Chapter 2: Analysis

**Introduction**

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. The technique has been applied in the study of mathematics and logic since before Aristotle (384–322 B.C.), though analysis as a formal concept is a relatively recent development. It is the systematic study of real and complex-valued continuous functions. It describes both the discipline of which calculus is a part and one form of the abstract logic theory.

Here, in this project I am developing an Online Clothes Shopping which is a web based programmed.

**Feasibility study**

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.

Some of the feasibility study are:

Technical Feasibility:

Validating that a given technology can support requirements or that a goal is technically possible. For example, an ecommerce project confirms that a partner's API can support a list of requirements for an integration project.

Schedule Feasibility:

Validating that a goal is possible with time constraints. For example, a fashion label discovers a trendy new fashion accessory at a fashion week. They do a feasibility check to see if they can produce and distribute the item in time for the Spring/Summer season.

Economic Feasibility:

Given the financial resources of the company, is the project something that can be completed? The economic feasibility study is more commonly called the cost/benefit analysis.

Legal/Ethical Feasibility:

What are the legal implications of the project? What sort of ethical considerations are there? You need to make sure that any project undertaken will meet all legal and ethical requirements before the project is on the table.

Operational Feasibility:

This measures how well your company will be able to solve problems and take advantage of opportunities that are presented during the course of the project.

**Requirement analysis:**

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. Requirements analysis means to analyze, document, validate and manage software or system requirements.

1. **Functional Requirements:**

A functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

Some of the functional requirements that are needed in my project are listed bellow

1. User
2. User Registration:

New user can register by creating a new account.

1. Login:

User must have valid account to login to site.

1. View and edit own details:

User can view and edit their personal details, payment method.

1. Purchasing:

User can purchase any product.

1. Giving feedback:

User can provide feedback about the product to customer care service center about their impression for the site and services.

1. Logout:

Customer must logout of the site after purchasing.

1. Visitors
2. Visiting the site:

Visitors can visit the site without registration.

1. Register
2. Customer care
3. Getting feedback from customer:

Responsible for receiving complaints and feedback from customer.

1. Providing solutions to customers:

Provide feasible solution to the customer on their complaints.

1. **Non-functional:**

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

Some of the non-functional requirements are as follows

1. Performance:

Performance is a quality attribute that describes the responsiveness of the system to various user interactions with it. Poor performance leads to negative user experience. It also jeopardizes system safety when it’s is overloaded.

1. Security:

Security requirements ensure that the software is protected from unauthorized access to the system and its stored data. It considers different levels of authorization and authentication across different user’s roles.

1. Response time:

1. Usability:

Usability defines how difficult it will be for a user to learn and operate the system. Usability can be assessed from different points of view like Efficiency of use, low perceived workload and intuitiveness.

1. Reliability:

Reliability defines how likely it is for the software to work without failure for a given period of time. Reliability decreases because of bugs in the code, hardware failures, or problems with other system components.

1. Availability:

Availability is gauged by the period of time that the system’s functionality and services are available for use with all operations. So, scheduled maintenance periods directly influence this parameter. And it’s important to define how the impact of maintenance can be minimized.

1. Maintainability
2. Backup
3. **Moscow prioritization:**

Moscow prioritization is a prioritization technique used in management, business analysis, project management, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement. The letters stand for:

* Must Have
* Should Have
* Could Have
* Won’t Have

Functional Requirement

|  |  |  |
| --- | --- | --- |
| S.N | Functional Requirement | Moscow Prioritization |
| 1. User | Registration | Must Have |
| Login | Must Have |
| View and edit own details | Must Have |
| Purchasing | Could Have |
| Giving feedback | Could Have |
| Logout | Should Have |
| 2. Visitors | Visiting the site | Could Have |
| Register | Could Have |
| 3. Customer Care | Getting feedback from customer | Could Have |
| Providing solutions to customers | Must Have |

Non-functional Requirements:

|  |  |  |
| --- | --- | --- |
| S.N | Non-Functional requirements | Moscow Prioritization |
| 1 | Performance | Could Have |
| 2 | Security | Should Have |
| 3 | Response Time | Could Have |
| 4 | Usability | Could have |
| 5 | Reliability | Should have |
| 6 | Availability | Could have |
| 7 | Maintainability | Could have |
| 8 | Backup | Should Have |

1. **Software requirements specification (SRS):**

A software requirements specification is a description of a software system to be developed. It is created based on the demand of user and relating all the workers. It may incorporate for the instance use of how customer will work together with programming system. The product essential determination of each and every fundamental need for venture improvement. To develop the product framework, we should have clear perception of a software framework. To achieve this, we need to incessant correspondence with customers to collect all necessities.

**Hardware requirements specification:**

Operating system: windows 7 or above

Processor: Dual Core Intel Pentium (Xeon) family or higher recommended

RAM: 4 GB RAM

HDD: 40 GB or more recommended

**System Architecture**

A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers.



