CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Tailoring software is suitable for the tailors and the fashion designers to help them to keep the database of their clients and their orders, measurement styles, payments and their references from them. Stitching is sending notifications to the customer about their orders, delivery status via E-Mail/ SMS.

It supports helps to manage all orders/sales, customers, income, expenses, measurements, so you can keep things organized and get a detailed report of your fashion business while you concentrate on other important things. it allows you to keep track of customer’s measurements, orders and let you send them SMS when work is done as well as print receipts for them .Stitching Software has kinds of roles to logins such as admin,staff,customer .

1.2 PROJECT SPECIFICATION

The proposed system is a website in which user can order online for products. Also that the customers how come to service center also have access to the web were she can view all the service history.

The system includes 2 modules. They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add, update or delete products details, manage user data etc. Admin can View all the registered users and also manage all his data.

2. Customer Module

Customer can register and they can order for products and also view also information about her products.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

A Tailoring shop is a small establishment set up anywhere in a market place to provide services to the individual for making tailored garments as per one’s choice /fashion. It is preferred to set up near to cloth shop to have more job work from shop and from individual knowledge of understanding fabric quality and skill in cutting and stitching are important to run a tailoring shop successfully. Tailoring shop may undertake job work for made ups and home textiles & interior decoration work.

A good number of customers still prefer to wear tailor-made garments as per their choice. It is cost effective in comparison to readymade garments in a rural and semi-urban area. Tailoring shop can do work for manufacturing of made ups and home furnishings. It may also take a small order from readymade garments manufacturing unitson job work basis. There is scope to undertake job work for interior decoration for tentsetc. on regular basis.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. Customer can register and they can order for products. Each customer can create their own profile .The proposed system rectify the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Difficult to handle the large information.
- It is highly time consuming.
- It gives greater workload to each staff.
- The system is manual one so error occurs may be high.
- It is very difficult to maintain and up-date information.
- There may be chances for losing the records.
- Accuracy is missing Fewer Users - Friendly.
- Accuracy not guaranteed.

- Not in reach of distant users

2.4 PROPOSED SYSTEM

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for growth of stitching center; on such consideration the system is proposed. In our proposed system there is admin who can view all the customers. It allows customers to make their service ordering and do their transactions by using online payment method .Users of this proposed system are admin and customer. The software application which avoids more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the distributors to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. It is very easy to record the information of online sales and purchases in thedatabases.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources, and the system will work in almost all configurations. It has got following features:

- High speed and accuracy.
- Provide high security.
- Good user interface.
- Easy edit, delete and search functionality.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

3.1.1 Economical Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system; otherwise further alterations will have to be made, if it is to have a chance of approved.

The proposed system is cost effective because of its experimental and user-friendly interface. The user can directly view and change the record

The analysis raises financial or economical question during the preliminary investigation to estimate the following,

□ The cost to conduct a full systems investigation.

□ The benefits in the form of reduced costs or fewer costly errors.

□ The cost if nothing changes.

□ The cost of hardware and software for the class application of the project being considered.

3.1.2 Technical Feasibility

This study is carried out to check the technical feasibility that is the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demand on the available technical resources. The developed system must have a modest requirement as only minimal or null changes are required for implementing this system. In Stitching Assistant project works in PHP and uses backend SQL Server. The technical possibility of proposed system is as follows:

□ The unit does process the hardware as well as related software for this project.

□ The proposed system does not require much technical details.

□ The current manual working is not so much sufficient.

□ It is just requires window operating system.

3.1.3 Behavioral Feasibility

It center’s on the reaction of the users. Since the system is user-friendly, user training is an easy matter. Any one, with the basic knowledge of computer can operate the system. The users need not have prior knowledge of PHP.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	-	Intel core i3
RAM	-	4 GB
Hard disk	-	1 TB

3.2.2 Software Specification

Front End	-	HTML, CSS
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server side scripting language designed for web development but also used as a general purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page ,it now stands for PHP: HypertextPreprocessor, a recursive acronym.PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page.PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

3.3.2 MySQL

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

The most creative and challenging phase of system life cycle is system design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementing the candidate system. The elegant design achieves its objectives with minimum use of resources.

The first step is to determine how the output is to be produced and in what format. The input and the database have to be designed to meet the requirements of proposed output.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram

- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

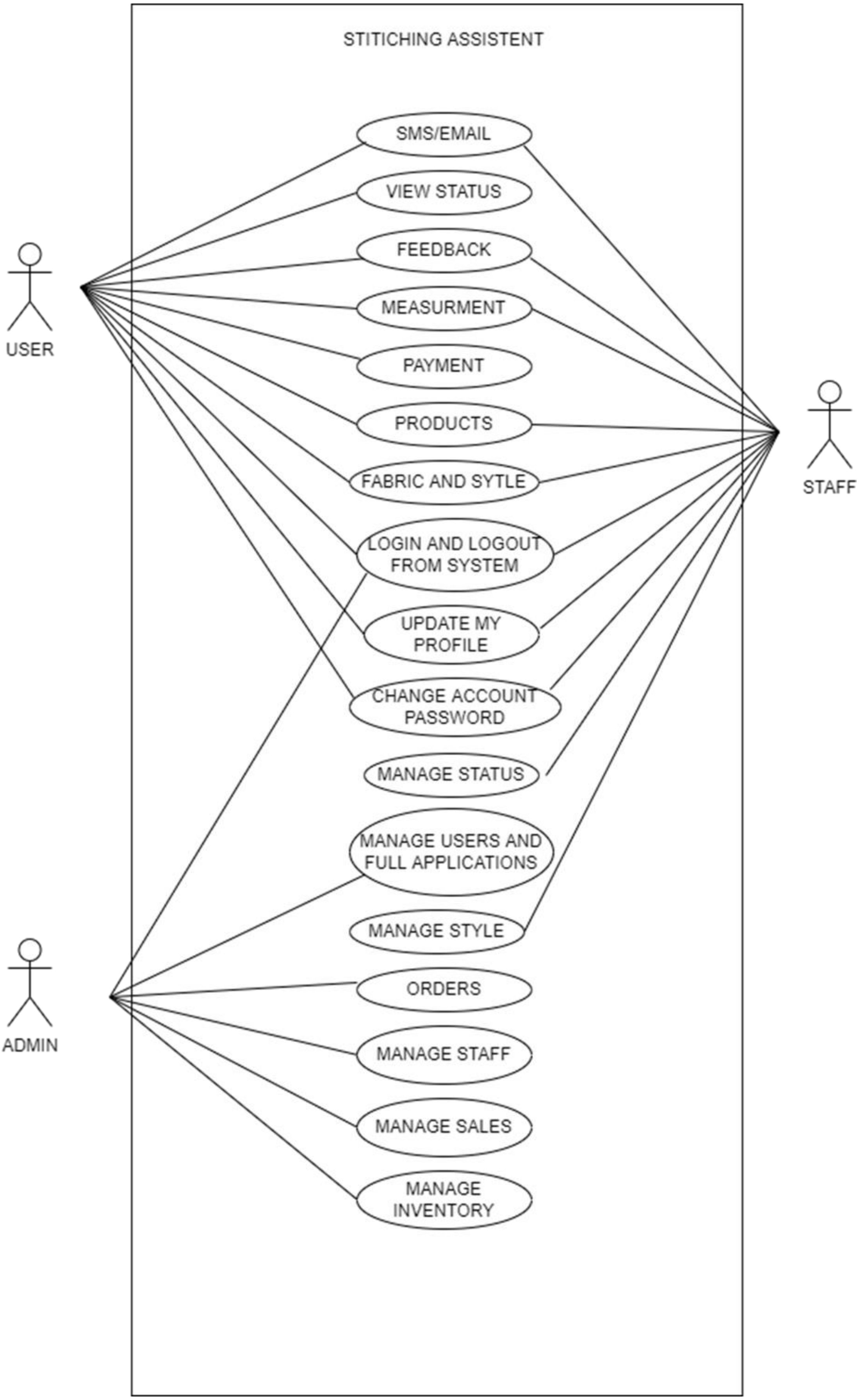
A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation forthe modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.


