InaniHub

A Project Report

Developed By

Pethani Madhuri 215490694010

Thummar Komal 215490694007

Guide By

Prof. Pooja Bhadauriya

In a partial fulfillment for the award of the degree of

Master of Computer Applications In

R.B. Institute of Management Studies

Submited By

Gujarat Technological University, Ahmedabad

June 2023

1. Introduction

1.1 Existing System

- E-platform means that the all the people can sell or buy all the thing online at the one platform.
- There are many e-platforms available in market.
- It is as follows –
    - Craigslist
    - Facebook Marketplace
    - Geebo
    - eBay
    - Gumtree
    - Oodle
- Our web-site is make for the same facility for all the users at world wide web.

1.2 Need for the new System

- In the old system not an every thing is cover in site. But in our site we have cover up all things for all the type of persons.
- We provide the cash on delivery to all the users.
- We provide 24/7 service.
- So full fill the all the requirement we make a new design for the users.

1.3 Objective of new System

- Provide the process project the capability to book different kind of passenger.
- Provide a ticketless flight travel using the proposed project ticketing system feature.
- Build the proposed project with the capability to include special handling procedure to a booking reference.
  Build the proposed project with the capability to make rebooking

1.4 Problem Definition

- Our project basically created for manage sell and purchase the products at on the one platform.
- We are provided REACT.JS as front end and NODE.JS as back end. The project processed through a sequence of well designed forms provided with validation to ensure.
- Consistency reliability and most importantly correctness of information in to the database.
- We provide the facility for sell and purchase at one platform.

1.5 Core Components

- UI/UX Web Application Components – This includes activity logs, dashboards, notifications, settings, statistics, etc.
- These components have nothing to do with the operation of a web application architecture. Instead, they are part of the interface layout plan of a web app
- Structural Components – The two major structural components of a web app are client and server sides.
- Client Component - The client component is developed in CSS, HTML, and JS. As it exists within the user's web browser, there is no need for operating system or device-related adjustments. The client component is a representation of a web application's functionality that the end-user interacts with.
- Server Component - The server component can be build using one or a combination of several programming languages and frameworks, including Java, .Net, NodeJS, PHP, Python, and Ruby on Rails. The server component has at least two parts; app logic and database. The former is the main control centre of the web application whilethe latter is where all the persistent data is stored.

1.6 Project Profile

Sr. No	Title	Detail
1.	Group No	25
2.	Project Title	Inani Hub
3.	Front-End Tool	REACT.JS
4.	Back-End Tool	Mongo DB
5.	Project Type	Web Application
6.	Project Duration	4 months
7.	Project Team Size	Two(2)
8.	Submitted By	Pethani Mahduri (215490694010) Thummar Komal (215490694007)
9.	Stream	MCA(Sem 4)
10.	Guide By	Prof. Pooja Bhadauriya
11.	Submitted To	R B INSTITUE OF MANAGEMENT AND STUDIES

1.7 Assumptions and Constraints

- Enani Hub is mainly used for buy and sell the Goods for buyers and sellers.
- Information every time they book,Saving a lot of time of all the users.
- it is confirmed that flight booking system has removed a lot of burden from the authorities and made the life of the customers very easy.

1.8 Advantages

- 24/7 all the Services available for all the users.
- Generated automated emails and SMS for the status.
- Real-time update.
- Easy cancellation process for Goods buy or sell.

1.9 Limitation

- Users can only pay after retirving or selling their products.
- We don’t provide online payment method

2. Requirement Analysis

2.1 Tools & Technology

2.1.1 ReactJS

- ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components.
- A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components arethe heart of all React applications
- It is an open-source, component-based front end library responsible only for the view layer of the application

2.1.2 Importance of ReactJS

- Fast
- Moduler
- Sacalable
- Flexible
- Popular
- Easy to learn
- Reusable UI Components

2.1.3 New Features of ReactJS

- Better language support
- Programmable control
- Event driven programming
- User authentication which account and role
- Increased performance compiled code

2.1.4 Benefit of of ReactJS

- Easy to learn
- Reduced Coding
- Reusable Components
- Performance Enhancement
- Support for Handy Tools
- Clean Abstraction

