

Which hardware device does the system need to handle:-No

Who was interface in the result to produce by the system:-Client

The actor of the system is administrator

SYSTEM DESIGN

3.1 INTRODUCTION

System design provide an understanding of the procedural details , necessary Implementing of the system recommended in the feasibility study. Basically it is all about the creation of a new system. This is a critical phase since it decides the quality of the system and has a major impact on the testing and implementation phase.

System design consists of three major steps.

Drawing of the expanded system data flow charts to identify all the processing functions required.

The allocation of the equipment and the software to be used.

The identification of the test requirements for the system.

Characteristics of design

A design should exhibit a hierarchical organization that make intelligent use of control among component of the software.

A design should be modular that is , the software should be logical.

A design should contain distinct and separable representation of data and procedure.

A design should lead to interface that reduce the complexity of the connection between modules and with the external environment.

3.2 TABLE DESIGN

DB design is required to manage large bodies of information. The management of data involves both the definition of the structure of storage of information and provisions of mechanism for the manipulation of information. For developing an efficient database certain conditions have to be fulfilled such as:

- **Control redundancyEase of use**
- **Data independence Accuracy and integrity**

There are five major steps in design process:

- **Identify table and relationship**
- **Identify the data that is needed for each table and relationship**
- **Resolve the relationship**
- **Verify the design**
- **Implement the design**

ART GALLERY

Table name: Login

Purpose: Login process

Primary key: lid

Column Name	Data type	Constraints	Description
lid	int	Primary key	Login id
username	varchar(70)	Not null	Username
password	varchar(45)	Not null	Password
type	varchar(45)	Not null	Type of user

ART GALLERY

Table name: registration

Purpose: Registration process

Foreign key: rid

Column Name	Data type	Constraints	Description
rid	int	Primary key	Registration id
fname	varchar(45)	Not null	First name
lname	varchar(45)	Not null	Last name
email	varchar(45)	Not null	Email
phone	int	Not null	Contact number
district	varchar(45)	Not null	District
city	varchar(45)	Not null	City
pin	int	Not null	Pincode
ass_regno	int	Not null	Society register number
adhaarno	varchar(16)	Not null	Adhaar card number
gender	varchar(45)	Not null	Gender
adrs	text	Not null	Address of user
created_on	datetime	Not null	Created datetime
lid	int	Foreign key	User_id
type	varchar(45)	Not null	Type of the user

ART GALLERY

Table name: reg_payment

Purpose: to store registration payment details

Primary key: id

Column Name	Data type	Constraints	Description
id	int	Primary key	Payment Id
u_id	int	Foreign key	User id
date	datetime	Not null	Payment date
status	varchar(45)	Foreign key	Pay status
card_holder	varchar(45)	Not null	Name of card holder
card number	varchar(16)	Not null	Card number
csv	varchar(3)	Not null	Cvv number
exp	date	Not null	Expiry date

Table name: bidding

Purpose: to store bidding details

Primary key: bid_id

Column Name	Data type	Constraints	Description
bid_id	int	Primary key	Bidding Id
b_id	int	Forigen key	Bid Id
bid_status	varchar(45)	Not null	Bid status
bid_price	varchar(45)	Not null	Bid Price
date	datetime	Not null	Date of bid
c_id	int	Foreign key	Customer id

ART GALLERY

Table name: product_bidding

Purpose: To store product bidding details

Primary key: pb_id

Column Name	Data type	Constraints	Description
pb_id	int	Primary key	Category Id
p_id	int	Foreign key	Product id
start	datetime	Not null	Bid start time
end	datetime	Not null	Bid end time

Table name: pro_registration

Purpose: to store the details registered products

Primary key: pid

Column Name	Data type	Constraints	Description
pid	int	Primary key	Pro_registration id
pname	varchar(90)	Not null	Name of the product
image	text	Not null	Image of the product
desc	varchar(300)	Not null	Description about the product
price	float	Not null	Price of the product
stock	varchar(45)	Not null	Stock of the product
logid	int	Foreign key	Id of user
type	varchar(45)	Notnull	Type of user
o_price	varchar(45)	Not null	Order price

ART GALLERY

Table name: cart

Purpose: to store cart details

Primary key: crt_id

Column Name	Data type	Constraints	Description
crt_id	int AI PK	Primary key	Cart Id
pid	varchar(45)	Foreign Key	Product Id
qnty	int	Not null	Total quantity
total	float	Not null	Total amount
usr_id	int	Foreign key	Id of user

Table name: payment

Purpose: to store payment details

Primary key: pay_id

Column Name	Data type	Constraints	Description
pay_id	int	Primary key	Payment Id
oid	int	Foreign key	Order Id
amount	float	Not null	Total amount
uid	int	Foreign key	Id of user
card_holder_name	varchar(60)	Not null	Name of card holder
card_number	varchar(16)	Not null	Card number
exp	date	Not null	Expiry date
payment_date	datetime	Not null	Payment date

ART GALLERY

Table name: orders

Purpose: to store order details

Primary key: oid

Column Name	Data type	Constraints	Description
oid	int	Primary key	Order id
l_id	int	Foreign key	User_id
total	float	Not null	Total amount
pay_status	varchar(45)	Not null	Payment status
date	datetime	Not null	Date
order_status	varchar(45)	Not null	Order Status

Table name: orderitem

Purpose: to store ordered items

Primary key: o_itm

Column Name	Data type	Constraints	Description
o_itm	int	Primary key	Order id
pid	int	Foreign key	Product_id
oid	int	Foreign key	Order_id
qnty	varchar(45)	Not null	Quantity
total	float	Not null	Total amount

ART GALLERY

Table name: notification

Purpose: to store notification

Primary key: id

Column Name	Data type	Constraints	Description
id	int	Primary key	Announcement Id
subject	varchar(60)	Not null	Subject of announcement
content	text	Not null	Content of announcement
date	datetime	Not null	Date of announcement

Table name: complaint

Purpose: to store complaints.

Primary key: cmp_id

Column Name	Data type	Constraints	Description
cmp_id	int	Primary key	Complaint id
lid	int	Foreign Key	Id of user
complaint	text	Not null	Complaint
reply	text	Not null	Admin Reply
complaint_date	date	Not null	Date of complaint
reply_date	datetime	Not null	Date of reply

ART GALLERY

Table name: feedback Purpose:

to store feedbacks. Primary key: fid

Column Name	Data type	Constraints	Description
fid	int	Primary key	Feedback Id
lid	int	Foreign key	Id of user
subject	varchar(100)	Not null	Subject of feedback
feedback	text	Not null	Feedback content
created_on	datetime	Not null	Date of feedback

3.3 DFD-DATA FLOW DIAGRAM

A graphical representation is used to describe and analysis the movement of data through a system manual or automated including the processes, storing of data and delays in the system. Data flow diagrams are the central tool and the basis from which other components are developed.

