

E-COMMERCE WEB SITE

1. BACHU BALA RAJU	-	18BD1A1204
2. VISWATEJ BHAMIDIPATI	-	18BD1A1208
3. GADILA SAI NIKSHAY REDDY	-	18BD1A1216
4. SWARNA MADHU BABU	-	18BD1A1254
5. BANGARU HARI PRASAD	-	19BD5A1202
6. MOHAMMED MUNAVAR	-	18BD1A1236

Problem Statement:

E-commerce provides an easy way to buy and sell products to a large number of customers. When users land on an ecommerce site they expect to find what they are looking for quickly and easily. Users are not sure about the brands or the actual products they want to purchase. Many customers nowadays search for their products on Google rather than visiting specific e-commerce sites. They believe that Google will take them to the e-commerce sites that have their product.

suppose a customer is interested in purchasing a mobile. His or her search for a mobile should list mobile brands, operating systems on mobiles, screen size of mobiles, and all other features. As the customer selects more and more features or options from the facets provided, the search narrows down to a small list of mobiles that suit his or her choice. If the list is small enough and the customer likes one of the mobiles listed, he or she will make the purchase.

The main problem is that categorization and feature listing of products should be taken care of. Misrepresentation of features can lead to incorrect search results. Another problem is that we need

to provide multiple products in the search results. For example, while displaying the list of all mobiles, we need to provide products of a brand. Once a brand is selected, another set of results for operating systems, network, and mobile phone features has to be provided. As more and more features are selected, we still need to show features within the remaining products.

It is also important that products are added to the index as soon as they are available. It is even more important that they are removed from the index or marked as sold out as soon as their stock is exhausted. For this, modifications to the index should be immediately visible in the search. And adding items to cart is another feature which helps in buying products at once.

We must also prevent data breaching. Data from the user must be kept private from data theft. The site must be capable of getting data from multiple types of databases like from sql and non-sql dbms.

Hence, The user interface should be interactive, responsive and easy to handle, since the user can use/handle the application with ease and reach his requirements in no time.

Software Requirement Specification

For

E-Commerce Web Site

Version 1.0 approved

Prepared by:

- | | | |
|----------------------------|---|------------|
| 1.Bachu Bala Raju | - | 18BD1A1204 |
| 2.Viswatej Bhamidipati | - | 18BD1A1208 |
| 3.Gadila Sai Nikshay Reddy | - | 18BD1A1216 |
| 4.Swarna Madhu Babu | - | 18BD1A1254 |
| 5.Bangar Hari Prasad | - | 19BD5A1202 |
| 6. Mohammed Munawar | - | 18BD1A1236 |

Keshav Memorial Institute of Technology

11-09-2020.

Table of Contents

Table of Contents
Revision History
1. Introduction
1.1 Purpose.....
1.2 Document Conventions.....
1.3 Intended Audience and Reading Suggestions.....
1.4 Product Scope
1.5 References.....
2. Overall Description
2.1 Product Perspective.....
2.2 Product Functions
2.3 User Classes and Characteristics
2.4 Operating Environment.....
2.5 Design and Implementation Constraints
2.6 User Documentation
2.7 Assumptions and Dependencies
3. External Interface Requirements
3.1 User Interfaces
3.2 Hardware Interfaces
3.3 Software Interfaces
3.4 Communications Interfaces
4. System Features
4.1 System Feature 1.....
4.2 System Feature 2 (and so on).....
5. Other Nonfunctional Requirements.....
5.1 Performance Requirements
5.2 Safety Requirements
5.3 Security Requirements
5.4 Software Quality Attributes
5.5 Business Rules
6. Other Requirements
Appendix A: Glossary.....
Appendix B: Analysis Models.....
Appendix C: To Be Determined List.....

Revision History

Name	Date	Reason for changes	Version
Week-1	11-09-20	SRS creation(Introduction)	1.0
Week-2	18-09-20	SRS updation	2.0
Week-3	25-09-20	SDLC,UML,Use-Case Diagram and Template	3.0
Week-4	2-10-2020	Sequential and collaboration diagrams	4.0
Week-5	9-10-2020	Class diagram	5.0

1.Introduction

The Software Requirement Specification is designed to document and describe the agreement between the customer and developer regarding the specification of the software product requested.

This documentation is done to provide a clear idea of customer requirements. This document can be used as reference in further development of the software system.

1.1 Purpose:

This is the software Requirement Specification for our ECOMMERCE web-site .Our project is all about e-shopping. It will facilitate the user to purchase things without going to store(offline). We will give the facility to merchants to upload their products on the store for sale. We make the entire exchange between the buyer and seller very easy.

1.2 Document Convention:

Heading:

Font-Size:16

Font-Style: Bold

Font: Times New Roman

Subheading:

Font-Size:14

Font-Style: Bold

Font: Times New Roman

Content:

Font-Size:12

Font: Times New Roman.

1.3 Intended Audience and Reading Suggestions:

This document is a prototype for the E-COMMERCE Website. This document will be useful for different audiences in various ways. Like for developers to design and implement the project .managers for maintaining cost and time. Advertisers for making the best advertisement about the features and how it is different from other e-commerce sites. Users can know if the application is up to reach (i. e whether he is satisfied or not) and he can make any changes if needed. Testers can test according to this document. Also functional and non-functional features are very helpful for developers.

1.4 Product Scope:

Our project is similar to other e-commerce websites with some new advancements. Our online shopping site will be available for 24hrs and 7days in a week. There will be maintenance once in a month. Our primary objective is to satisfy the customer. Through online shopping he will save a lot of time. Companies will not need to have extra stock. They are able to make products on demand. We will guide users to buy good products through our online help system. We also ask users for ratings of the seller based on the quality provided by the particular seller so that we can take necessary action on the seller.

User details will be kept safe and secure from data theft. We conduct surveys on how we can make our site better and what changes can be made. We also maintain a database for different users.

1.5 References:

We took references from different websites like Flipkart, Amazon, Snapdeal, Myntra and Walmart etc.

http://users.csc.calpoly.edu/~csturner/courses/308w09/Ch9_DocReqs.pdf

https://www.academia.edu/8831452/Software_Requirements_Specification_for_FSoft_D

2. Overall Description

2.1 Product Perspective

Our project is the follow on e-shopping projects. It will contain most of the functionalities from these e-shopping websites like vendors to setup online shops, customers to browse through shops and administrator to accept and reject new vendors and maintain list of shop categories. It is a Mobile-Friendly website with highly customizable themes. And it is produced with some new advancements like advanced search engine bar for easy searching inserting, high data security for user data from data theft and also cool and interactive user interface to make it comfortable as well as easy for everyone to use.

2.2 Product functions

2.2.1 Administrator

- Admins should be able to insert, modify and delete products.
- Can accept or reject users/vendors according to the site policies and payment methods.
- Add and edit product categories and also arrange them by categories.
- Can able to warn buyers about the discounts and coupons they can avail.
- Can record products returned by the users.