2.2.1 Mongodb

- MongoDB, the most popular NoSQL database, is an open-source document-oriented database.
- The term 'NoSQL' means 'non-relational'.
- It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data.
- This format of storage is called BSON (similar to JSON format)

Features

- It is a non-relational and document-oriented database
- It is suitable for hierarchical data storage
- It has dynamic schema
- In terms performance , it Is much faster

Advantages

- is easy to install, use and schema-less database.
- Due to it is the ability of a schema-less database, the code which we create defines the schema
- Data is stored in Binary JSON format, which is key-value pair, no joins complexity is needed

2.3.1 JavaScript

- Java script is a scripting language often used to client side web development.
- Java script was influenced by many language and was designed to have similar work to java but be easier for non- programmers to work with.

Features

- In the community of web developers and surfers java script is highly popular as client side scripting language for the web browser.

Support of object

- Java script is an object oriented language however the way java script handles object inheritance is bit different from conventional objected programming language like java due to this java script support most of the object oriented concepts while being simple to learn and use.

2.4.1 CSS

- Css cascading style sheets.
- Css use to control the style and layout of multiple web pages all once.
- Styles are normally stored in style sheets. External style sheets are stored in css file.
- Styles ware added to html to solve a problem.
- Multiple style definition will cascaded in to one

2.5.1 User Characteristic

- User should be comfortable with English language.
- Basic knowledge about computer.
- Use able to put required in formation secure user login account.
- These users are usually responsible for insuring that
- A design is feasible and software.
- More often than not software is design for a client

2.6.1 Education level

- User should be conformable with English language

2.2 Hardware & Software

2.1 Hardware

Hard disk	500GB
Process	2.33GHZ
System Type	64 bit operating system
RAM	4.00GB

2.3 Software

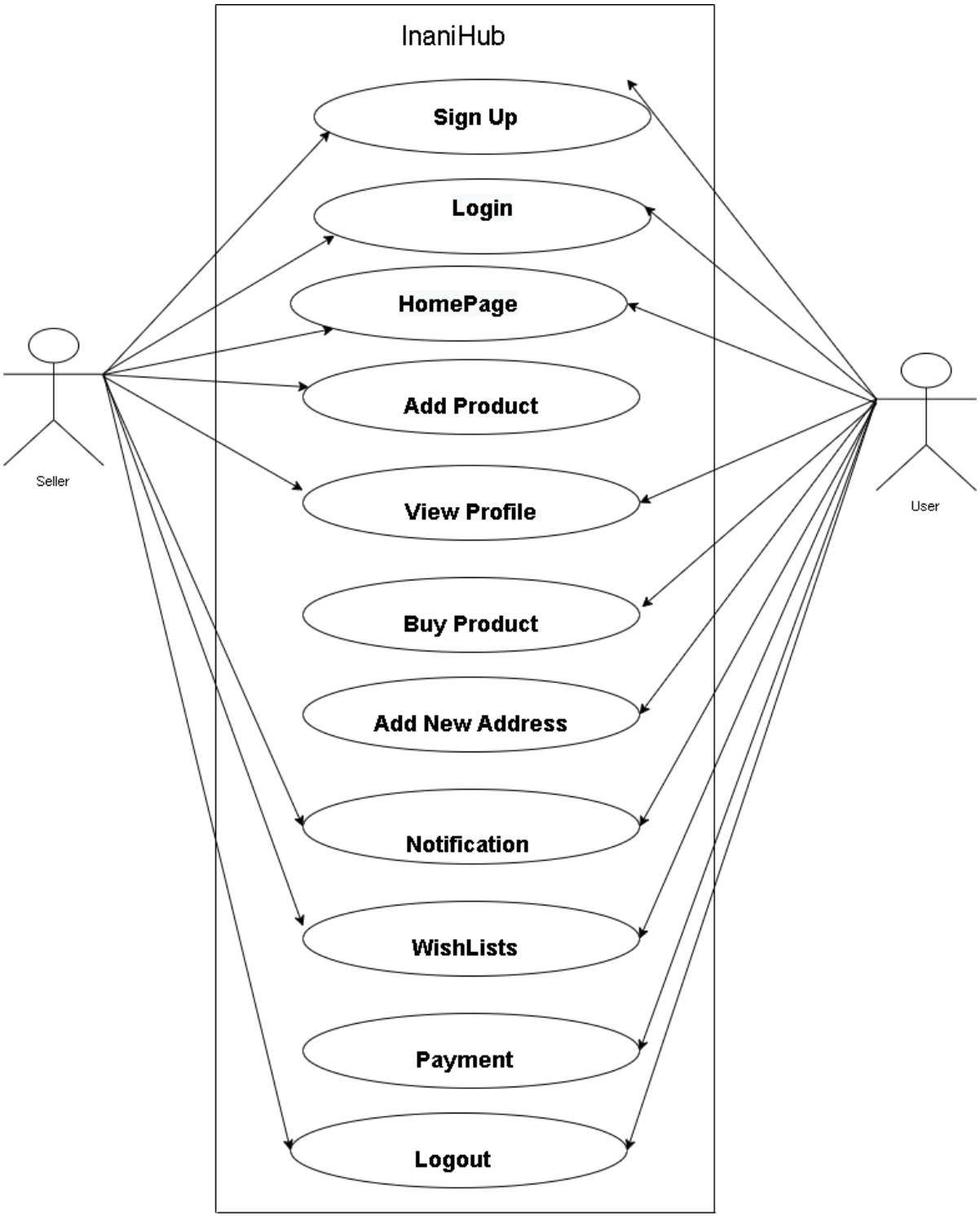
Operating System	Microsoft windows-10
Development Tools &	React.js (visual studio -code)
Technology Back End	Mongodb

