



# Dhirubhai Ambani Institute of Information and Communication Technology

## IT – 632 Software Engineering

*Prof. Saurabh Tiwari.*

ThriftKart

### By: -

202112068	BHATT RUTVI SAMIR
202112023	EKANSH GUPTA
202112051	JADHAV SHAILESH SUNILBHAI
202012033	KHUSH SHAH
202112062	SHAH NAJU MANISH
202112012	KUNAL SINGH KAPOOR
202112011	ANANNYASHREE SHARMA
202112029	SAKSHAM JAIN
202112044	KHATRI MOHAMADSULTAN GULAMABBAS
202112095	SANJEEV SINGH BHADOURIA
202112116	SHAH GATHA DILIP

# INDEX

<b>Sr No.</b>	<b>Content</b>	<b>Page</b>
<b>1</b>	<i>Introduction</i>	<i>3</i>
<b>2</b>	<i>Scope of the Project</i>	<i>4</i>
<b>3</b>	<i>Requirements</i>	<i>5</i>
<b>4</b>	<i>Process Model</i>	<i>9</i>
<b>5</b>	<i>Defining the Users and Stakeholders</i>	<i>11</i>
<b>6</b>	<i>Use-Case Diagram</i>	<i>12</i>
<b>7</b>	<i>Tech Stack</i>	<i>21</i>

# INTRODUCTION

The broad definition of e-commerce transactions refers to the selling and buying of products and services over computer-mediated networks while the end process of payment and delivery is managed offline. Electronic Commerce (e-commerce) is defined as the conduct of commerce in goods and services, with the assistance of telecommunications and telecommunications-based tools such as the Internet. E-commerce is often used in a much broader sense, to mean essentially the same as “electronic business” (EB). E-commerce encompasses many areas, which include electronic catalogues that refer to means whereby sellers can communicate their offerings to potential buyers.

ThriftKart is an E-commerce (virtual) store, where customers can browse the catalog and select products of interest. Customers directly buy used goods without any intermediary service over the internet. It allows users to search for products based on different categories, add in cart or save in wish-list. It provides online as well as offline (Cash on Delivery) payment options for registered users. The main idea is to develop a web-based interface for e-commerce of used goods, the user-friendly interface will make the shopping experience pleasant for the users. This project is an attempt to provide the advantages of online shopping to customers of physical stores for buying used goods.

# SCOPE OF THE PROJECT

This system is designed to provide quality service to the customers and to allow admin a quick and easy means to setup and perform sales and other core business over the internet. It also allows admin to track the e-commerce data on the site, for improving business strategies. The system allows the users to create a free account and update their details as and when required. It also allows users to save products in a wishlist which provides a better experience of searching and buying products at any other time. The current system is confined only to shopping for old products categorically. It allows users to filter products, based on their needs, budget, trend, etc. It can be extended to have an easy to use check out process, with availability of convenient paying options such as UPI, credit, debit, net banking, Cash on delivery, etc. Users can have multiple shipping and billing information saved, which can be updated any time with ease. During the checkout process, users can easily select any of their available shipping options, as well as mode of payment, providing the best buying experience. This system also provides users with records of their previously bought items, tempting them to buy again, thus improving the business in the best way possible. Users can track their orders any time they want and download their order invoice any time, which reduces the risk of losing their billing information, for return/replacement purpose. Feedback is the essence of any system, hence this system comes with contact us features which helps in smooth interaction between a customer and the admin.

# REQUIREMENTS

## FUNCTIONAL REQUIREMENTS

Identification of functional requirements: The Functional Requirements describes the functionality of a system that how a system should react to a particular set of inputs and what should be the corresponding output. The above problem statement gives a brief description of the proposed system. And after analysing, we might easily identify some of the basic functionality of the system:

<b>RN</b>	<b>DESCRIPTION</b>
<b>FR1</b>	Ability to add/edit all personal information
<b>FR2</b>	Allow customers to set default shipping and billing addresses.
<b>FR3</b>	Provide "remember me next time" auto-login functionality (future scope).
<b>FR4</b>	Ability to view "pre-shipping" order status.
<b>FR5</b>	Provide progress bar for filling out first time registration (future scope).
<b>FR6</b>	Customer authentication using OTP sent on their emails.
<b>FR7</b>	Customers can view the product in a categorized way.
<b>FR8</b>	Top-selling/ Top Rated products will be shown on the homepage of the customer.
<b>FR9</b>	Ability to search on product name, keywords, description text, item number, partial item number and/or category keyword.
<b>FR10</b>	Customers can give the rating to the purchased products, and can also add some comments about the product.