STITICHING ASSISTENT
USER
STAFF
ADMIN
SMS/EMAIL
VIEW STATUS
FEEDBACK
MEASURMENT
PAYMENT
PRODUCTS
FABRIC AND SYTLE
LOGIN AND LOGOUT FROM SYSTEM
UPDATE MY PROFILE
CHANGE ACCOUNT PASSWORD
MANAGE STATUS
MANAGE USERS AND FULL APPLICATIONS
MANAGE STYLE
ORDERS
MANAGE STAFF
MANAGE SALES
MANAGE INVENTORY

4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Sequence Diagram Notations –

i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.

ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram

iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

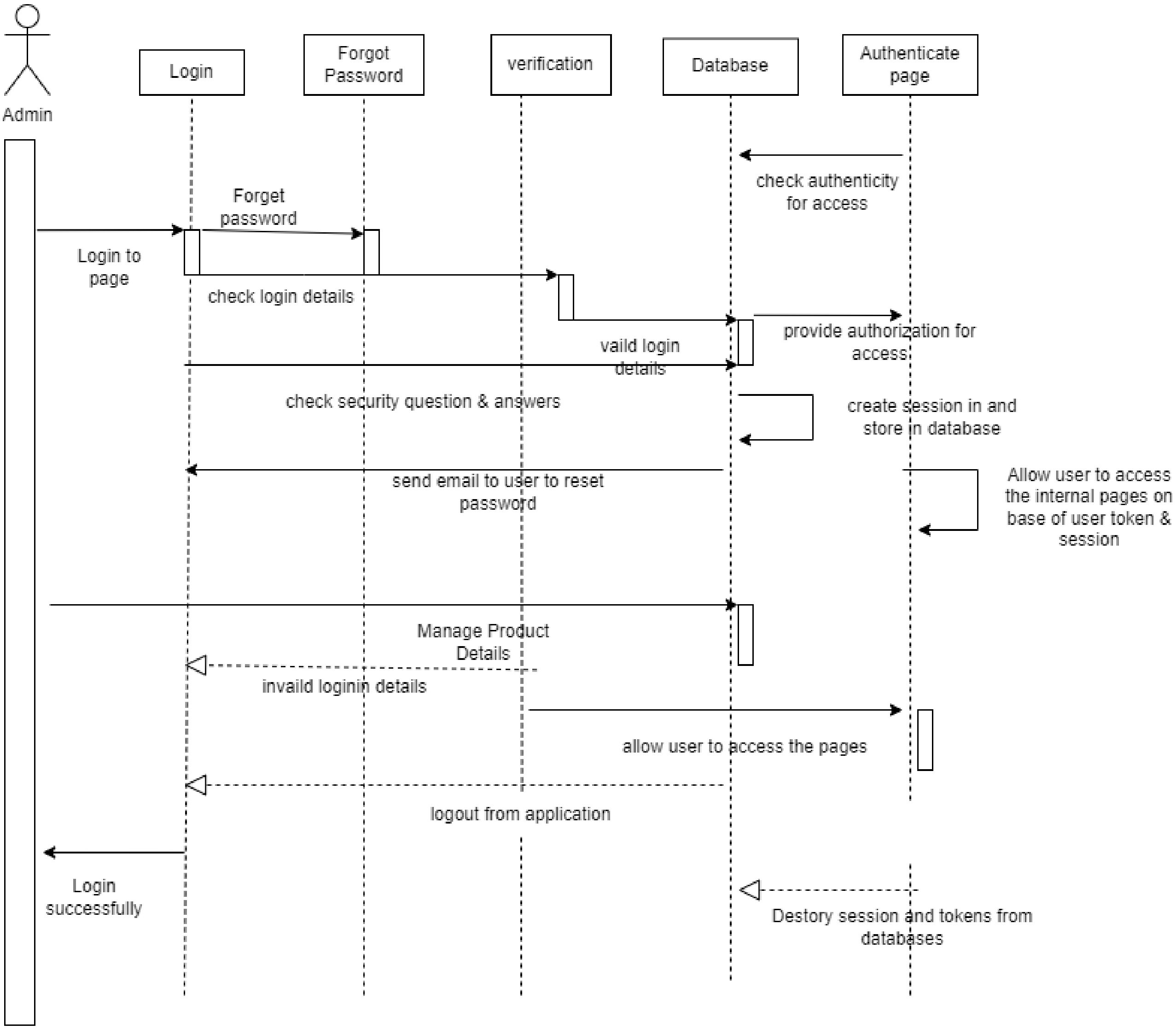
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message

- Reply Message
- Found Message
- Lost Message

iv. **Guards –** To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

Uses of sequence diagrams –

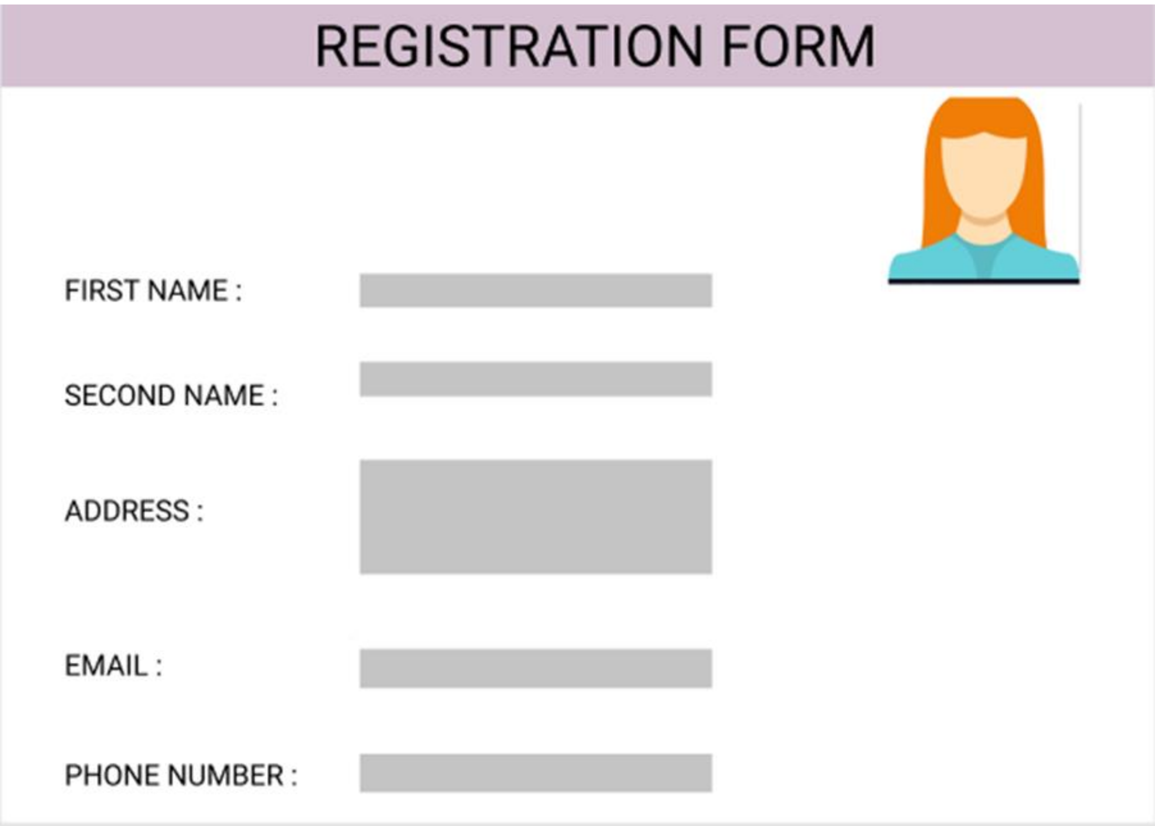
- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.