2.2.2 Customers/Users

- The users must be provided with all the information about the new products and discounts available,
- Users are given provision to check and change their account details.

- Users should have the ability to check required products through brand,specs and any information related to that product.
- Can be able to return or exchange the product according to terms and conditions.
- Users must have provision to suggest the products to be brought into the site and review the seller and the product.

2.3 Operating Environment

- This e-commerce website shall operate in all famous browsers,for a model we are taking ,Internet Explorer Versions 7.0,8.0 and 9.0,with flash player 9 and JavaScript.
- It can be opened on window 98,XP,8,10,7 and also on mac Os.
- The processor should be at least Pentium 3 or above.
- Ram should be greater than 512 mb.

2.4 User Characteristics

Users of this website are customers,merchants and administrators who maintain the website.We have system administrator who will maintain overall web store.there are merchants who upload their products on the website.customers who purchase the things.so, customers and merchants assumed to have basic knowledge of computers and Internet browsing.Administrators of the system should have knowledge about the internal modules and are able to rectify problems arise.

2.5 Design and Implementation Constraints

- Users may access this website with a device having full and stable internet connection and internet browsing capabilities.
- The information of the users,products and indexing will be stored in a database that can be accessed by the website.

- This site will be up 24X7.
- Software does not require any specific operating System to run, it can run on any platform even in mobile .
- Users must have their correct username and passwords to enter into their accounts and do actions.

2.6 Assumptions and Dependencies

The website requires following third party products.

- Microsoft SQL server to store the database.
- ASP.net to develop the product.
- JSON is used for extracting data from the database/SQL server.

The success of this E-Commerce website depends on

- Existence of an internet service to all people.
- All Buyers and Sellers comfortable with computers/laptops/phones and have enough knowledge to work with the product.
- E-Commerce websites must be friendly and easy-to-use.
- The search mechanism should be simple , fast and secure.

3. Specific requirements

3.1 Functional requirements

These requirements include the development of search tools, sorting, filtering, navigation, as well as the visual components of the site which can be maintained by the administrator.

3.1.1 Administrator:

Requirement ID : RI.01.01

Title : Database Management

Description : This should Control the entire database and keep track of all records of customers and employee details. If there is any problem in the database (whether it is for accessing or updating) will be solved as soon as possible without delay.

Priority : 2

Requirement ID : RI.01.02

Title : Contact and Giving Permission to Vendors

Description: : Contact with the vendors and give permission to sell their product under the site. Quality should be checked for the first time before selling ,after that the quality can be assessed by the ratings given by the customer, so that we can proceed with the seller

Priority : 2

Requirement ID : RI.01.03

Title : View all details

Description : Able to view the details of all employees and control the whole site.

Priority :3

3.1.2 Customers:

Requirement ID : R1.02.01

Title : Registration

Description : New users should sign up by creating new ID with either mobile number or mail account which is mandatory for placing an order.

Priority :1

Requirement ID : R1.02.02

Title : Login

Description : Customers must have a valid credentials (registered at the time of registration)to enter into the site.

Priority :1

Requirement ID : R1.02.03

Title : View and edit Own Details

Description : Can view/edit his personal details, payment details, status of the order etc.

Priority :2

Requirement ID : R1.02.04

Title : Choosing and comparing products

Description : Can view all available and similar products and can compare them and make a choice for purchasing products.

Priority :4

Requirement ID : R1.02.05

Title : Purchasing

Description : Can purchase any product through a valid internet banking way.

Priority :2

Requirement ID : R1.02.06

Title : Giving Feedback to Customer Care

Description : Customer will be asked to give review to the 24X7 Customer Care Service center about their impression for the quality of the product purchased, site and services.

Priority :4

Requirement ID : R1.02.07

Title : Returns & Refunds

Description : Customers can return their products and can expect the refund .

Priority :3

3.1.3 Shop Owner:

Requirement ID : R1.03.01

Title : Taking Permission from Administrator

Description : Vendors must take permission from the Administrator for selling their products under the site. Administrator will test product's quality according to its market price to permit vendor for selling purpose. Shop owner should update the site management about the quantity of the stock at the time of stock ending and at the time of stock arrival, so that management can add the stock and delete the stock from the indices of the database

Priority :1

Requirement ID : R1.03.02

Title : Advertising Vendor's Own Products

Description : Responsible for making advertisements of his products, but the site will not be responsible for any kind of advertisements about products

Priority :2

Requirement ID : R1.03.03

Title : Getting Feedback from the Customers

Description : Responsible for receiving complaints, queries, handling call to the service executive (if needed) and feedback from the customers.

Priority :3

Requirement ID : R1.03.04

Title : Providing Solutions to Customers

Description : Provide feasible solutions to the customers on their complaints and queries in less time.

Priority :2

3.2 Non-functional Requirements

3.2.1 Performance Requirements

- The system shall accommodate a high number of items and users without any fault.
- Responses to view information shall take no longer than 5 seconds to appear on the screen.

3.2.2 Safety Requirements

- System use shall not cause any harm to human users.

3.2.3 Security Requirements

- System will use secured database
- Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
- System will have different types of users and every user has access constraints.

3.2.4 Error handling

- site shall handle expected and unexpected errors in ways that prevent loss in information and long downtime period.

Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

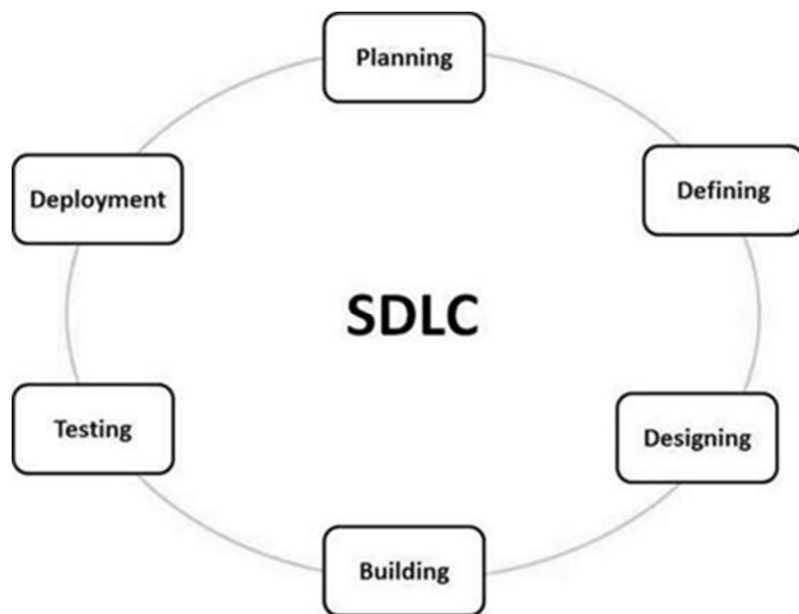
- SDLC is the acronym of Software Development Life Cycle.
- It is also called the Software Development Process.

- SDLC is a framework defining tasks performed at each step in the software development process.
- ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages –

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters such as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any).