The transformation data, from input to output through process may be described logically and independently of the physical components associated with the system.

They are termed logical dataflow diagrams showing the actual implementation and the movement of data between people, department and workstation. DFD is one of the most important modeling tool is used in system design. DFD shows the flow of data through different process in the system.

PURPOSE

The purpose of the system design is to create architecture for the evolving implementation and to establishing the common tactical policies that must be used by desperate elements of the system. We begin the design process as soon as we have reasonably completed model of the behaviour of the system. It is important to avoid premature designs, where in develop designs for analysis reaches closer. It is important to avoid delayed where in the organization crashes while trying to complete an unachievable analysis model.

Throughout our project, the context flow diagrams, data flow diagrams and flow charts have been extensively used to achieve the successful design of the system. In my opinion " efficient design of data flow and context flow diagram helps to design the system successfully without much major flows within the scheduled time " . This is the most complicated part in the project. In the designing process, my project took more than the activities in the software life cycle. If we design a system efficiently with all the future enhancements the project will never become junk and it will be operational.

The data flow diagrams were first developed by Larry Constantine as a way of expressing system requirements in graphical form. A data flow diagram is also known as "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. It functionality decomposes the requirement specification down to the lowest level. Data Flow Diagram depicts the information flow, transformation flow and the transformations that are applied as data move from input to output. Thus DFD describes what data flows rather than how they are processed.

Data Flow Diagram is quite effective: especially when the required design is unclear and the user and analyst need a notational language for communication. It is one of the most important tools used during system analysis. It is used to model the system components such as the system process, any external entities that interact with the system and information flows in the system.

Data Flow Diagrams are made up of a number of symbols, which represents system components. Data Flow modeling method users four kinds of symbols, which used to represent four kinds of system components. There are

1. Process.
2. Data stores.
3. Data flows.
4. External entity.

PROCESS

Process shows the work of the system. Each process has one or more data inputs and procedure one or more data outputs. Processes are represented by rounded rectangles in Data Flow Diagram. Each process has a unique name and number. This name and number appears inside the rectangle that represents the process in a Data Flow Diagram.

DATA STORES

A data source is a repository of data. Processes can enter data, into a store or retrieve the data from the data store. Each data has a unique name.

DATA FLOWS

A data flow shows the passage of data in the system and is represented by lines joining system components. An arrow indicates the direction of flow and the line is labeled by name of the data flow.



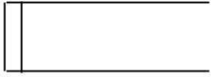

EXTERNAL ENTITY

External entities are outside the system but they either supply input data into the system or use other systems output. They are entities on which the designer has control. They may be an organizations customer or other bodies with which the system interacts. External entities that use the system data are sometimes called sinks. These are represented by rectangles in the Data Flow Diagram. Four basic symbols are used to construct data flow diagram. They are symbols that represented by rectangles in the Data Flow Diagram

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, data transformations and data storage. The points at which data are transformed by enclosed figures, usually circles, which are called nodes.

Basic data flow diagram symbols are :

ART GALLERY

	Source or destination of system data
	Processes
	A Database/table
	Flow of data

Four steps are commonly used to construct a DFD.

1. Process should be named and numbered for easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and left to right.
3. When a process is exploded into lower level details they are numbered.
4. The names of data stores, sources and destinations are written in capital letters.