Admin
Login
Forgot Password
verification
Database
Authenticate page
check authenticity for access
Forget password
Login to page
check login details
provide authorization for access
vaild login details
check security question & answers
create session in and store in database
send email to user to reset password
Allow user to access the internal pages on base of user token & session
Manage Product Details
invaild loginin details
allow user to access the pages
logout from application
Login successfully
Destory session and tokens from databases

4.5 USER INTERFACE DESIGN

4.5.1-INPUT DESIGN

Form Name : User Registration



REGISTRATION FORM

FIRST NAME :

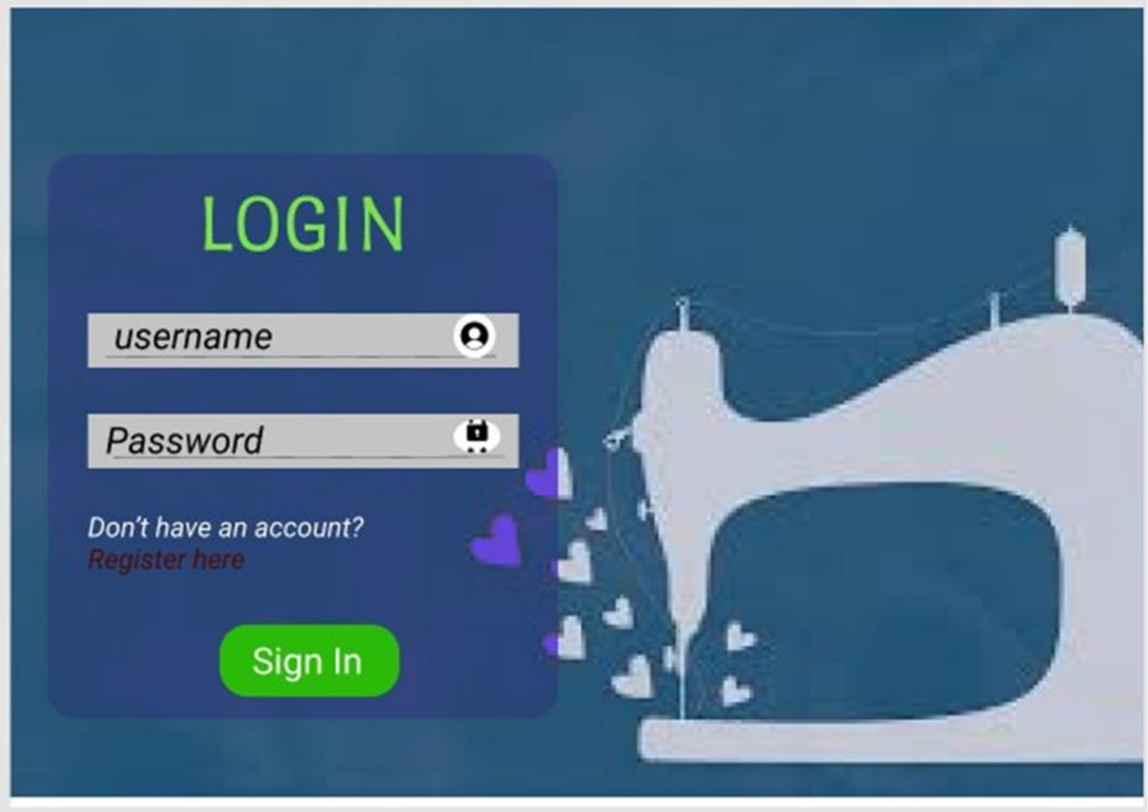
SECOND NAME :

ADDRESS :

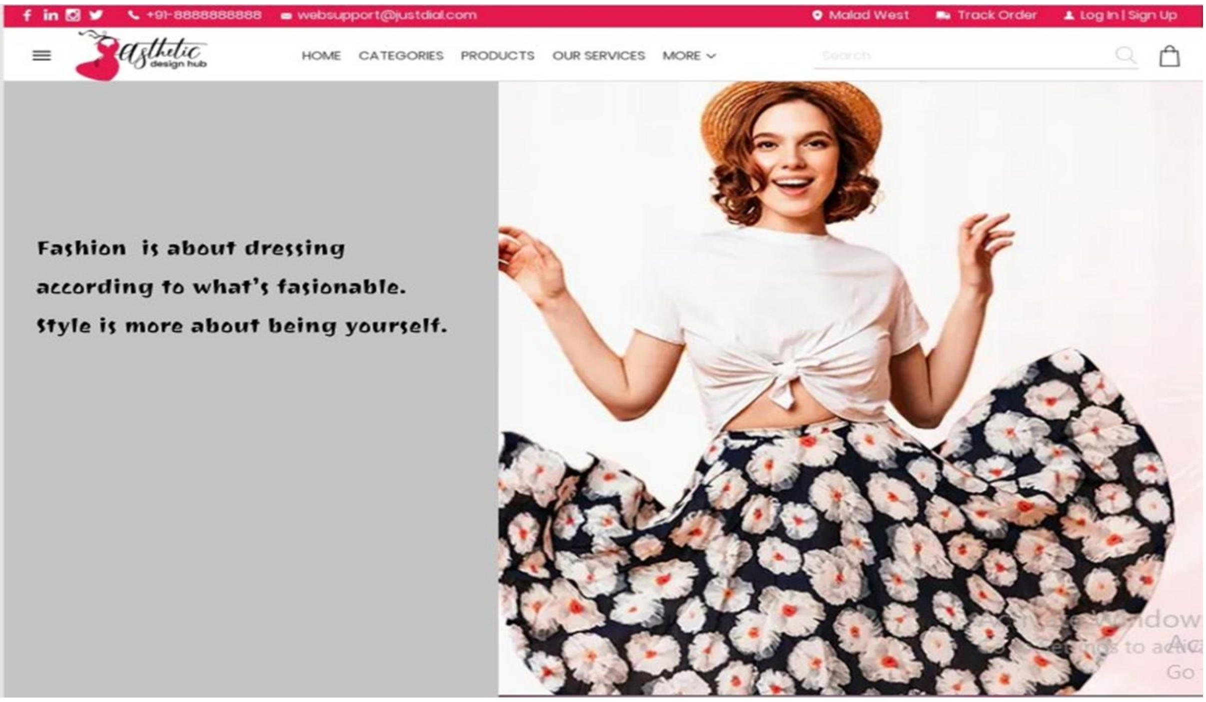
EMAIL :

PHONE NUMBER :