The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred to as "Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

Following are the most important and popular SDLC models followed in the industry –

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

SRS Document

1.Introduction

1.1 Purpose of Document

Provide an introductory paragraph explaining the purpose of this document. Its purpose is to explicitly cite all functions that the project shall do. This document is the primary document, upon which the design, source code, and test plan all base their content. This document is used to determine if the final delivered product provides everything that it was supposed to. The Client, User, and Software Engineering representatives often negotiate the content of this document.

1.2 Scope

Provide two paragraphs, the first describing the scope of the product, with the second describing the scope of this document. Remember that "scope" basically means the extent of activity or influence, or range of operation. Be sure that the two paragraphs in this section distinguish between the scope of the product, versus the scope of this document.

You will probably find that in most of the Software Engineering documents that you create in this course, the paragraph for scope of product will be identical (as expected). Specifically for this document, the scope includes all team members and their responsibilities for specifying the product's requirements.

1.3 Objective

A project objective describes the desired results of a project, which often includes a tangible item. An objective is specific and measurable, and must meet time, budget, and quality constraints. ... A project may have one objective, many parallel objectives, or several objectives that must be achieved sequentially.

1.4 Proposed System

The proposed system should have the following features. The transactions should take place in a secured format between various clients in the network. It provides flexibility to the user to transfer the data through the network very easily by compressing the large amount of file.

2. Requirements Specifications

2.1 Functional Requirements

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.

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases. Functional requirements are supported by non-functional requirements (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>," while non-functional requirements take the form "system shall be <requirement>." The plan for implementing functional requirements is detailed in the system design, whereas *non-functional* requirements are detailed in the system architecture.

2.2 Non-Functional Requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. Also known as system qualities, nonfunctional requirements are just as critical as functional Epics, Capabilities, Features, and Stories. They ensure the usability and effectiveness of the entire system. Failing to meet any one of them can result in systems that fail to satisfy internal business, user, or market needs, or that do

not fulfill mandatory requirements imposed by regulatory or standards agencies. In some cases, non-compliance can cause significant legal issues (privacy, security, safety, to name a few).

2.3 Software Requirements

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation package and need to be installed separately before the software is installed.

2.4 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

3. Literature Survey

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project.

It is the most important part of your report as it gives you a direction in the area of your research. It helps you set a goal for your analysis - thus giving you your problem statement.

When you write a literature review in respect of your project, you have to write the researches made by various analysts - their methodology (which is basically their abstract) and the conclusions they have arrived at. You should also give an account of how this research has influenced your thesis.

Descriptive papers may or may not contain reviews, but analytical papers will contain reviews. A literature review must contain at least 5 - 7 published researches in your field of interest.

4.System Designing

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

Diagrams in the UML

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

1. Activity Diagrams – We use Activity Diagrams to illustrate the flow of control in a system. We can also use an activity diagram to refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on the condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram.

2. Use Case Diagrams – Use Case Diagrams are used to depict the functionality of a system or a part of a system. They are widely used to illustrate the functional requirements of the system and its interaction with external agents(actors). A use case is basically a diagram representing different scenarios where the system can be used. A use case diagram gives us a high level view of what the system or a part of the system does without going into implementation details.

3. 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.

4. Class Diagram – The most widely used UML diagram is the class diagram. It is the building block of all object oriented software systems. We use class diagrams to depict the static structure of a system by showing the system's classes, their methods and attributes. Class diagrams also help us identify relationships between different classes or objects.

5. Implementation

The software implementation stage involves the transformation of the software technical data package (TDP) into one or more fabricated, integrated, and tested software configuration items that are ready for software acceptance testing. The primary activities of software implementation include the:

- Fabrication of software units to satisfy structural unit specifications.
- Assembly, integration, and testing of software components into a software configuration item.
- Prototyping challenging software components to resolve implementation risks or establish a fabrication proof of concept.

- Dry-run acceptance testing procedures to ensure that the procedures are properly delineated and that the software product (software configuration items (CIs and computing environment) is ready for acceptance testing.

6. Testing

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises Validation and Verification.

Software Validation

Validation is the process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

- Validation ensures the product under development is as per the user requirements.
- Validation answers the question – "Are we developing the product which attempts all that user needs from this software ?".
- Validation emphasizes on user requirements.

Software Verification

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

- Verification ensures the product being developed is according to design specifications.
- Verification answers the question– "Are we developing this product by firmly following all design specifications ?"
- Verifications concentrate on the design and system specifications.

7.Conclusion

SRS helps the customers to define their needs with accuracy, while it helps the development team understand what the customers need in terms of development. Investing time in writing the SRS document will lead to the successful development of the software the customer needs.

SOFTWARE REQUIREMENTS

Functional Requirements:

- These are statements of services the system should provide
 - =>how the system should react to particular inputs and
 - =>how the system should behave in particular situations
- In some cases, the functional requirements may also explicitly state
 - => What the system should not do
- The functional requirements definition of a system should be both
 - => Complete [i.e. It means that all services required by the user should be defined]
 - => Consistent [i.e. it means that requirements should not has contradictory definitions]

Non- Functional Requirements:

- These are constraints on the services (Or) functions offered by the system
- They include
 - => Timing Constraints
 - => Constraint on development process
 - => Standards and so on...

- Some non-functional requirements may be process rather than product requirements
- Customer imposes these process requirements for two reasons:

=> System Quality

=> System Maintainability

Non-Functional Requirements Types:

Product Requirements Process Requirements External Requirements

(i) Product Requirements:

These requirements results from the need for the delivered product, to behave in a particular way

Example:

- Requirements on how fast the system must execute and how much memory it

requires

- Reliability Requirements [i.e, acceptable failure rate]
- Portability Requirements

(ii) Organizational Requirements:

- These requirements are consequence of organizational policies and procedures

Example:

Implementation requirements such as programming language (Or) design method

used

- Delivery Requirements which specify when the product and its documentation to be

Delivered

(iii) External Requirements:

- This requirements arise from factors external to the system and its development

process

Example:

- Interoperability Requirements which specify how the system interacts with systems in other organizations
- Legislative Requirements, which ensure that the system operates within the law

An Overview of UML

Unified Modeling Language (UML) is a general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language, it is rather a visual language. We use UML diagrams to portray the behavior and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

A Conceptual Model of UML

- A conceptual model can be defined as a model which is made of concepts and their relationships.
- A conceptual model is the first step before drawing a UML diagram. It helps to understand the entities in the real world and how they interact with each other.

As UML describes the real-time systems, it is very important to make a conceptual model and then proceed gradually. The conceptual model of UML can be mastered by learning the following three major elements

–

- UML building blocks
- Rules to connect the building blocks
- Common mechanisms of UML

Object Oriented Concepts Used in UML –

1. Class – A class defines the blueprint i.e. structure and functions of an object.

2. Objects – Objects help us to decompose large systems and help us to modularize our system. Modularity helps to divide our system into understandable components so that we can build our system piece by piece. An object is the fundamental unit (building block) of a system which is used to depict an entity.