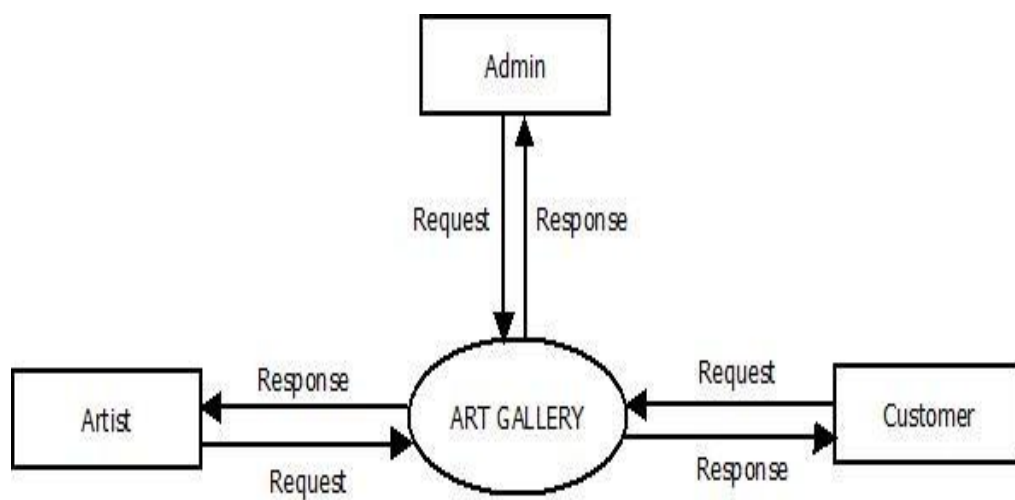
ART GALLERY

Level-0



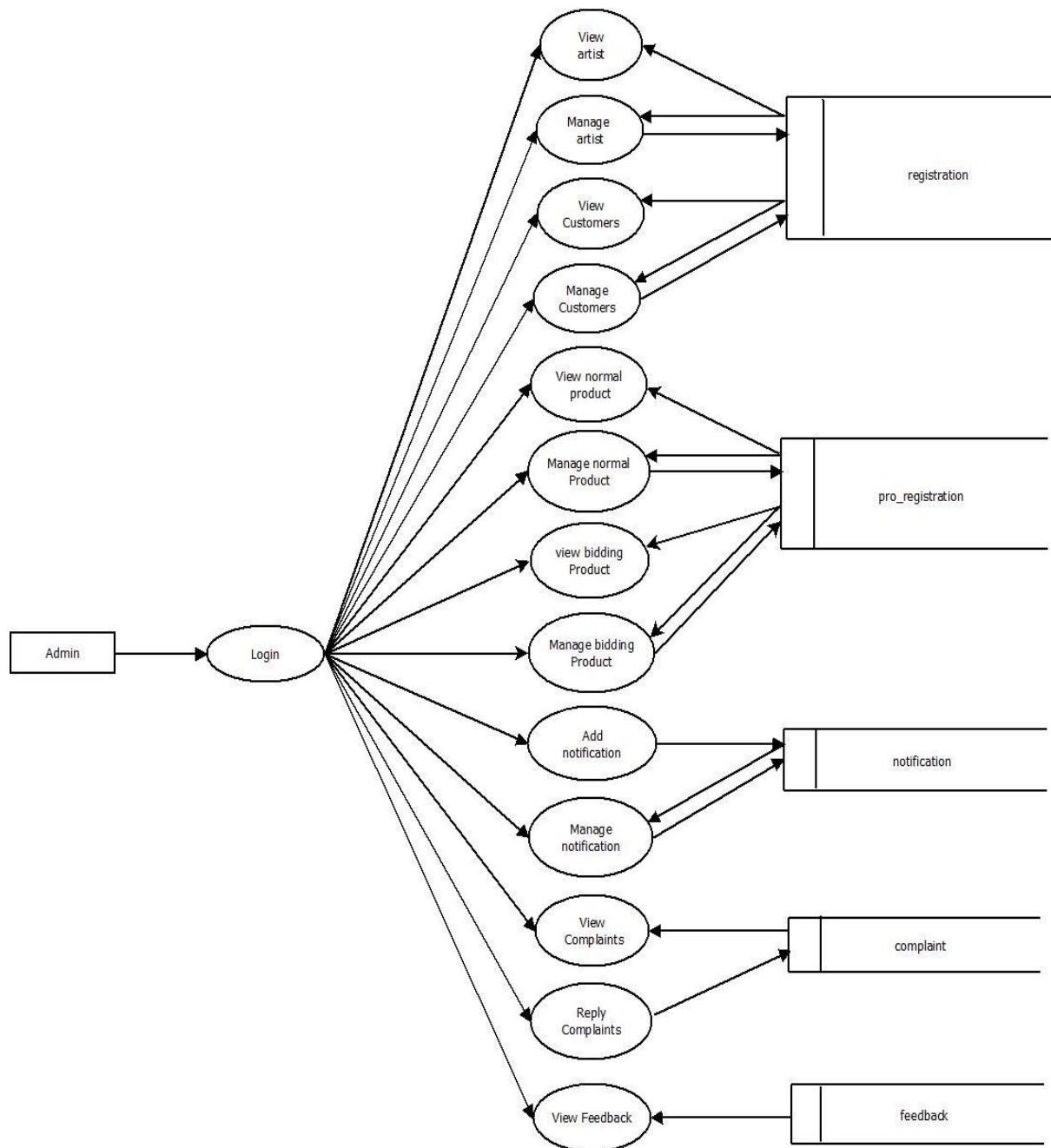
ART GALLERY

Level-1

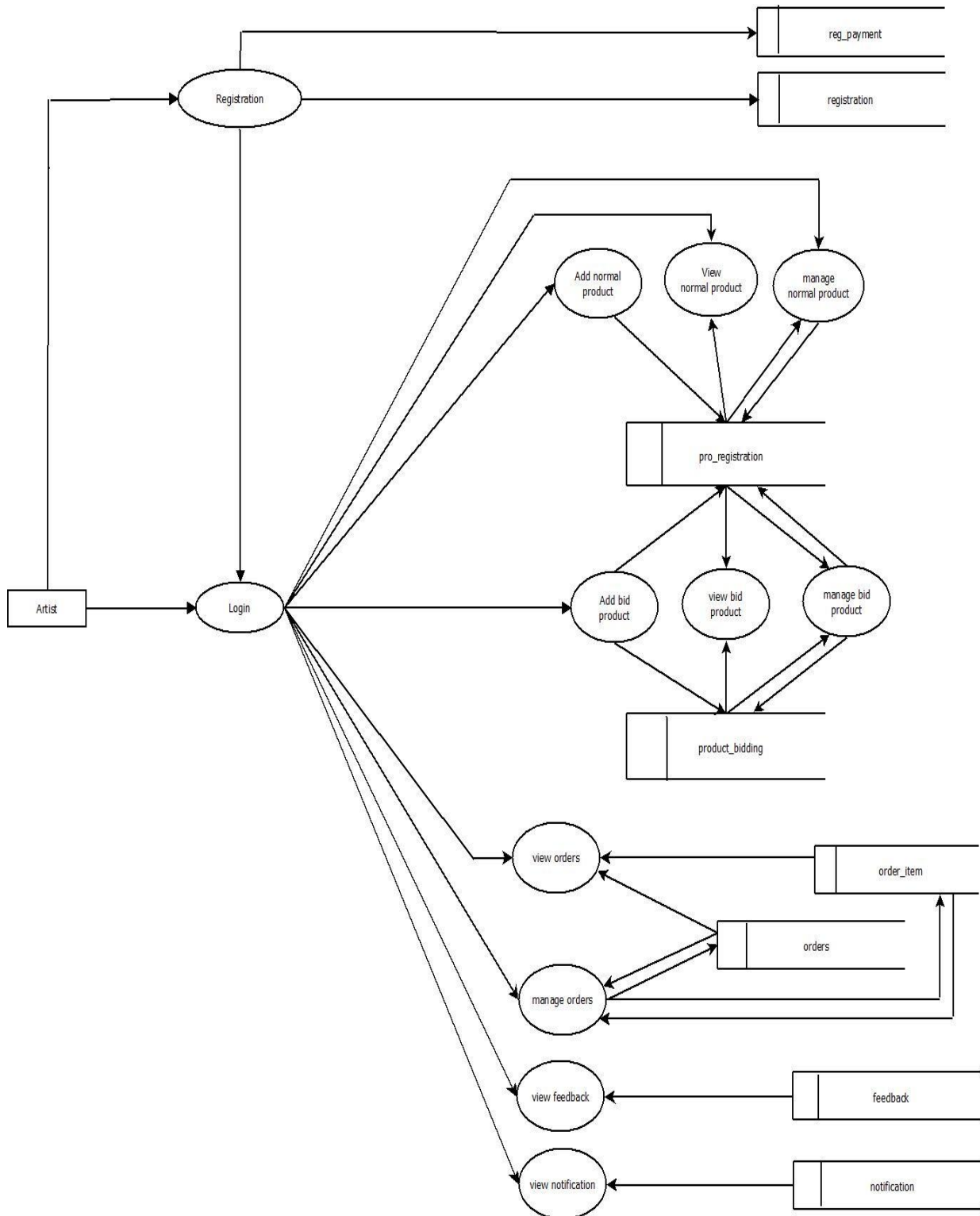


ART GALLERY

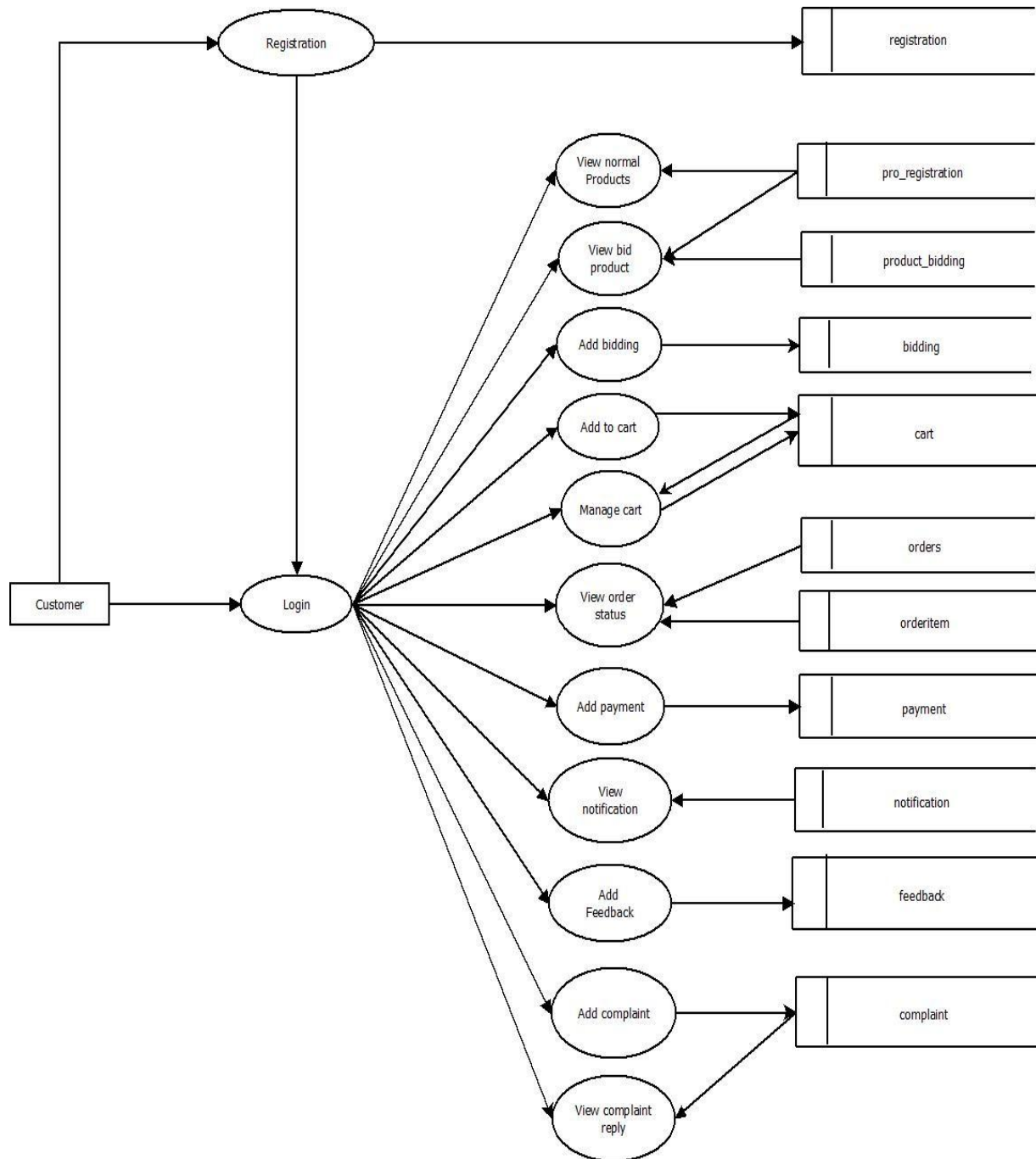
Level-2



ART GALLERY



ART GALLERY



3.4 INTEREACE DESIGN

3.4.1 INPUT INTEREACE

Input design is a part of overall system design, which requires very careful attention. If data going into the system is correct, then the processing and output will magnify these errors. Thus the designer has a number of clear objectives in the different stages of input design.

- ◆ To produce a cost effective method of input.
- ◆ To achieve the highest possible level of accuracy.
- ◆ To ensure that input is acceptable to and understand by the user.

3.4.2 OUTPUT INTEREACE

At the beginning of the output design various types of outputs such as external, internal, operational and interactive and turn around are defined. Then the format, content, location, frequency, volume and sequence of the outputs are specified. The content of the output must be defined in detail. The system analysis has two specific at this stage.

- ➔ To interpret and communicate the results of the computer part of a system to the users in a form, which they can understand, and which meets their requirements.
- ➔ To communicate the output design specifications to programmers in a way in which it is unambiguous, comprehensive and capable of being translated into a programming language.

3.5 SOFTWARE DESCRIPTION

WATERFALL MODEL

Waterfall Model is used to develop this application with object oriented concepts. The waterfall model is another iterative model that has been proposed. As the name suggests, the activities in this model can be organized like a waterfall that has many cycle.

The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach. It is termed as waterfall because the model develops systematically from one phase to another in a downward fashion. This model is divided into different phases and the output of one phase is used as the input of the next phase. Every phase has to be completed before the next phase starts and there is no overlapping of the phases.