Form Name : User Login

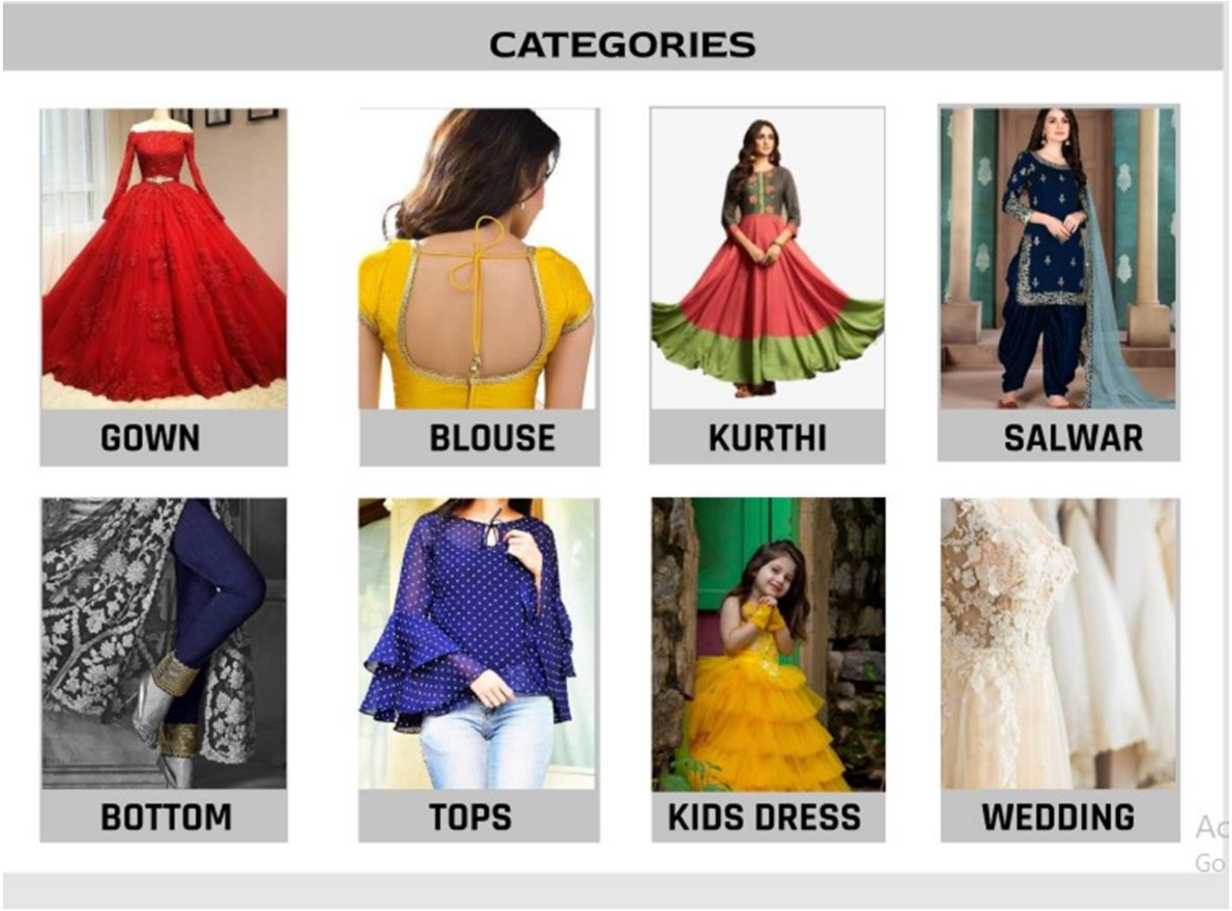


Form Name : Home Page



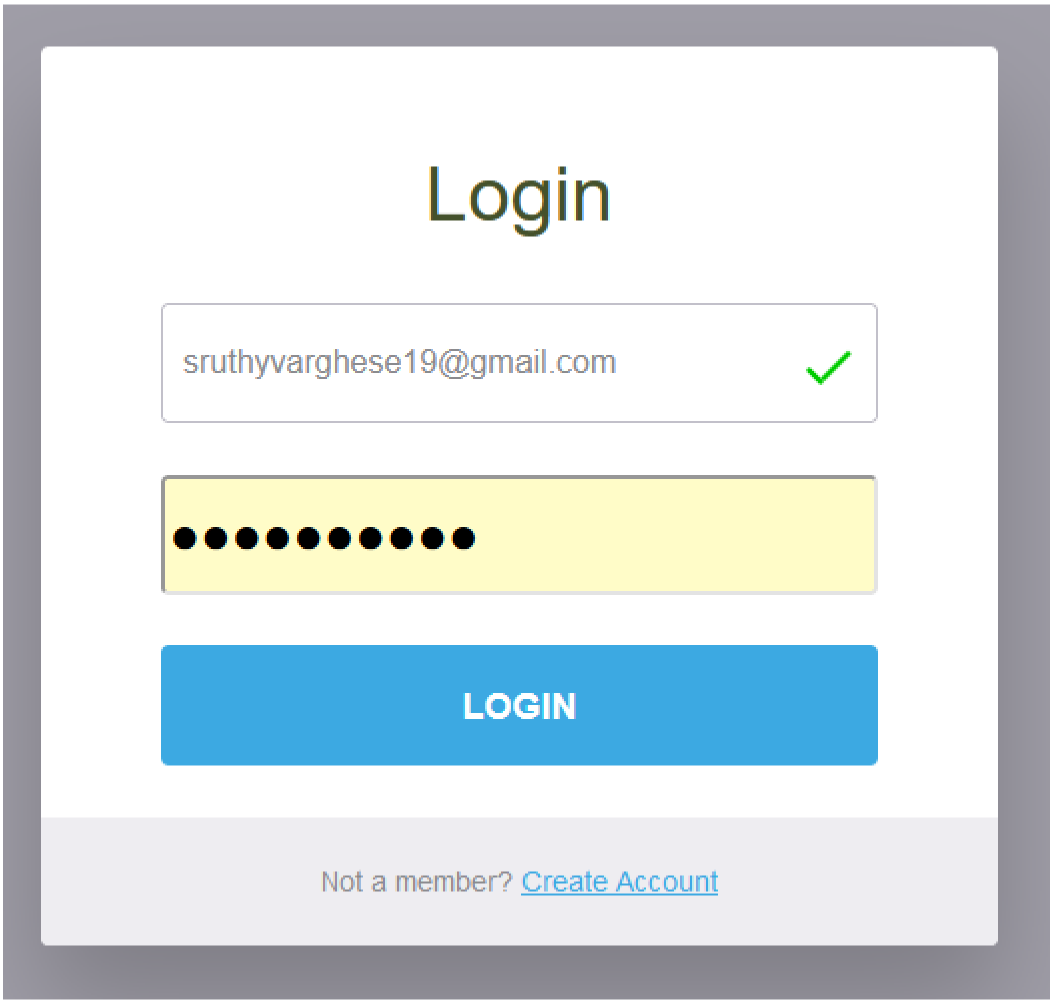
Fashion is about dressing according to what's fasionable. Style is more about being yourself.