3. Inheritance – Inheritance is a mechanism by which child classes inherit the properties of their parent classes.

4. Abstraction – Mechanism by which implementation details are hidden from the user.

5. Encapsulation – Binding data together and protecting it from the outer world is referred to as encapsulation.

6. Polymorphism – Mechanism by which functions or entities are able to exist in different forms.

Diagrams in the UML

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

There are two broad categories of diagrams and they are again divided into subcategories –

1.Structural Diagrams – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

2.Behavior Diagrams – Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

Structural Diagrams

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are –

- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

1.Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagrams consist of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature.

Active class is used in a class diagram to represent the concurrency of the system.

Class diagrams represent the object orientation of a system. Hence, it is generally used for development purposes. This is the most widely used diagram at the time of system construction.

2.Object Diagram

Object diagrams can be described as an instance of class diagrams. Thus, these diagrams are more close to real-life scenarios where we implement a system. Object diagrams are a set of objects and their relationship is just like class diagrams. They also represent the static view of the system. The usage of object diagrams is similar to class diagrams but they are used to build a prototype of a system from a practical perspective.

3.Component Diagram

Component diagrams represent a set of components and their relationships. These components consist of classes, interfaces, or collaborations. Component diagrams represent the implementation view of a system.

During the design phase, software artifacts (classes, interfaces, etc.) of a system are arranged in different groups depending upon their relationship. Now, these groups are known as components. Finally, it can be said component diagrams are used to visualize the implementation.

4.Deployment Diagram

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed. Deployment diagrams are used for visualizing the deployment view of a system. This is generally used by the deployment team.

Behavioral Diagrams

Any system can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered. Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams –

- Use case diagram
- Sequence diagram
- Collaboration diagram
- Statechart diagram

- Activity diagram

1.Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, a use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

2.Sequence Diagram

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

Interaction among the components of a system is very important from implementation and execution perspective. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

3.Collaboration Diagram

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links.

The purpose of the collaboration diagram is similar to a sequence diagram. However, the specific purpose of collaboration diagrams is to visualize the organization of objects and their interaction.

4.Statechart Diagram

Any real-time system is expected to be reacted by some kind of internal/external events. These events are responsible for state change of the system.

Statechart diagram is used to represent the event driven state change of a system. It basically describes the state change of a class, interface, etc. State chart diagram is used to visualize the reaction of a system by internal/external factors.

5. Activity Diagram

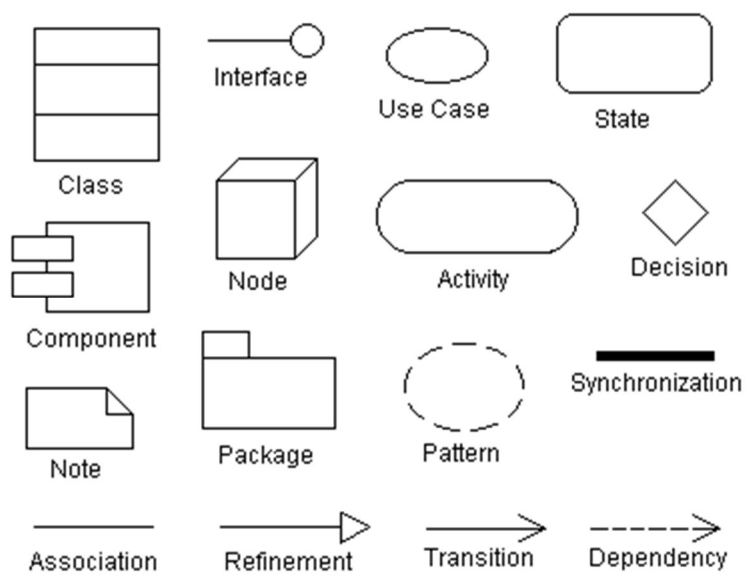
Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

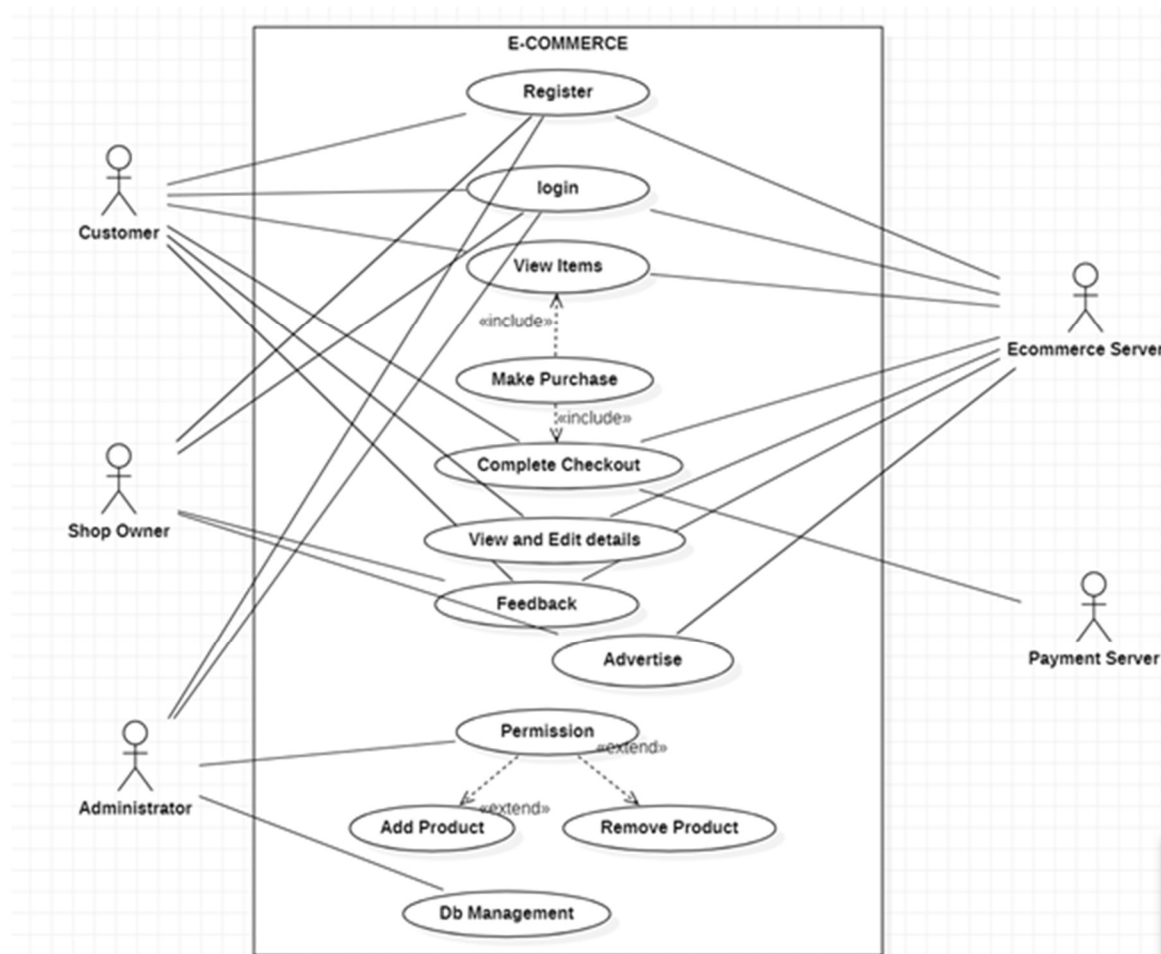
Diagram Elements

Some of the graphical constructs from which diagrams are made are:

- Icon: graphical symbol of fixed size and shape (doesn't hold contents)
- Two-dimensional symbols: have variable size and can expand to hold contents, may be divided into compartments
- Paths: sequences of line segments with attached endpoints. The endpoints are always symbols (no dangling paths). May also have icons at the end to qualify the meaning of the path symbol.
- Strings: text
- Name: A string that uniquely identifies some model element within some scope
- Label: A string attached to a graphic symbol
- Keyword: Text enclosed within "«" and "»" to convey some concept. There are many keywords so we don't need zillions of specialized graphical symbols.
- Expression: A linguistic formula that yields a value
- Some model elements:



Use case diagram



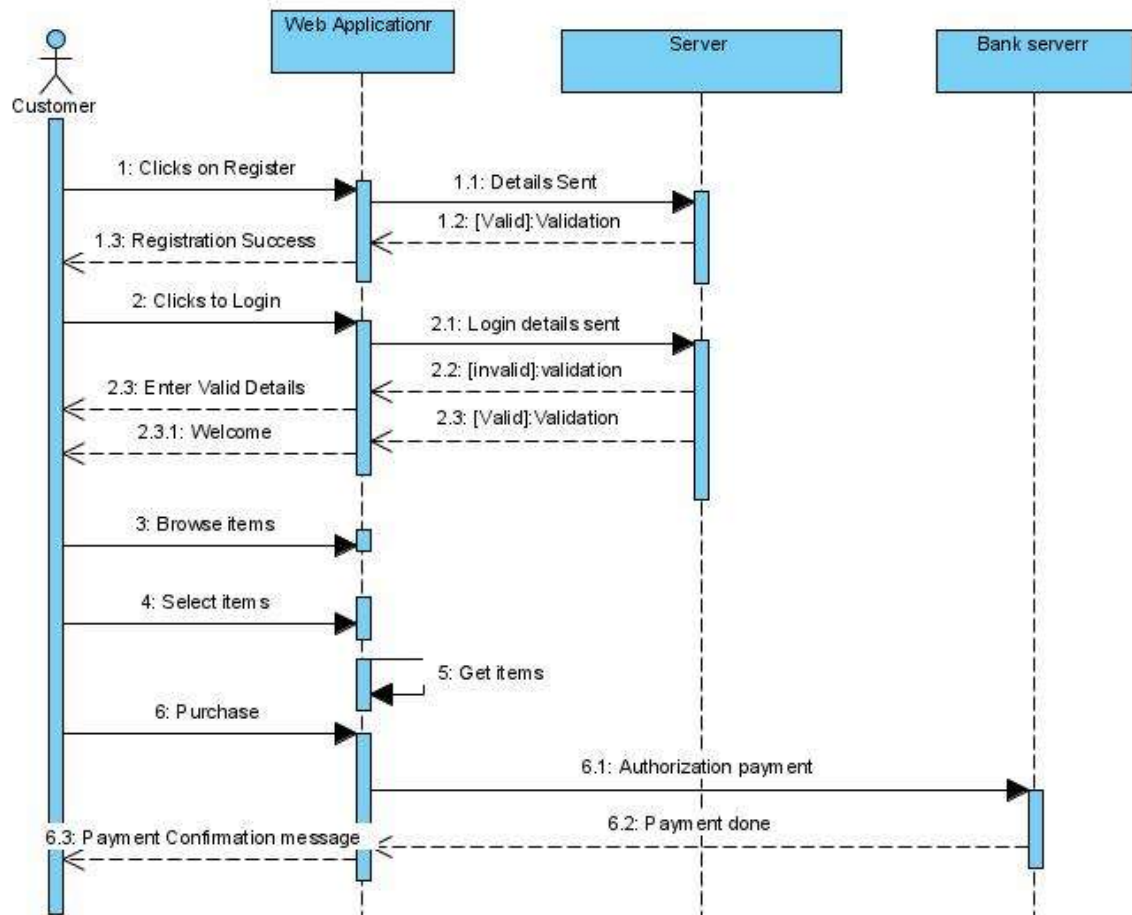
Use case template

Use Case ID:	040816540236		
Use Case Name:	E-Commerce		
End Objective:	Automate and facilitate the whole process of shopping		
Created by:	1.Bachu Bala Raju 2.Viswatej Bhamidipati 3.Gadila Sai Nikshay Reddy 4.Swarna Madhu Babu 5.Bangar Hari Prasad 6. Mohammed Munawar	On (date):	September 24,2020
User/Actor:	Customer,Seller and Administrator		
Trigger:	Customer buying product by logging in the site		
Basic/Normal Flows			
User Actions		System Actions	