Fig: 3-tier architecture

**Use case:**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved

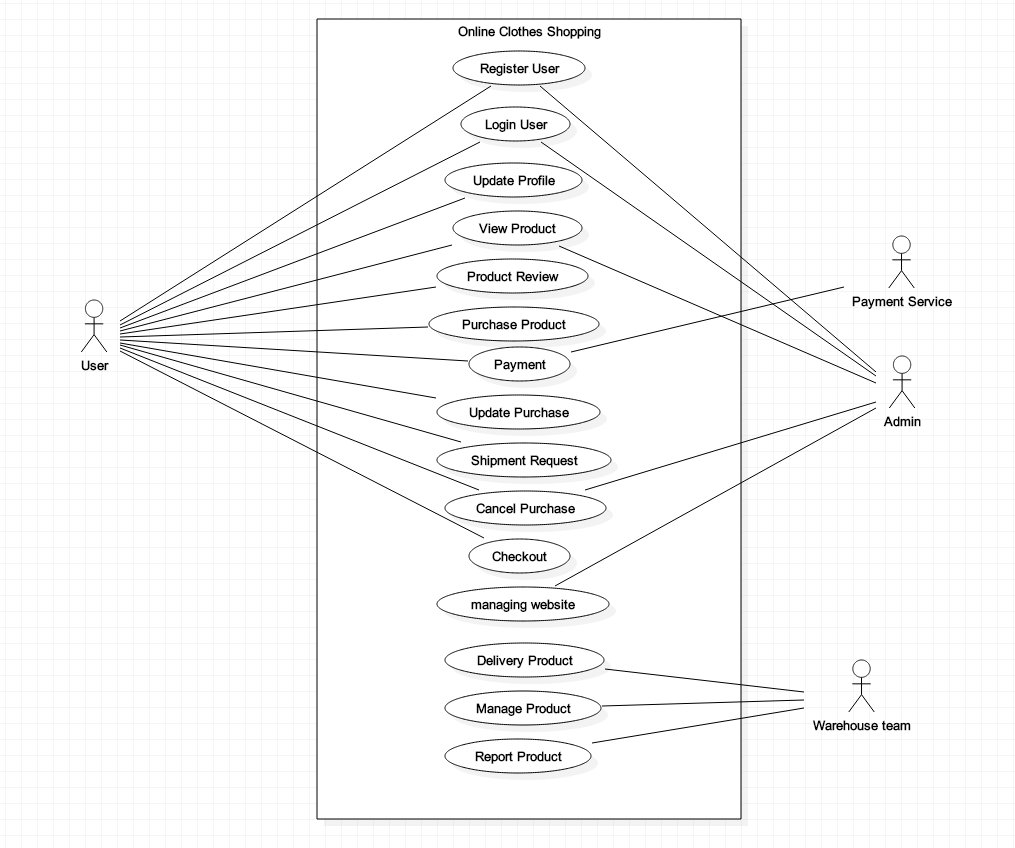


Fig: Use case diagram of online clothes shopping

**Initial class diagram**

A class diagram is a type of diagram and part of a unified modeling language (UML) that defines and provides the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes.

Natural Language Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Nouns | Selected/Rejected | Justification |
|  | Customer | No | It is rejected because a similar noun has been selected as a candidate class. |
|  | Clothes | Yes | It is the main aspect of this project which is feasible to be a class. |
|  | Brands | Yes | It is connected with the project and has important role. |
|  | Web portal | No | Immaterial to the system. |
|  | Product | Yes | An important aspect to the system. |
|  | Website | No | It itself is a complete package. |
|  | Delivery charge | Yes | A part of the project which has an important role. |
|  | Company | No | It does not contribute anything to the system. |
|  | User | Yes | An important aspect of the system. |
|  | System | No | A system cannot be used as class. |
|  | Shop | Yes | It is also an important aspect of the system. |
|  | Price | Yes | It contributes in the system. |
|  | Time | No | Immaterial to the system. |

**Class Diagram**

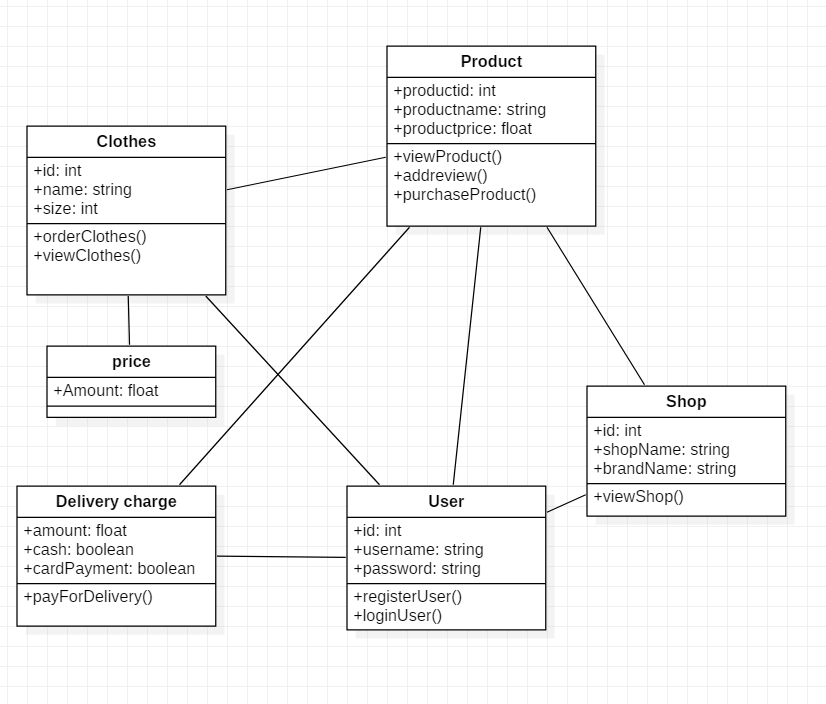


Fig: Class Diagram

**Conclusion**

Finally, I have completed the analysis. In this system user should allow to browse through different products. It saves time a lot and can view the details information about the order. In this project I have used PHP for programming language, software XAMPP server and MySQL for database. User should allow review the order, see the menu list, remove, add and many more.