Form Name :



4.5.2 OUTPUT DESIGN

User Login



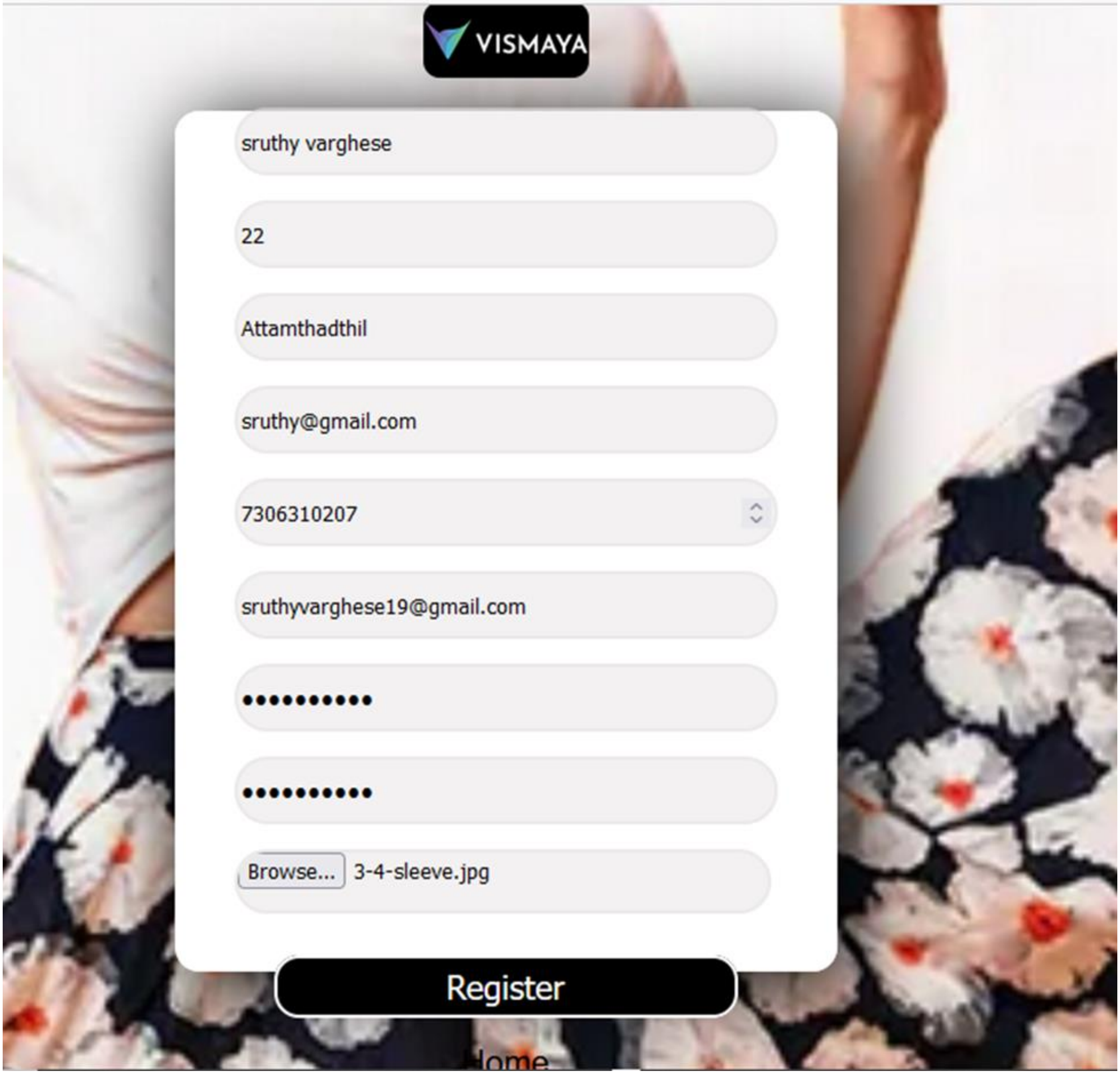
Login

sruthyvarghese19@gmail.com ✓

●●●●●●●●●●

Not a member? Create Account

User Registration



sruthy varghese

22

Attamthadthil

sruthy@gmail.com

7306310207

sruthyvarghese19@gmail.com

••••••••••

••••••••••

Browse... 3-4-sleeve.jpg

4.6. DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN

Login

Primary key **: login_id**

Field	**Type**	**Constraints**	**Descriptions**
login_id	int(10)	primary key	unique number to identify
reg_id	int(10)	foreign key	
username	varchar(20)	not null	
password	varchar(20)	not null	

register

Primary key : **regid**

Foreign key **u_id** references table **login**

Field	**Type**	**Constraints**	**Descriptions**
reg_id	int(10)	primary key	unique number to identify
u_id	int(10)	foreign key	
Name	varchar(20)	not null	
Address	varchar(25)	not null	
Email	varchar(20)	not null	
Phone_no	numeric(12)	not null	
Photo	varchar(25)	not null	

Pro_category

Primary Key : cat_id

Field	DataType	Constraints	Descriptions
cat_id	int	primary key	uniquely identify
cat_name	varchar(20)	not null	category name
cat_status	int(3)	not null	category status
cat_image	varchar(12)	not null	category images

Pro_details

Primary key : pro_id

Foreign key cat_id references table category

Field	DataType	Constraints	Descriptions
pro_id	Int	Primary key	Uniquely identify
cat_id	Int(3)	Foreign key	Category id
pro_name	varchar(20)	not null	Product name
pro_price	numeric(10)	not null	Product price
pro_img	varchar(20)	not null	Product image
pro_dc	numeric(12)	not null	Product discount
Pro_des	varchar(25)	not null	Product descriptions

Fabric_details

Primary key **fabric_id**

Field	**DataType**	**Constraints**	**Descriptions**
Fabric_id	int	primary key	Uniquely identify
pro_id	int(3)	Foreign key	Product id
Fabric_pattern	varchar(10)	not null	
Fabric_color	varchar(10)	not null	
Fabric_price	numeric(5)	not null	
Fabric_material	varchar(10)	not null	

Order_details

Primary key **order_id**

Field	**DataType**	**constraints**	**Descriptions**
order_id	int	Primary key	uniquely identify
pro_id	Int(3)	foreign key	product id
status	varchar(3)	not null	Status

measurements

Primary key :

Foreign key **measu_id**

Field	**DataType**	**Constraints**	**Descriptions**
measu_id	int(10)	primary key	unique number to identify
cat_id	int(10)	foreign key	Category Id
measu_size	varchar(20)	not null	Brand size
measu_bust	varchar(15)	not null	Bust
measu_bottomwaist	int(20)	not null	Bottom Waist
measu_topwaist	int(15)	not null	Top Waist
measu_hips	int(25)	not null	Hips
measu_shoulder	int(20)	not null	Shoulder
measu_bottomlen	Int(15)	not null	Bottom Length

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm’s execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code where removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- □ The active user must be aware of the benefits of using the new system.
- □ Their confidence in the software is built up.
- □ Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of theirrole in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to book service online and view all information.

CHAPTER 8

BIBLIOGRAPHY

BOOKS REFERED	AUTHOR
SYSTEM ANALYSIS AND DESIGN	: ELIAS MAWAD
ANALYSIS AND DESIGN OF INFORMATION SYSTEM	: JAMES ASENN
INTRODUCTION HYPER TEXT MARKUP LAGUAGE	: NAVEEN PRAKASH

WEBSITES:

www.google.com

www.yahoo.com

www.wikipedia.com

www.jquery.com

CHAPTER 9

APPENDIX

9.1 Sample Code

Login.html

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Login Form </title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
<link rel="stylesheet" type="text/css" href="login.css">
</head>
<body>
<div id="login-form-wrap">
  <h2>Login</h2>
  <form id="login-form">
    <p>
    <input type="text" id="username" name="username" placeholder="Username" required><i
class="validation"><span></span><span></span></i>
    </p>
    <p>
    <input type="password" id="password" name="email" placeholder="password" required><i
class="validation"><span></span><span></span></i>
    </p>
    <p>
    <input type="submit" id="login" value="Login">
    </p>
  </form>
  <div id="create-account-wrap">
    <p>Not a member? <a href="reg.html">Create Account</a><p>
  </div>
</div>
```

```
</body>
</html>
```

Login.php

```
<?php
include 'connection.php';
session_start();

if(isset($_POST['submit'])){
  $username=$_POST['username'];
  $password=$_POST['password'];

  $sql="SELECT * FROM login where username ='$username' AND password = '$password'";
  $result =mysqli_query($con,$sql);
  if($result){
    if($row=mysqli_fetch_array($result)){
  if($row[3]=="admin"){
    ?>
  <script type="text/Javascript">
    window.location.href="admin_index.php";
   </script>
   <?php
  }else if($row[3]=="user"){
    $_SESSION['user']=$row['loginid'];