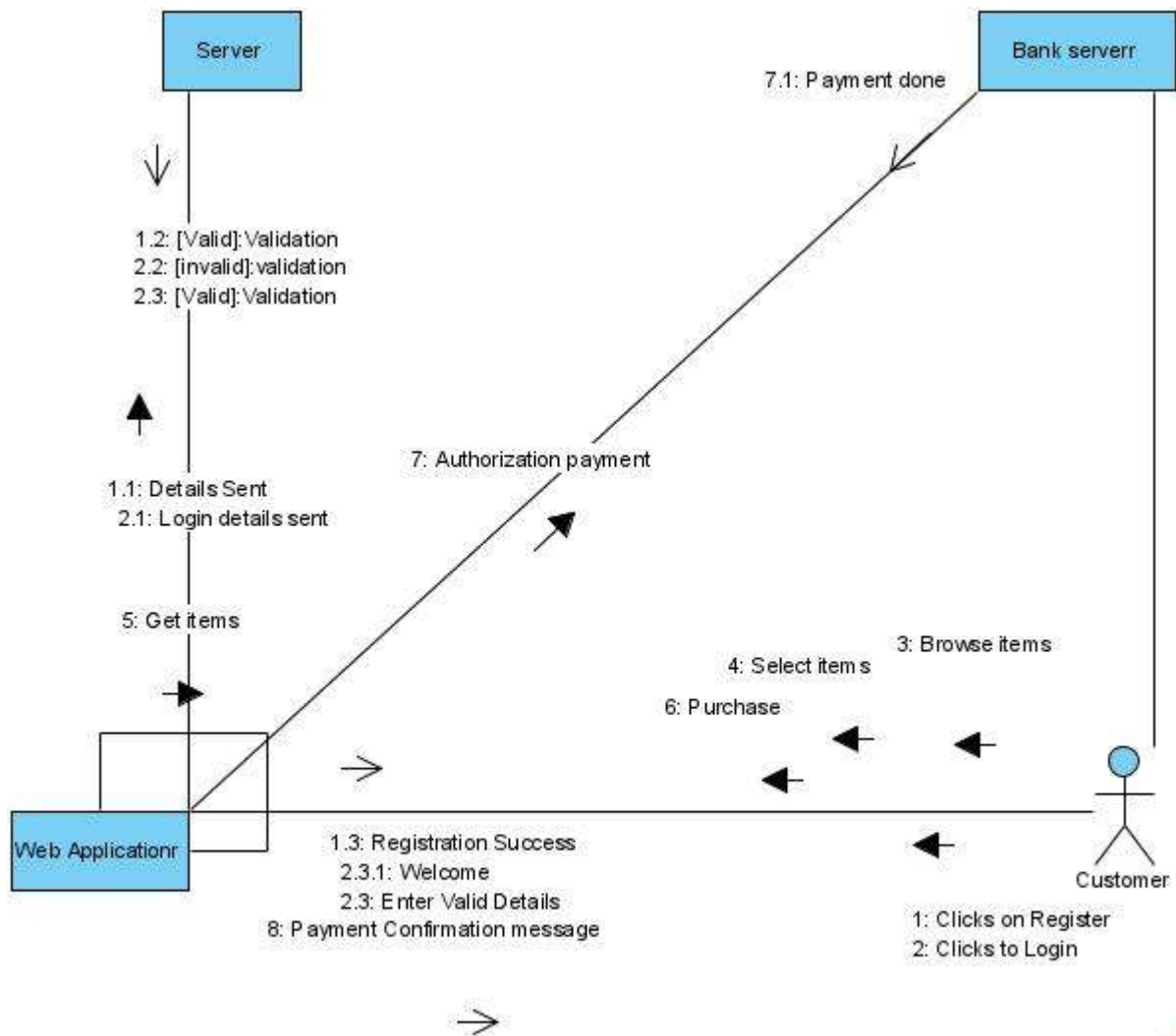
The user login to the site by entering details.	Login page requests the user to provide a proper username and password.
User views suggested products with search bars.	The home page suggests to the user some products the user might want to buy or the user can search the required product in the search bar.
Users make purchases of the product with appropriate purchase options.	Site will provide users with COD or other online payment methods.
User wants to view and edit his/her own details	Site will provide the user to edit the changes
User completes the purchase and checkout.	Site processes the purchase and gives track of the product.

Exception Flows	
User Actions	System Actions
The user tries to login but doesn't have an account on the website.	The page requests the user to register an account in the registration page before committing the login.
The user tries to login by entering details.	The details entered are incorrect .So “Please check the username or password entered” message is displayed and the user needs to enter correct details.
User tries to buy a product	User wants to buy the product, but that product is out of stock then it displays “Out of stock” for that respective product.