<b>FR11</b>	Average of all the ratings will be shown with all the comments/feedback.
<b>FR12</b>	Registration Page allows user to create an account (user get access to login functionality).
<b>FR13</b>	Registering users gives email, fname, lname and password.
<b>FR14</b>	Login Page It allows user to login (user get access to the checkout page and payment functionality).
<b>FR15</b>	Payment can be done online (done while purchasing product) and offline (after receiving the product).
<b>FR16</b>	Payment status and transaction id (online purchase) will be maintained.
<b>FR17</b>	Product will be delivered to the default address, given during registration.
<b>FR18</b>	Customers can select a custom address for the particular delivery.
<b>FR19</b>	Shipping status will be shown to the customers which are updated by the admin.
<b>FR20</b>	Add card information like card number, expiry_date etc.
<b>FR21</b>	A user can add multiple card details and can set a single card_info to be used by default.
<b>FR22</b>	Adds details about address and the user's number.
<b>FR23</b>	A user can add multiple contact details and can set a single to contact_details to be used by default.
<b>FR24</b>	Adds a product in the shopping cart for a buyer, given the buyer Id.
<b>FR25</b>	Adds a product to the wishlist of a buyer, given the buyer id.
<b>FR26</b>	Reports of Top selling products, Total purchase, and category-wise product selling can be viewed by the admin.

<b>FR27</b>	Report can be of 1 month, 6 months or a year.
<b>FR28</b>	Admin can also able to view custom date range report.

## NON - FUNCTIONAL REQUIREMENTS

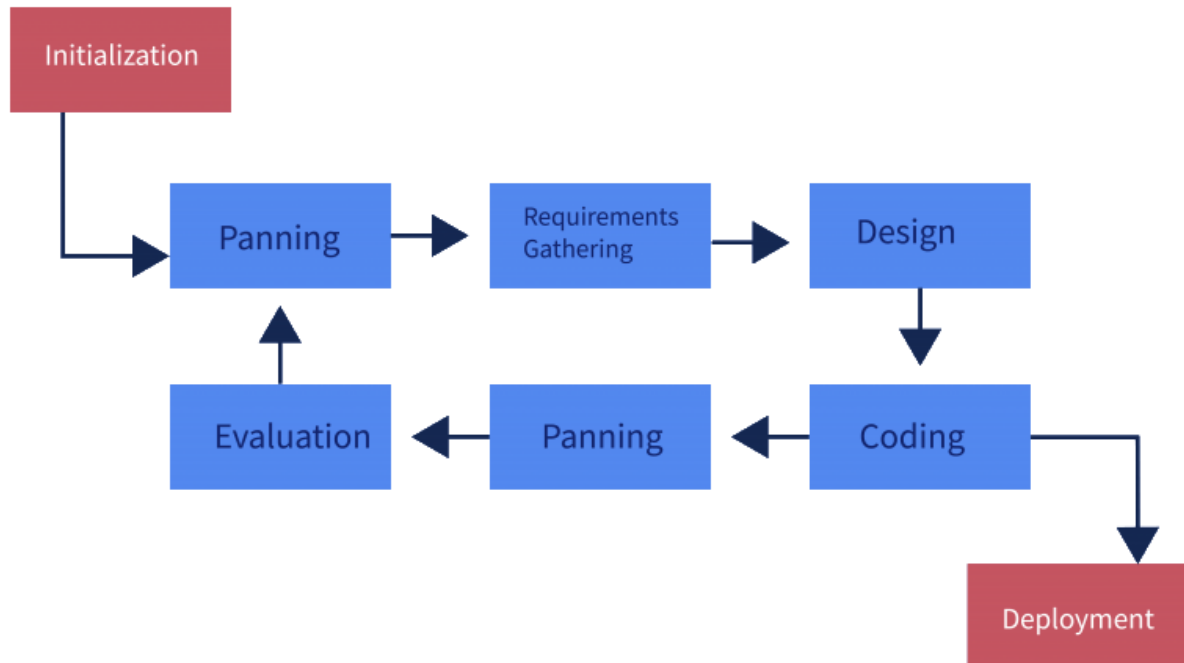
Identification of non-functional requirements: The non-functional requirements are not directly related what functionalities are expected from the system. However, NFRs could typically define how the system should behave under certain situations. It is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. It will define constraints which affect how the system should do it.

NR	Comments	Description
NR1	Security	System should be able to provide authenticity and permissible access to data and operation.
NR2	Integrity	Integrity of data will be provided and operation can be done 24 * 7.
NR3	Performance	System should be able to handle multiple user requests parallely.
NR4	Scalability	Server should be able to handle a large number of requests without performance degradation.
NR5	Portability	System shall be independent of any operating system and will work on any browser.
NR6	Maintainability	Maintenance/Operations costs should be minimal for the website.
NR7	Reliability Requirement	The system should provide a reliable environment to both customers and owners.  All orders should be reached at the admin without any errors.
NR8	Usability	Website should be easy to use, even for a non-technical person.