    ?>
  <script type="text/Javascript">
    window.location.href="user_index.php";
   </script>

    <?php
  }
  else
{
  echo"Invalid Username and Password";
}
}
  }
}
?>

<!DOCTYPE html>
<html lang="en" >
<head>
  <meta charset="UTF-8">
  <title>Login Form </title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
<link rel="stylesheet" type="text/css" href="login.css">
</head>
<body>
<div id="login-form-wrap">
  <h2>Login</h2>
  <form id="login-form" method="post" action="">
    <p>
    <input type="text" id="username" name="username" placeholder="Username" required><i
class="validation"><span></span><span></span></i>
    </p>
    <p>
    <input type="password" id="password" name="password" placeholder="password" required><i
class="validation"><span></span><span></span></i>
    </p>
    <p>
    <input type="submit" id="login" name="submit" value="Login">
    </p>
  </form>
  <div id="create-account-wrap">
    <p>Not a member? <a href="register.php">Create Account</a><p>
  </div>
</div>
```

```
</body>
</html>
```

Addparts.php

```
<!DOCTYPE html>
<html lang="en">
  <head>
     <meta charset="utf-8">
     <title>Vismaya</title>
     <meta content="width=device-width, initial-scale=1.0" name="viewport">
     <meta content="Free Website Template" name="keywords">
     <meta content="Free Website Template" name="description">

     <!-- Favicon -->
     <link href="img/favicon.ico" rel="icon">

     <!-- Google Font -->
     <link
href="https://fonts.googleapis.com/css2?family=Barlow:wght@400;500;600;700;800;900&di
splay=swap" rel="stylesheet">

     <!-- CSS Libraries -->
     <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
rel="stylesheet">
     <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">
     <link href="lib/flaticon/font/flaticon.css" rel="stylesheet">
     <link href="lib/animate/animate.min.css" rel="stylesheet">
     <link href="lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">

     <!-- Template Stylesheet -->
     <link href="css/style.css" rel="stylesheet">
  </head>
```

```
<body>
    <!-- Top Bar Start -->
    <div class="top-bar">
        <div class="container">
            <div class="row align-items-center">
                <div class="col-lg-4 col-md-12">
                    <div class="logo">
                        <a href="index.html">
                            <img src="img/vismaya.png" alt="Logo">
                        </a>
                    </div>
                </div>
                        <div class="col-4">
                            <div class="top-bar-item">
                                <div class="top-bar-icon">
                                    <i class="fa fa-phone-alt"></i>
                                </div>
                                <div class="top-bar-text">
                                    <h3>Call Us</h3>
                                    <p>+7306310207</p>
                                </div>
                            </div>
                        </div>


                  </div>
                </div>
            </div>
        </div>
    </div>
    <!-- Top Bar End -->

    <!-- Nav Bar Start -->
    <div class="nav-bar">
        <div class="container">
```

```
            <nav class="navbar navbar-expand-lg bg-dark navbar-dark">
              <a href="#" class="navbar-brand">MENU</a>
              <button type="button" class="navbar-toggler" data-toggle="collapse" data-
target="#navbarCollapse">
                <span class="navbar-toggler-icon"></span>
              </button>

              <div class="collapse navbar-collapse justify-content-between"
id="navbarCollapse">
                <div class="navbar-nav mr-auto">
                  <a href="admin_index.php" class="nav-item nav-link active">Home</a>
                   <a href="addorder.php" class="nav-item nav-link">Add Order</a>
                   <a href="addmeasurment.php" class="nav-item nav-link">Add
Measurment</a>
                   <a href="logout.php" class="nav-item nav-link">Logout</a>
                </div>
              </div>
            </nav>
          </div>
        </div>
        <!-- Nav Bar End -->
        <?php
require_once "connection.php";
if(isset($_POST['submit'])){

    $title = $_POST['title'];
    $type=$_POST['type'];
    $description=$_POST["description"];
    $image=$_POST['image'];

    $sql = "INSERT INTO part(title,type,description,image)
values('$title','$type','$description','$image')";
    $result = mysqli_query($con, $sql);
    if($result){
        echo "<script>";
            echo 'alert("Inserted Successfully")';
        echo "</script>";
    }else{
        echo "<script>";
            echo 'alert("Failed to insert")';
        echo "</script>";
    }

}

?>
<!DOCTYPE html>
<html>
<head>
    <title>Insert parts</title>
    <link rel="stylesheet" href="./css/style.css">
</head>
<style>
    body{
      background-image: url("img/pic2.jpg");
      background-attachment: fixed;
      background-position: center;
      background-repeat: no-repeat;
      background-size: cover;

    }
  </style>
<body>
```

```
<section>
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-9">
            <h4 class="head-tag">Insert Measurment Parts</h4>
                <div class="container">
                    <div class="row">
                        <div class="col-1"></div>
                        <div class="col-10 panelBack">
                            <form action="partadd.php" method="POST" enctype="multipart/form-data"><br><Br>
                                <input type="text" name="title" class="input-txt" placeholder="Enter title"><br><br>
                                <input type="text" name="type" class="input-txt" placeholder="Enter type"><br><br>
                                <input type="text" name="description" class="input-txt" placeholder="Enter description"><br><br>
                                <input type="file" name="image"class="input-txt" placeholder="Upload image"><br><br>



                                <input type="submit" name="submit" value="Insert" class="btn btn-primary" style="margin-left: 70%;">
                            </form>
                        </div>
                        <div class="col-1"></div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>
```

```
</body>
</html>
```

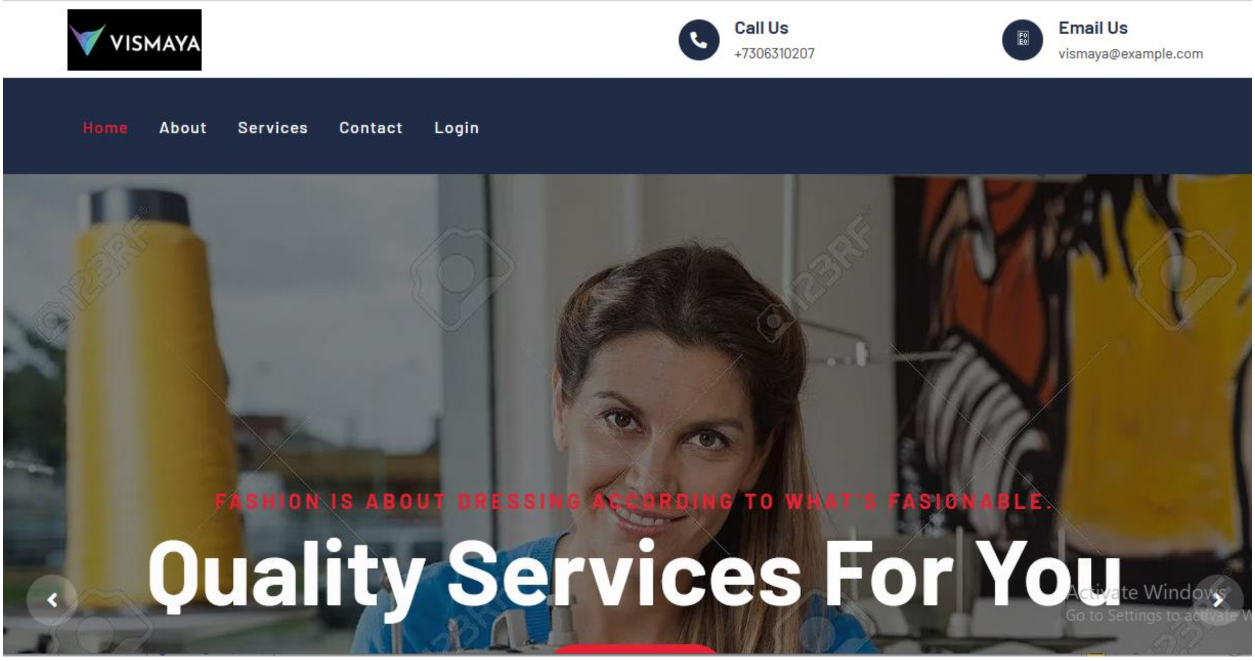
partedit.php

```
<?php
    include 'connection.php';
    $id=$_GET['id'];
    $sql="select *from `part` where id='$id'";
    $result=mysqli_query($con,$sql);
   $row=mysqli_fetch_assoc($result);
    $title=$row['title'];
    $type = $row['type'];
    $description = $row['description'];
    $image = $row['image'];