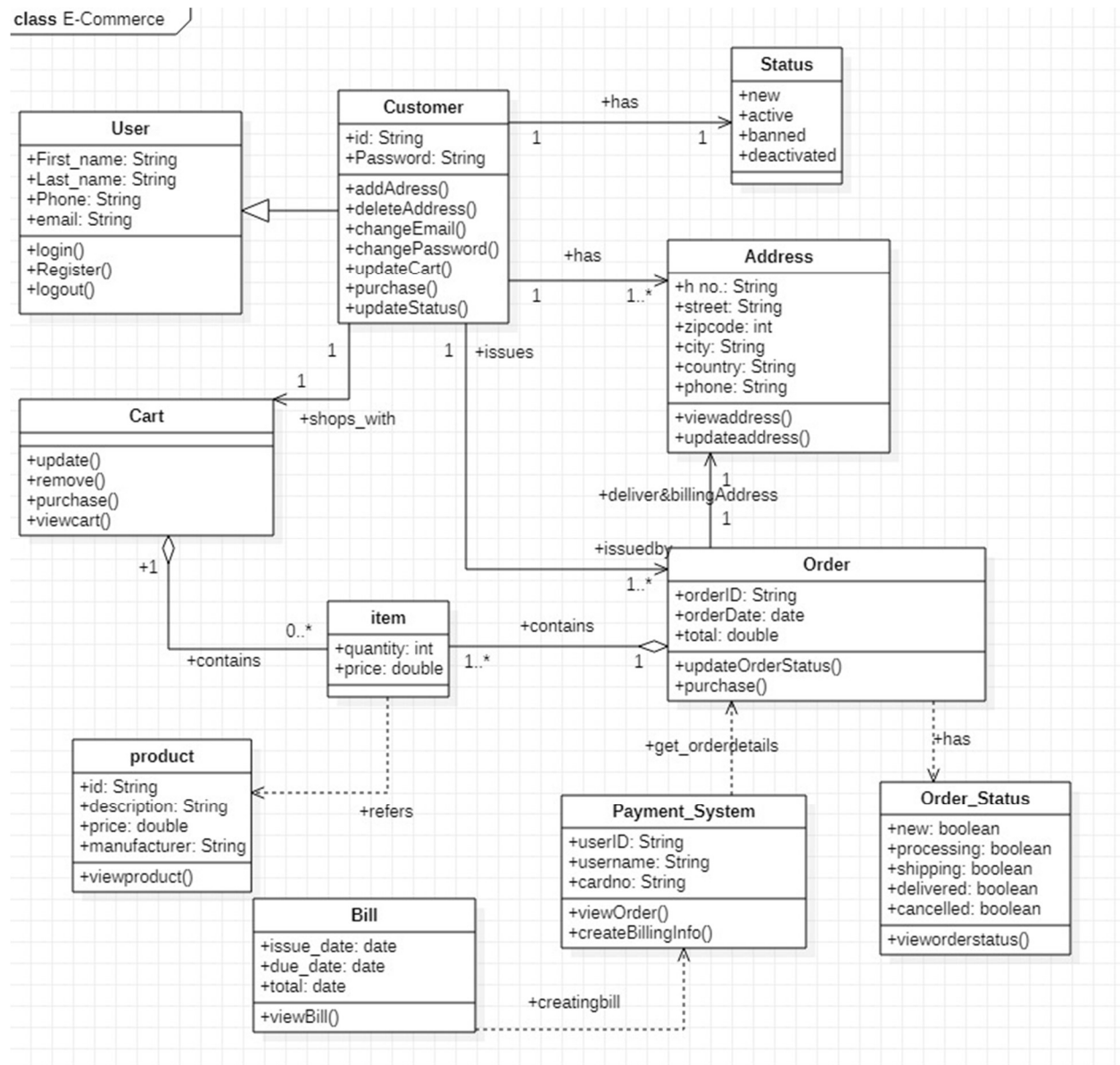
Sequence Diagram



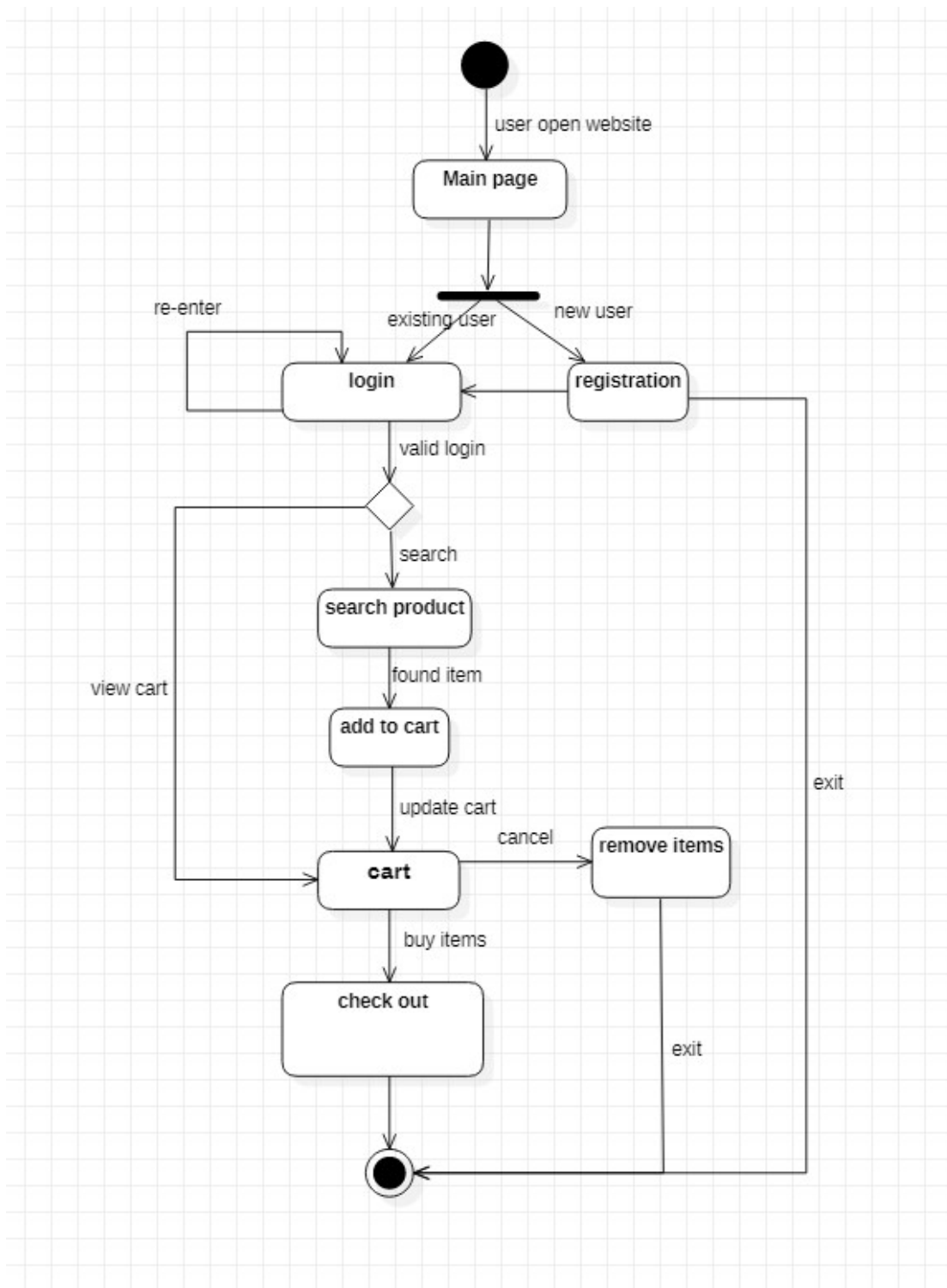
Collaboration Diagram



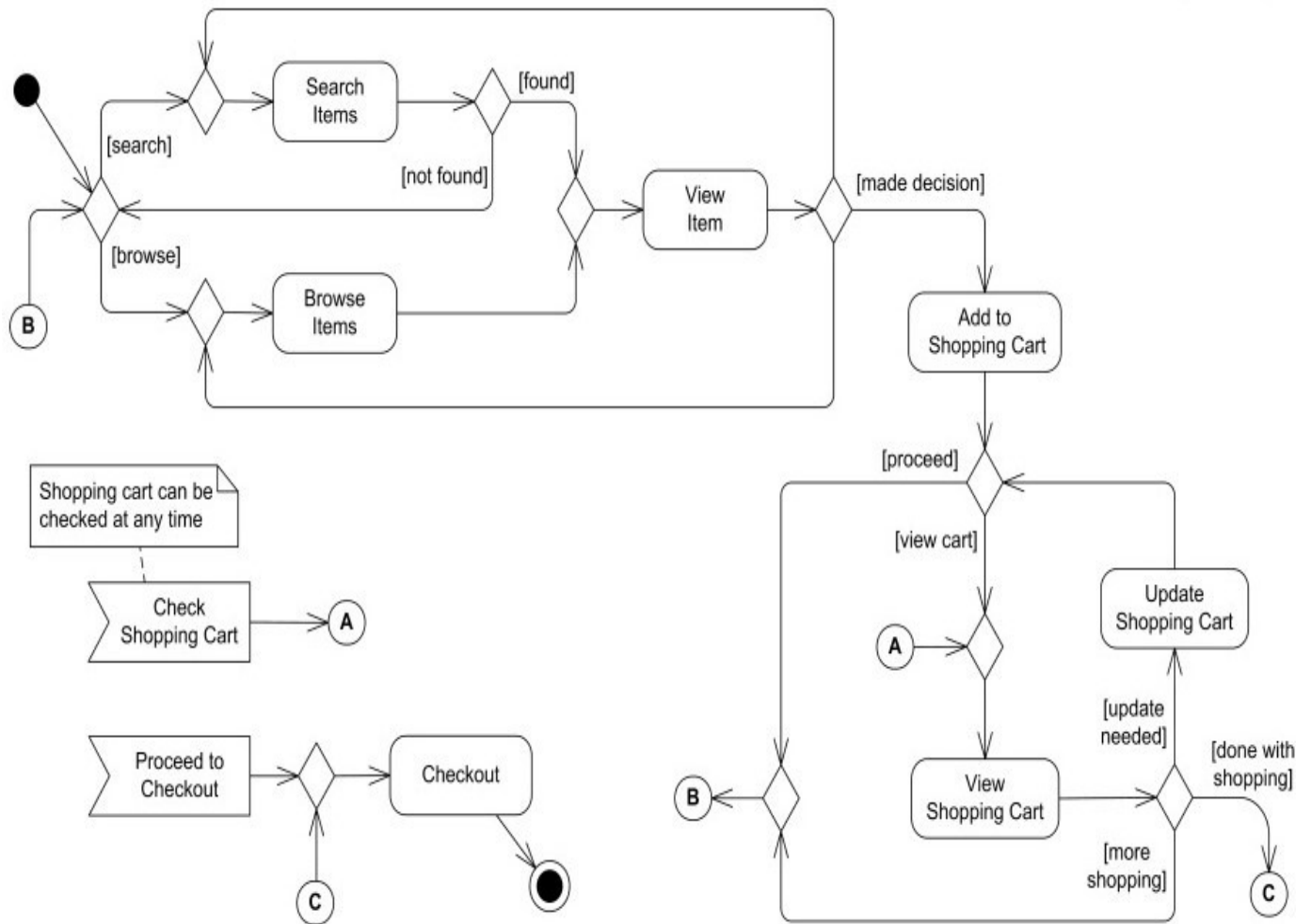
Class Diagram