There are five phases in waterfall model, they are:

- Requirement analysis and specification
- Design
- Implementation and unit testing
- Integration and system testing
- Operation and maintenance

BOOTSTRAP 4

Bootstrap is a free and open-source CSS framework directed at responsive, mobile first front end web development. It contains CSS and (optionally) java Script –based design templates for typography, forms, buttons, navigation and other interface components.

Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden. According to Twitter developer Mark Otto:

A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company.

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011.[5] It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

PYTHON

Python is an interpreter, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it.[30][31] With Python 2's end-of-life, only Python 3.5.x[32] and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source[33] reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

FLASK

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries.[3] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.[4]

Applications that use the Flask framework include Interest and LinkedIn.

MYSQL

MYSQL is the most popular Open Source database, is developed, distributed and supported by MYSQL LAB. MYSQL is a database management system. MYSQL is a relational database management system. The SQL part of “MYSQL” stands for “Structured Query Language” the most common standardized language used to access databases. MYSQL software is Open Source. Open source means that it is possible for anyone to use and modify it. Anybody can download the MYSQL software from the Internet and use it without paying anything. The MYSQL database server is very fast, reliable and easy to use. It was originally developed to handle large database much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Though under constant development, MYSQL Server today offers a rich and useful set of functions. Its connectivity, speed and security make MYSQL Server highly suited for accessing databases on the Internet.