# PROCESS MODEL

## Iterative Process Model



### 1. Requirement and Planning Stage:

During this phase, scope of the project is decided and technical requirements are gathered and evaluated for feasibility.

### 2. Design Stage:

In this stage, the project flow and the system design is decided with the help of Use - Case Diagram and the Requirements (Functional and Non - Functional Requirements) followed by UI Design.

### 3. Coding Stage:

All of the requirements, planning, and design plans are executed and coded. During the code development, they must implement a unit test at each level which will followed by the testing stage.

### 4. Testing Stage:

In this stage the developer and the tester has to ensure that any bugs created while coding are fixed and no new bugs are created. We can also check in with the project stakeholders to run some tests and inquire about any feedback they may have.

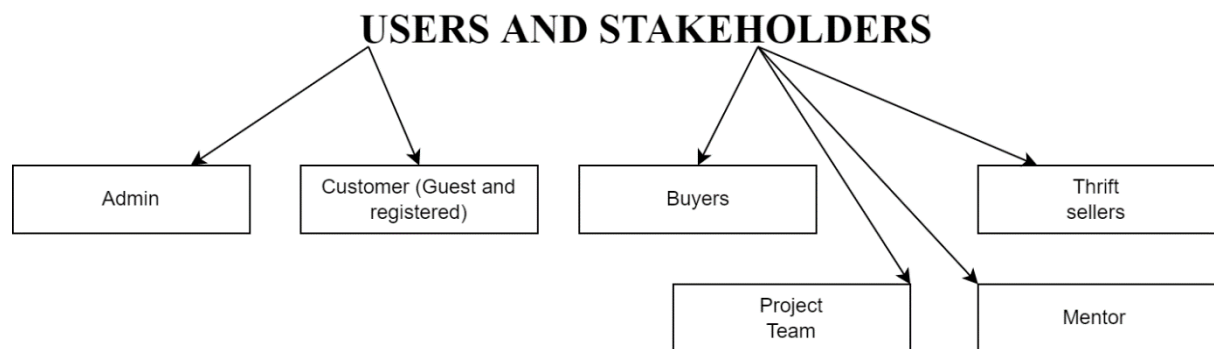
### **5. Evaluation Stage:**

In this stage evaluation for all the previous stages is done. The system is examined by the development team, and other teams responsible for developing the project to see if the outcomes satisfy their expectations.

### **Reason For Choosing Iterative Process Model:**

- ✓ Requirements of the system are clearly defined and understood for the early stages.
- ✓ Some functionalities or requested enhancements may evolve with time.
- ✓ In an iterative model we are building and improving the system step by step. Hence, we can track the defects at the early stages. This avoids the downward flow of the defects.
- ✓ After going through all the stages a new requirement plan is implemented as a part of the next iteration.

# DEFINING THE USERS AND STAKEHOLDERS OF THE PROJECT



# USE CASE DIAGRAM

[Click here to view the Diagram](#)

## Sign Up

<b>Use Case name</b>	<b>Sign Up</b>
<b>Brief Description</b>	A user who wants to purchased something from website, must have to register by filling appropriate details
<b>Actors</b>	Guest User
<b>Pre condition</b>	The user must have fill all the details in form and should have unique email id which should not be already registered
<b>Post condition</b>	If user registered successfully then should be redirected to login, else no change in state
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. System will request the user to fill out the form</li> <li>2. Email will be sent to user &amp; system will verify the email</li> <li>3. Successful registration will be notify</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If email id already registered then it will redirect it to login</li> <li>2. If any details are not filled accordingly to it was asked then account not created notification will be sent and user may restart the process</li> </ol>

## Sign In

<b>Use Case name</b>	<b>Sign In</b>
<b>Brief Description</b>	To access the system and features user have to log in into the system by using valid email id and password
<b>Actors</b>	Registered User , Admin
<b>Pre condition</b>	User must have registered
<b>Post condition</b>	If entered credentials are valid details redirect to relevant homepage or else no change in current state
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. System will ask user to enter credentials</li> <li>2. System will authenticate the credentials</li> <li>3. Redirected to relevant homepage</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If credential are not valid or empty than system will notify the user with the error message</li> <li>2. And user can restart the process</li> </ol>