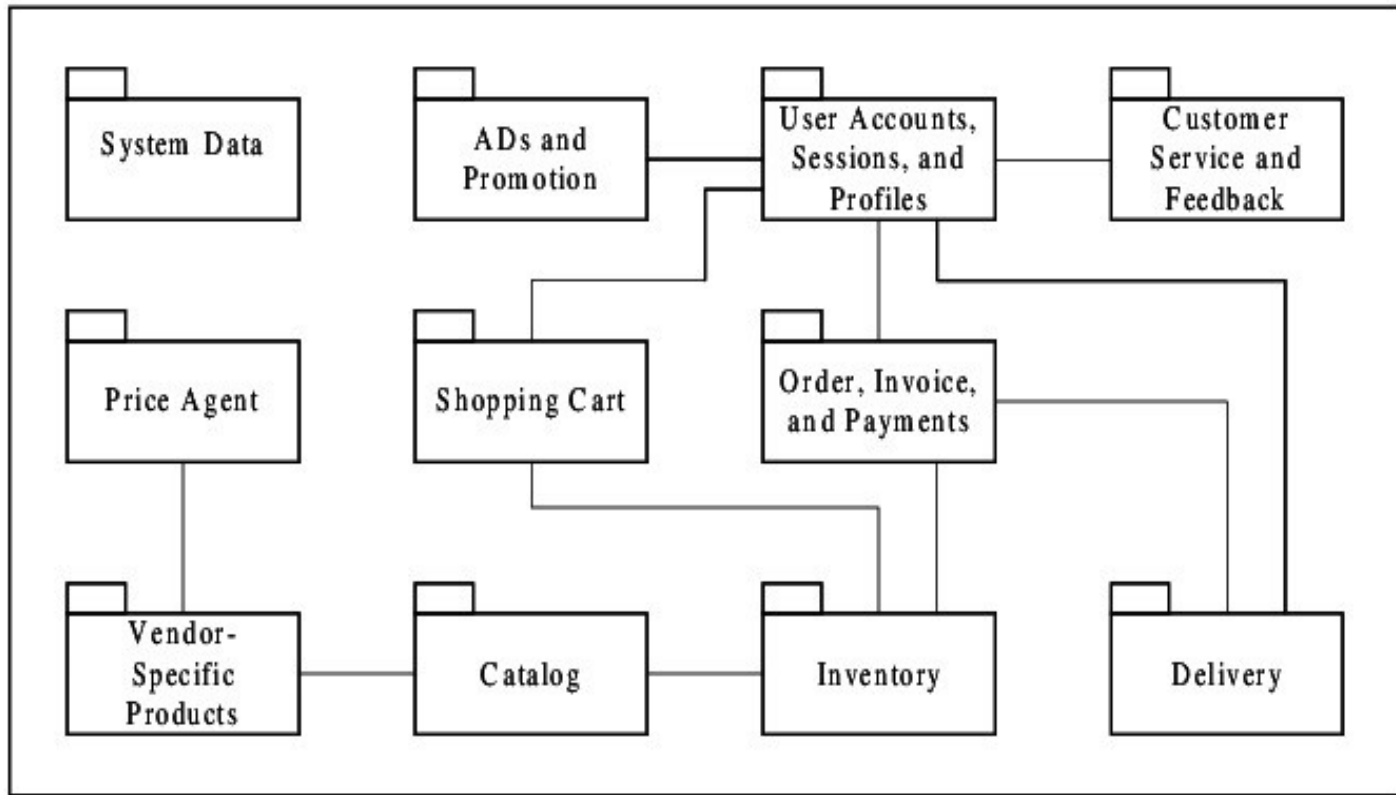
State chart Diagram



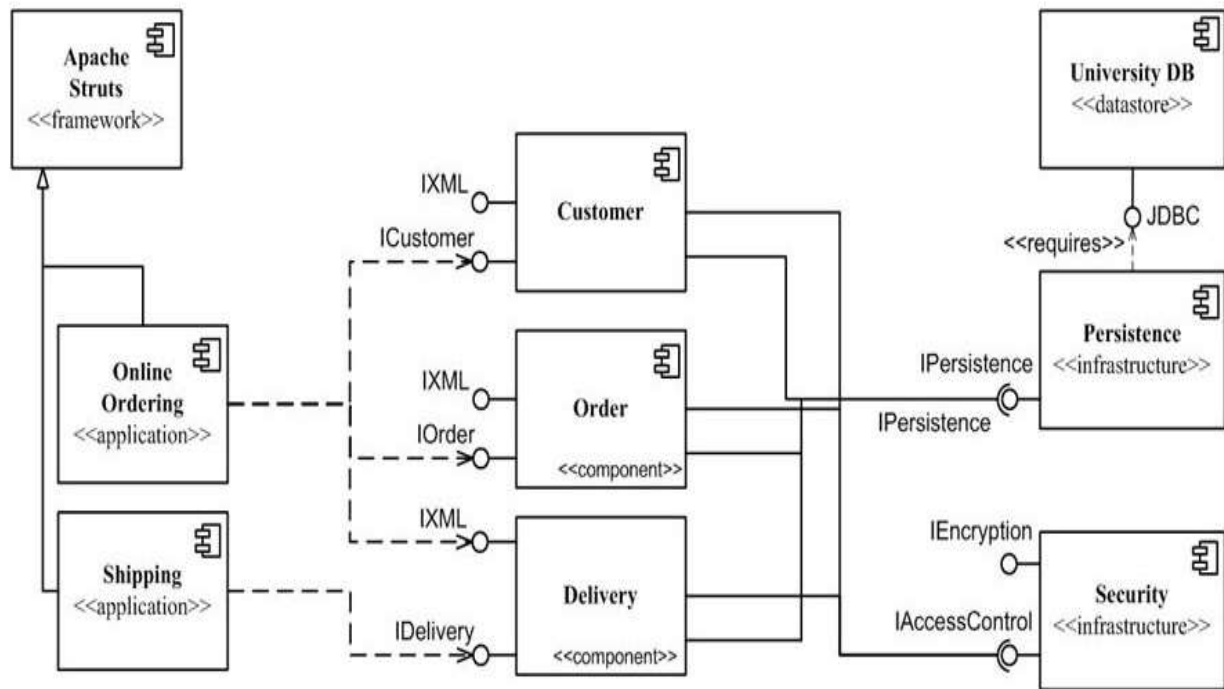
Activity Diagram



Package Diagram



Component Diagram



Deployment Diagram