    if(isset($_POST['s']))
    {
    $officer_id=$_GET['upid'];
    $officer_name = $_POST['name'];
    $o_policestation = $_POST['station'];
    $officer_rank = $_POST['rank'];
    $officer_location = $_POST['location'];
    $officer_email = $_POST['email'];
    $officer_password = $_POST['password'];



        mysqli_query($con,"UPDATE `part` SET
        'id`='$id',`title`='$title',`type`='$type',`description`='$description',`image`='$image' where
id='$id'");

            echo "<script>alert('Updated');</script>";
            header('location: measurment_parts.php');
    }
 ?>

 <!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body{
  font-family: Calibri, Helvetica, sans-serif;
  background-color: pink;
}
.container {
    padding: 50px;
  background-color: white;
}

input[type=text], input[type=password], textarea {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}
input[type=text]:focus, input[type=password]:focus {
  background-color: orange;
  outline: none;
}
 div {
            padding: 10px 0;
         }
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}
.registerbtn {
  background-color: #4CAF50;
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}
.registerbtn:hover {
  opacity: 1;
}
</style>
<style> form{
  padding: 150px 170px;
}</style>
</head>
<body>
<form method='post' action="" >

  <div class="container">
  <center>  <h1>update </h1> </center>
  <hr>
  <div class="modal-header">

                         </div>
                     <div class="modal-body">
                    <div class="card-body card-block">
                    <div class="form-group">
                       <label for="company" class=" form-control-label">Title:</label>
```

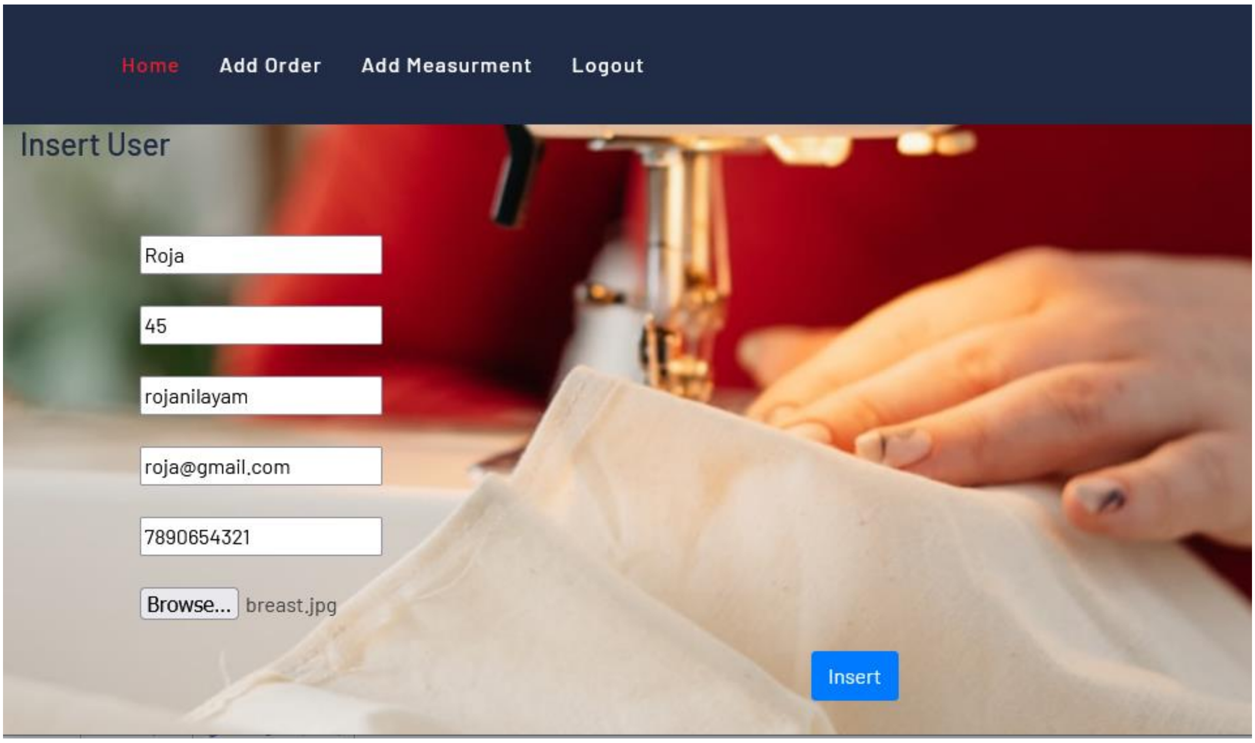
```
                <input type="text"   class="form-control" name="title"  id="title"
onfocusout="f1()" value=<?php echo $title?>>
              </div>
              <div class="form-group">
                    <label for="company" class=" form-control-label">Type </label>
                 <input type="text"  class="form-control" name="type" id="type"
onfocusout="f1()" value=<?php echo $type?>>
              </div>
              <div class="form-group">
                    <label for="company" class=" form-control-label">Description</label>
                 <input type="text"  class="form-control" name="description"  id="description"
onfocusout="f1()" value=<?php echo $description?>>
              </div>
              <div class="form-group">
                    <label for="company" class=" form-control-label">image</label>
                 <input type="file"  class="form-control" name="label" id="image"
onfocusout="f1()" value=<?php echo $image?>>
              </div>

           <div class="modal-footer">
              <button type="button" class="btn btn-danger" data-dismiss="modal" style="margin-
right: 66%">Cancel</button>
                 <button type="submit" class="btn btn-primary" name="s">Update</button>
           </div>