## Manage Product

<b>Use Case name</b>	<b>Manage Product</b>
<b>Brief Description</b>	Admin can add the product and their description , update product details and can remove the products
<b>Actors</b>	Admin
<b>Pre condition</b>	User must be admin and logged in to the system
<b>Post condition</b>	If use case successfully ended then Product details should be updated in system
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. System will display the added product</li> <li>2. User can select operation (add/update/delete) , according to that will redirected to particular operation page</li> <li>3. User have to fill appropriate details if asked</li> <li>4. If all the details or operations are correct then details will be updated</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If user filled wrong details or done invalid operation</li> <li>2. Then related error message will be shown and data will not be updated</li> </ol>

## Search

<b>Use Case name</b>	<b>Search</b>
<b>Brief Description</b>	User can search the products and according to that result will be shown
<b>Actors</b>	Registered user,Admin
<b>Pre condition</b>	User must logged in to the system
<b>Post condition</b>	If product is available then it should be show to user
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. User will choose to search for product</li> <li>2. User will guided to select required criteria to search for the product</li> <li>3. According to that criteria if product will available then it must be displayed to user</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If user filled wrong details or products are not available then according to that error message will be shown to the user.</li> </ol>

## Manage Account

<b>Use Case name</b>	<b>Manage Account</b>
<b>Brief Description</b>	User can update profile details, add account details and can see order details for their account
<b>Actors</b>	Registered user
<b>Pre condition</b>	User must be log in to the system
<b>Post condition</b>	If use case successfully ended then profile details should be updated in system or details should be displayed according to request
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. System will display the profile details in profile tab</li> <li>2. User can select operation (update profile or add card details ,view order details) , according to that they will redirected to particular operation page</li> <li>3. User have to fill appropriate details if asked</li> <li>4. If all the details or operations are correct then details will be updated or displayed</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If user filled wrong details or done invalid operation</li> <li>2. Then related error message will be shown and data will not be updated or displayed</li> </ol>



## Manage cart

<b>Use Case name</b>	<b>Manage cart</b>
<b>Brief Description</b>	User add or remove products from the cart
<b>Actors</b>	Registered user
<b>Pre condition</b>	User must be log in to the system and product should be available
<b>Post condition</b>	If use case successfully ended then cart will be updated
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. User will select product or select add cart option</li> <li>2. Quantity will be asked and added in to the cart</li> <li>3. If user select remove from cart option then product will removed from the cart</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If product is not available then error message should be display and should not be added into the cart</li> </ol>

## Checkout

<b>Use Case name</b>	<b>Checkout</b>
<b>Brief Description</b>	User can start the shipping process for the product added into the cart
<b>Actors</b>	Registered user
<b>Pre condition</b>	User must be log in to the system and product should be already added into the cart
<b>Post condition</b>	If use case successfully ended then order will be placed
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. User will process the checkout</li> <li>2. Shipping address will be asked or it will select default</li> <li>3. Payment option should be select</li> <li>4. according to that if payment was done then order will be placed</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If payment is not done or canceled then user can select other option or redirect to the beginning</li> <li>2. If product is not available then message should be display and should not be ordered</li> </ol>

## Report

<b>Use Case name</b>	<b>Report</b>
<b>Brief Description</b>	User or admin can view the report
<b>Actors</b>	Registered user, Admin
<b>Pre condition</b>	User must logged in to the system, For Registered user some order should be done in that period Or for Admin some purchase should be done in that period
<b>Post condition</b>	If report is available then it should be show to users
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Registered User will choose to view order details / Admin will guided to select required criteria to generate report</li> <li>2. According to that criteria if report are available then it must be displayed to user</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If user selected wrong date range then redirected to select again</li> <li>2. If orders or purchase transaction not done in given range of date then appropriate message will be shown</li> </ol>

## Feedback

<b>Use Case name</b>	<b>Feedback</b>
<b>Brief Description</b>	User can give feedback for product and admin can view it
<b>Actors</b>	Registered user,Admin
<b>Pre condition</b>	User must logged in to the system, User must have purchased the product before giving feedback
<b>Post condition</b>	If purchase authentication is done feedback should be updated in to the system
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. After user have purchased the product , rating option will be given</li> <li>2. User can give the rating or can update if already given</li> <li>3. Admin can view that rating</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. If user have not purchased product then they will be notified and ask for the purchase the product first</li> </ol>

# TECH STACK

## SOFTWARE INTERFACE

The following will be the technology stack used in this application:

Tools	Technologies
UI/UX Design	PhotoShop
Backend	NodeJS , Express,Js, JavaScript
Frontend	HTML, CSS, JavaScript, ReactJs
Database	MongoDB
Version Control System	GitHub
IDE	Visual Studio Code
Category	Web Application