TESTING

4.1 TESTING AND EVALUATION

Testing is a process of executing program with the intent of finding an error. Software testing is a critical element of software quality assurance and the ultimate review or specification, design and coding. Testing includes verification of the basic logic of each program and verification that the entire system works properly. Testing demonstrates that software functions appear to be working according to specification. In addition, data collected as testing conducted provide a good indication of software quality as a whole. The debugging process is the most unpredictable part of testing process.

Testing begins at the module level and works towards the integration of the entire computer based system testing and debugging are different activities, during the development and modification of the software. There are two types of verifications but any testing includes debugging strategy for software testing must accommodate low level tests that are necessary to verify that a small source code segment has been currently implemented as well as high level tests that validate major system function, against customer requirements. No testing is complete without verification and validation part.

The goals of verification and validation activities are to access and improve the quality of work products generated, they are lifecycle verification and formal verification. Life Cycle verification is the process of determining the degree to which the products of the given phase of the development cycle fulfill the specification established during the prior process. Formal verification is the rigorous mathematical demonstration that source code confirm to its specifications. Validation is a process of evaluating software at the end of the software development process to determine completion with the requirements

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The primary objectives, when we test software are the following:

- ➔ Testing is a process of exceeding with the intent of finding an error.
- ➔ A good test is one that has a high probability of finding an undiscovered error.
- ➔ A successful test is one uncover undiscovered errors.

Automated testing is performed at four different levels:

- * Unit testing.
- * Integration testing.
- * System testing.
- * Acceptance testing.

UNIT TESTING

Unit testing comprises the set of tests performed by an individual programmer prior to the integration of the system. Testing removes residual bugs and improves the reliability of the system. Testing allows the developer to find out the design faults if any, and enable correction if needed. Exhaustive unit testing has to be carried out to ensure the validity of the data. In order to successfully test the entire package, unit test is carried out. Each module was tested as when it was developed. Thus it proved easier to conduct minute testing operation and correct them then and there.

INTEGRATING TESTING

Bottom - up integration is the traditional strategy used to integrate the component of software system into a functional whole. Bottom - Up integration consists of unit testing, followed by subsystem testing and followed by testing of entire system. Unit testing has the goal of discovering the errors in the individual parts of the system

Parts are tested in isolation from one another in an artificial environment known as “Test Harness”, which consists of driver of programs and data necessary to exercise the modules unit testing should be as exhaustive as possible to entire that each representative case handled

by each module has been tested. Unit testing is eased by a system structure that is composed of small loosely coupled modules.

A subsystem consists of several modules that communicate with each other through well-defined interfaces. Normally, subsystem implements a major segment of the total system the primary purpose of the subsystem is to verify operation of the interfaces between two modules in the subsystem. Both control and data interfaces must be tested. Large software system may require several levels of subsystems. Lower level subsystems are successively combined to form higher level subsystems. In most software systems, exhaustive testing of subsystem capabilities is not feasible due to the combination complexity of the module interface. Therefore, test cases must be carefully chosen to exercise the interface in the desired, manner.

ACCEPTANCE TESTING

Acceptance testing involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements. It is not to unusual for two sets of acceptance test to be run, those developed by the quality group and those developed by the customer. In addition to the functional and performance tests, stress tests are performed to determine the limitation of the system. For example, a compiler might be tested to determine the effect of the symbol table overflow, or real-time system might be tested to determine the effect of simultaneous arrival of numerous high priorities interrupts.

SYSTEM TESTING

System testing is similar to integration testing, but instead of integrating modules into programs for testing, programs in a system. System testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

4.2 TESTING STRATEGIES

A strategy for software testing integrated software test case design method in to a well-planned series of steps that results in the successful construction of the software. The strategy provides a road map that describes the step to be conducted as part of testing, when these steps are planned and then undertaken, and how much effort, time and resources will be required. Therefore any testing strategy must incorporate test planning, test case, design, test execution, resultant data collection and evaluation. A software testing strategy should be flexible enough to promote reasonable planning and management tracking as the project progress. The general characteristics of software testing strategies are

- ◆ Testing begins at the component level and works “outward” toward the integration of the entire computer system.
- ◆ Different testing techniques are appropriate at different point in time.

A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements. A strategy must provide guidance for the practitioner and set of mile stones for the manager. Because the step on the test strategy occurs at a time when deadline pressure begins to rise, progress must be measurable and problem must surface as early as possible.

The software teams approach to testing is defining a plan that describes an overall strategy and a procedure that defines specific testing steps and test that will be conducted. In the proposed system, if the administrator makes any attempt to login to the application without entering his password, then the system will not allow the user to login to the application.

4.3 TESTING TECHNIQUES

The various testing techniques are given below.

WHITE BOX TESTING

White-box testing also called a glass box testing, is a test case design method that goes to the control structure of a procedural design to derive test case. Using white-box testing methods, the software engineer can derive test cases that

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercised all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their optional bounds.
- Exercise internal data structure to ensure their validity.

White-box testing was successfully conducted on our system. All independent paths with in a module have been exercised on their true and false sides.

BLACK BOX TESTING

Black - box testing also called behavioural testing, focuses on the functional requirements of the software. It is complementary approach that is likely to uncover different classes of errors than white -box methods.

Black - Box testing attempts to find errors in the following categories.

- ◆ Incorrect or missing functions.
- ◆ Inter face errors.
- ◆ Errors on the data structure or external database access.
- ◆ Behaviour or performance errors.
- ◆ Initialisation and termination errors.

IMPLEMENTATION

IMPLEMENTATION

The implementation phase is less reactive. It is primarily concerned with user training, site preparation and file conversion . It is also tests the users acceptance.

After the completion of project it is aimed to implement or install the system in the company. Prior to the implementation site preparation are made. It includes the assurance of hard and software's that have already met. After the new system is implemented then checks for the readiness and accuracy of the system to access, update and retrieve data from new files. Once the programs become available, test data are read into the comport and processed against the files provided for testing. If successful, the programs are run with „live data“. Thus the data's that the company want to keep in the new system are entered, processed and verified the output. Otherwise, a diagnostic procedure is used to locate and corrects errors in the program. Major activities in implementation stages are:

User training

While computer programs are being developed and tested, the analyst will start familiarizing users with procedures to be used in the new system. The data entry operators must be trained to enter data from the source documents and the correct errors.