           </div>
              </div>

</form>
</body>
</html>
```

9.2 Screen Shots

Home page



Add customers



Insert User

Roja

45

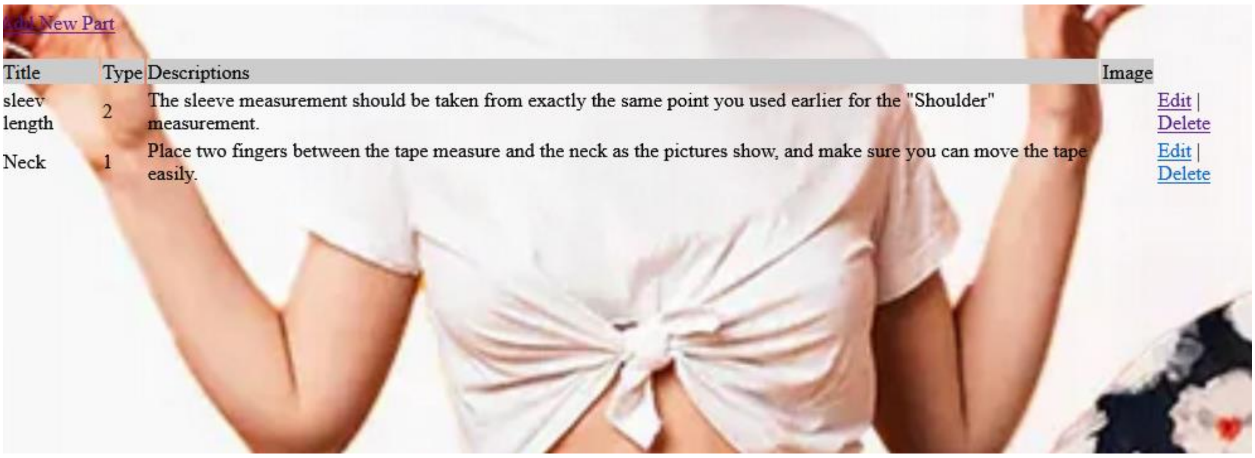
rojanilayam

roja@gmail.com

7890654321

breast.jpg

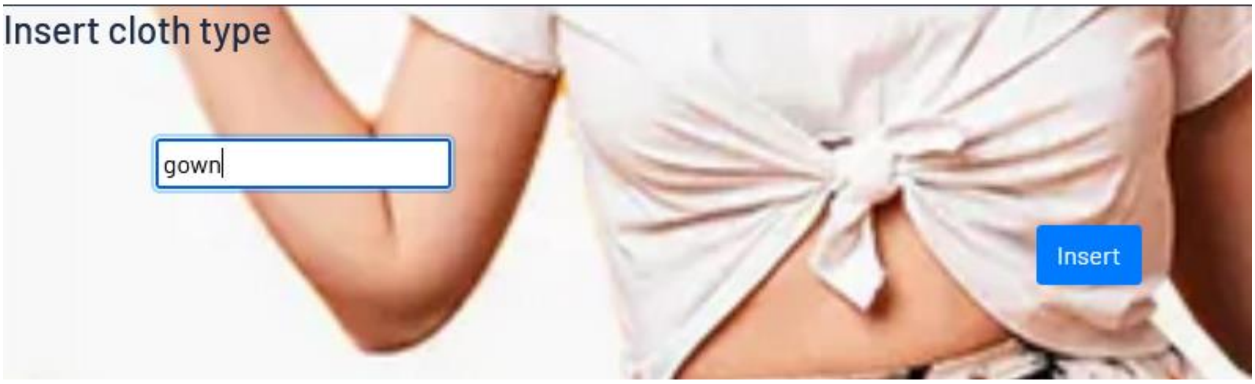
Manage measurment



Add New Part

Title	Type	Descriptions	Image	
sleev length	2	The sleeve measurement should be taken from exactly the same point you used earlier for the "Shoulder" measurement.		Edit | Delete
Neck	1	Place two fingers between the tape measure and the neck as the pictures show, and make sure you can move the tape easily.		Edit | Delete

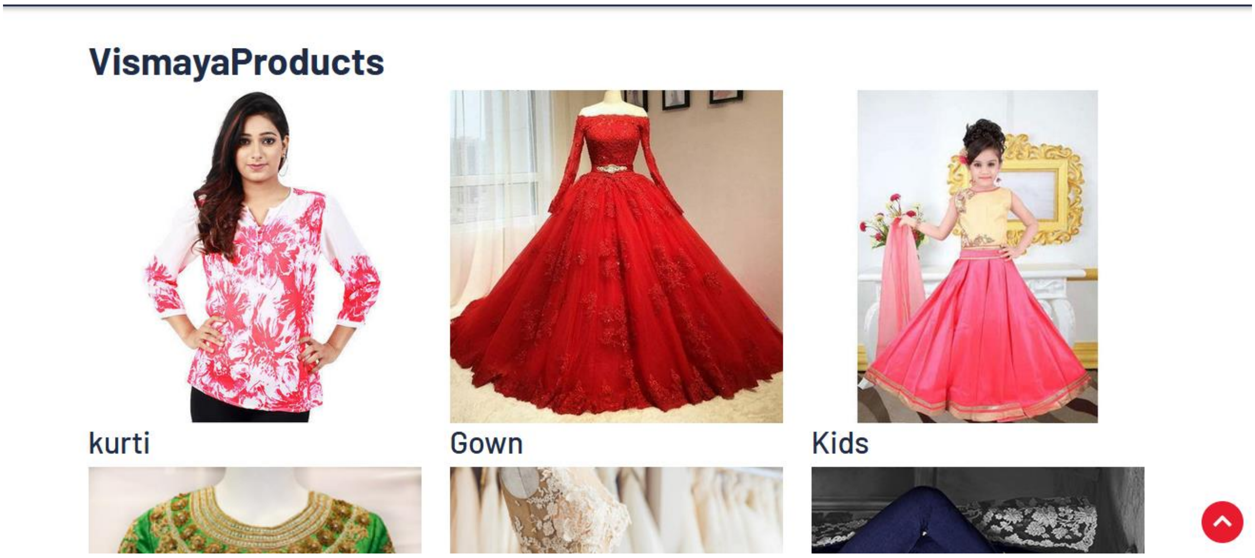
Add cloth type



Insert cloth type

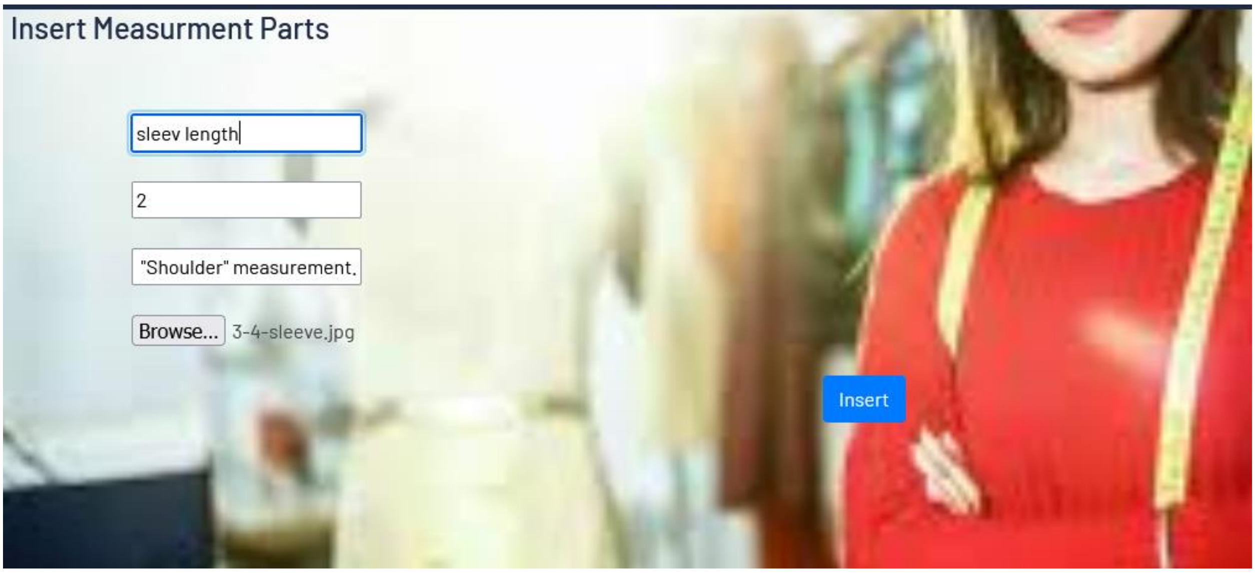
gown

Products



VismayaProducts

Insert Measurment Parts



Insert Measurment Parts

sleev length

2

"Shoulder" measurement.