Implementation procedure

Implementation phase is to last phase regarding any system development, before any system is implemented. Various tests are performed the system is reviewed thoroughly. The changeover selects is used to change the existing system to the new proposed system. In this phase, the system enters the operation and routine maintenance stage.

Operational documentation

This system is developed in such a way that the existing system facilities are enough for implementation. The hardware facilities are sufficient enough to implement the developed system.

SYSTEM MAINTENANCE

SYSTEM MAINTENANCE

Software Development has many phases. These phases include requirements Engineering, Architecting, design, implementation, testing, software deployment and maintenance. Maintenance is the last stage of software life cycle. After the product has been released, the maintenance phase keeps the software up to date with environment changes and changing user requirements. The earlier phases should be done so that the product is easily maintainable. The design phase should plan the structure in a way that can be easily altered. Similarly, the software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment implementation phase should create code that can easily read, understood, and changed. Maintenance can only happen efficiently if the earlier phases are done properly.

There are four major problems that can slow down the maintenance process, unstructured code, maintenance programmers having insufficient knowledge of the system, documentation being absent, out of date or at best insufficient and software maintenance phase relies on these problems being fixed earlier in the life cycle.

Maintenance consists of four parts. Corrective maintenance with fixing bugs in the code. Adaptive maintenance deals with adapting the software to new environments. Perfective maintenance deals with updating the software according to changes in user requirements. Finally preventive maintenance deals with updating documentation and making the software more maintainable. All changes to the system can be characterized by these four types of maintenance. Corrective maintenance is “traditional maintenance” while the other types are considered as “software evolution”.

CONCLUSION

CONCLUSION

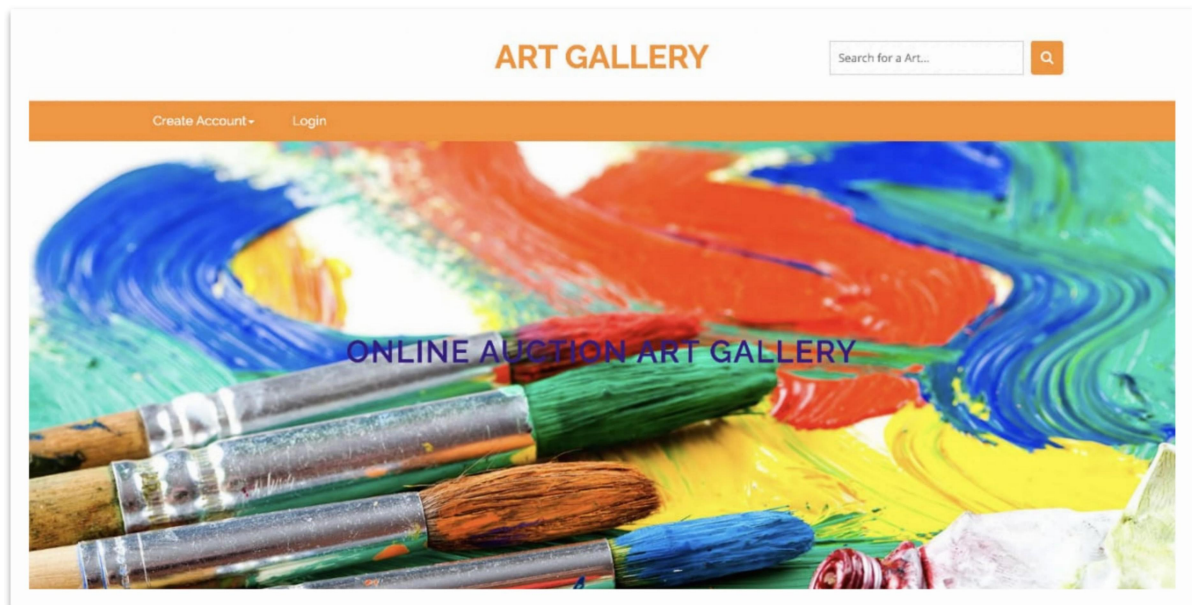
This ART GALLERY system helps Artists to Sell their art without selling to an intermediators (shops etc..). Admin will control/guide overall view of communication between the users (Artist and Customers). Admin and users will login themselves by the system on the basis of username and password. The Artist uploads the details about their art with all descriptions. The customer can view art and they can make a bid and make pay the amount and order the product they need..

The complaints by customer, which can be viewed by the admin. Admin will reply for the complaints that received from the customers and take action against the complaints. By this web application, the Artist can get more profit by eliminating the intermediary other than selling it in shop or at in the exhibitions.

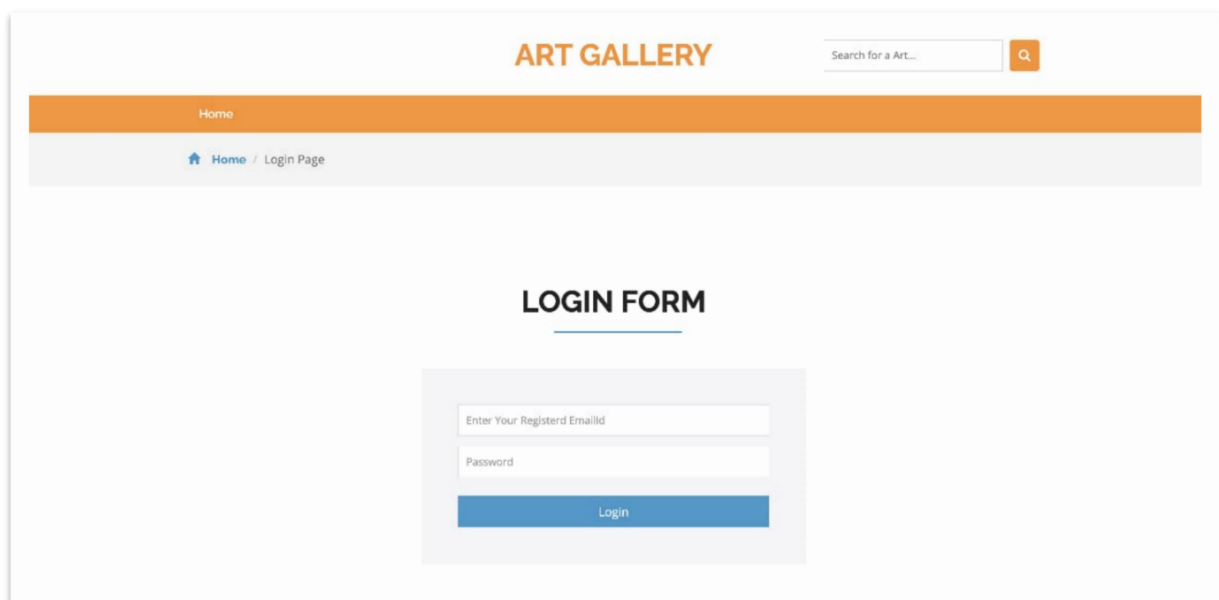
.

APPPENDIX

Index Page



Login Page



Artist Registration

REGISTER HERE

ARTIST INFORMATION

Akshil

panathur

kasargod

kanhangad

671306

9400132253

acv@gmail.com

4564

4532123476879874

Registration Payment Details

₹1000

manyaga

456

June 2022

Register

Customer Registration

REGISTER HERE

CUSTOMER INFORMATION

aruna

arunas house

kasargod

kanhangad

671306

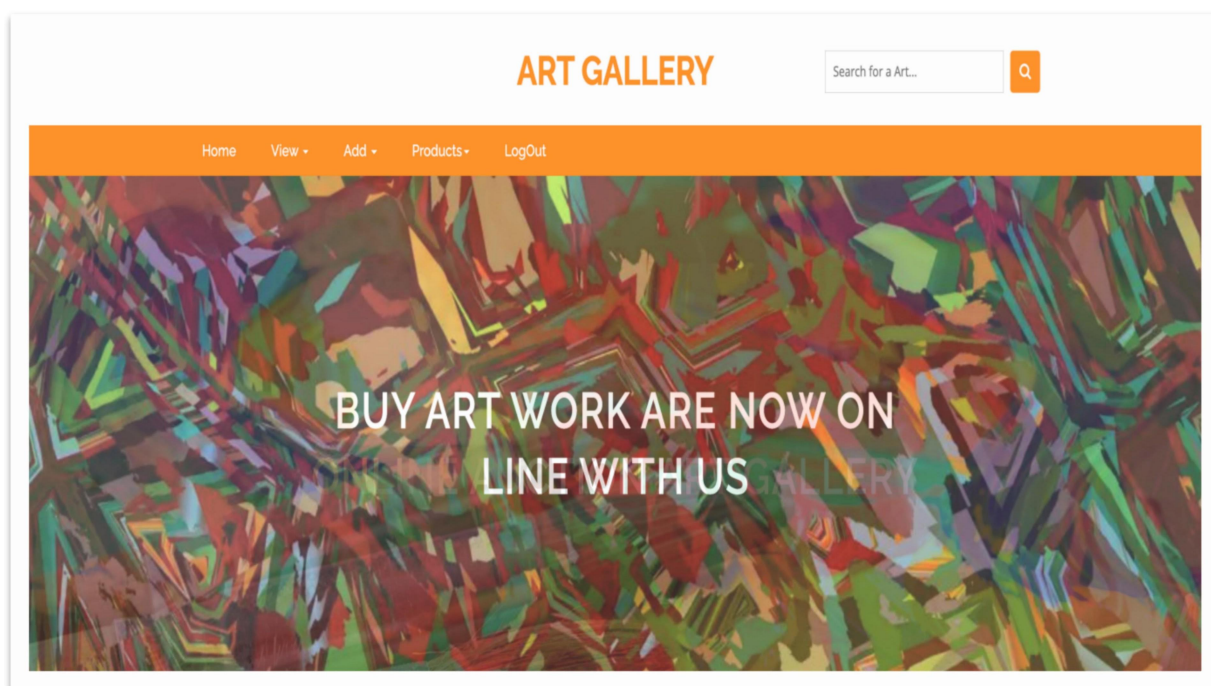
9400132253

oaranuna@gmail.com

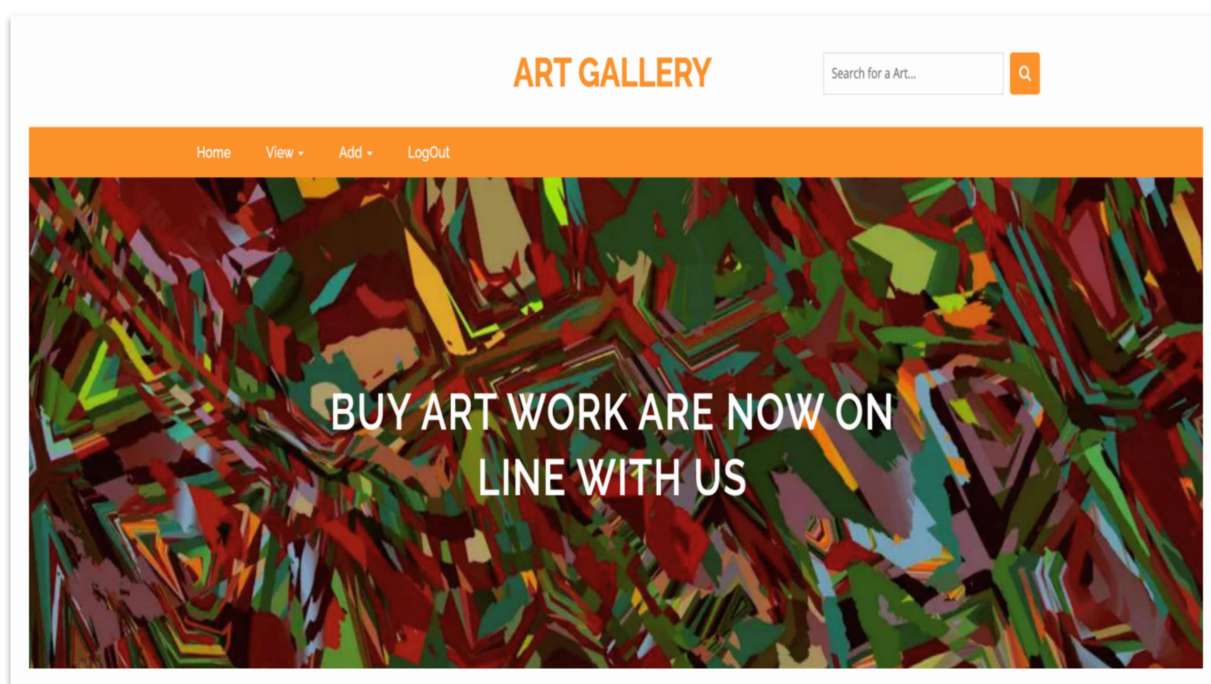
Register

Home

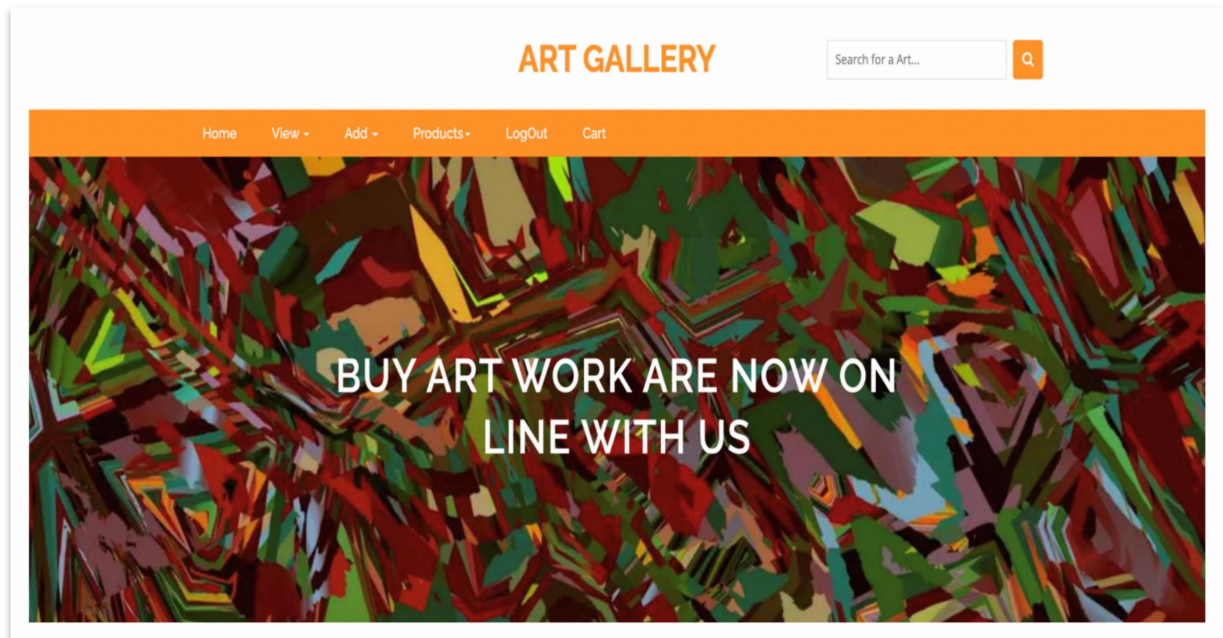
Admin Home



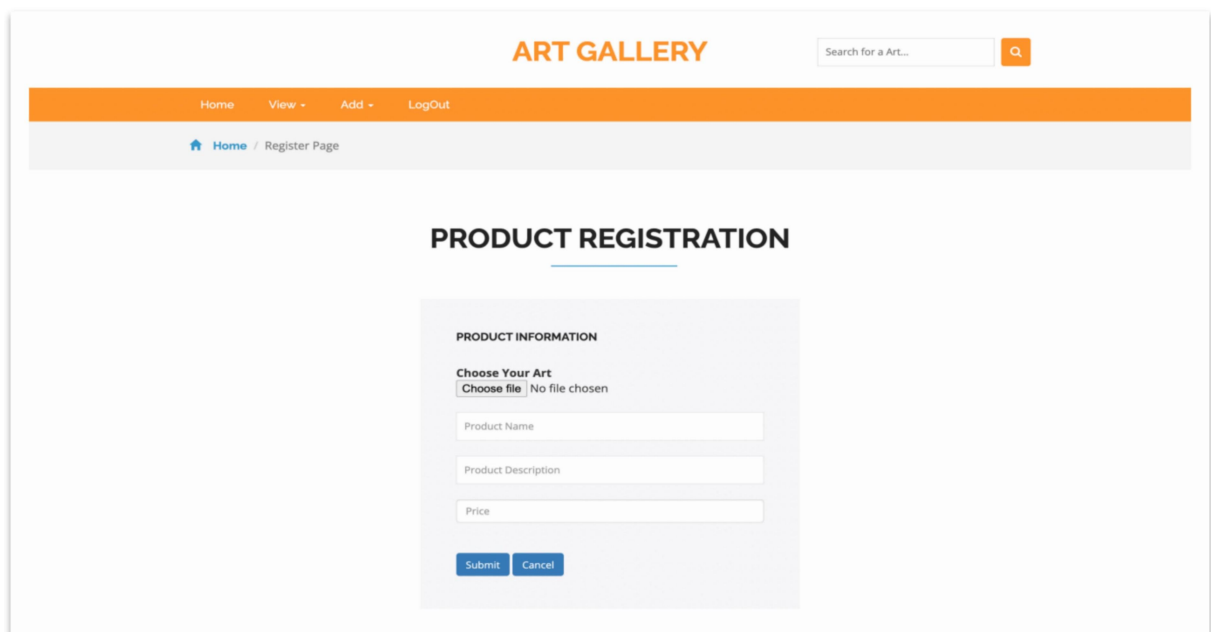
Artist Home



Customer Home



Artist Add Products



Artist Bidding Product Registration

ART GALLERY

Search for a Art...

[Home](#) [View](#) [Add](#) [LogOut](#)

[Home](#) / [Register Page](#)

PRODUCT REGISTRATION

PRODUCT INFORMATION

Choose Your Art

Choose file | No file chosen

Product Name

Product Description

Price

Please add your Bidding End Date

dd/mm/yyyy, --:-- --

Submit Cancel

Customer View Normal Product


ART GALLERY

Search for a Art...

[Home](#) [View](#) [Add](#) [Products](#) [LogOut](#) [Cart](#)

[Home](#) / [Login Page](#)

PRODUCTS




new

New

₹ 234.0

ADD TO CART



test

Test

₹ 123.0

ADD TO CART

ST PIUS X COLLEGE

Page

Admin View User List

ART GALLERY

Search for a Art...

Q

HomeView +Add +Products +LogOut

View User_List

Name	Email	Phone	District	City	Pin	Action
anjusha	anjusha@gmail.com	8848000094	kannur	payyanur	670307	<button>Remove</button>

Admin View Normal Product


ART GALLERY

Search for a Art...

Q

HomeView +Add +Products +LogOut

View Normal Product

Product Name	Image	Description	Price	Action
new		new	234.0	<button>Remove</button>

BILIOGRAPHY

BIBLIOGRAPHY

www.bootstrap.com

www.w3schools.com

www.stackoverflow.in