2.3 Targeted Users

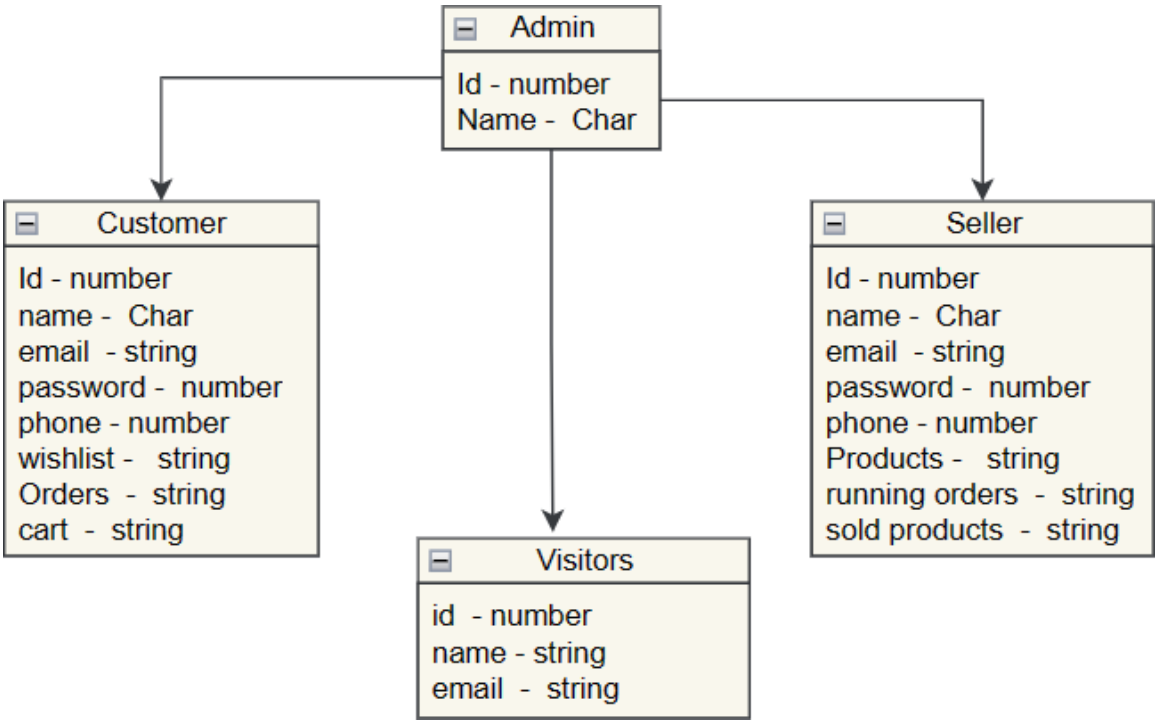
- These users are the users for whom we create our site for those to do their work more fast , reliable and efficiently.

3. System Design

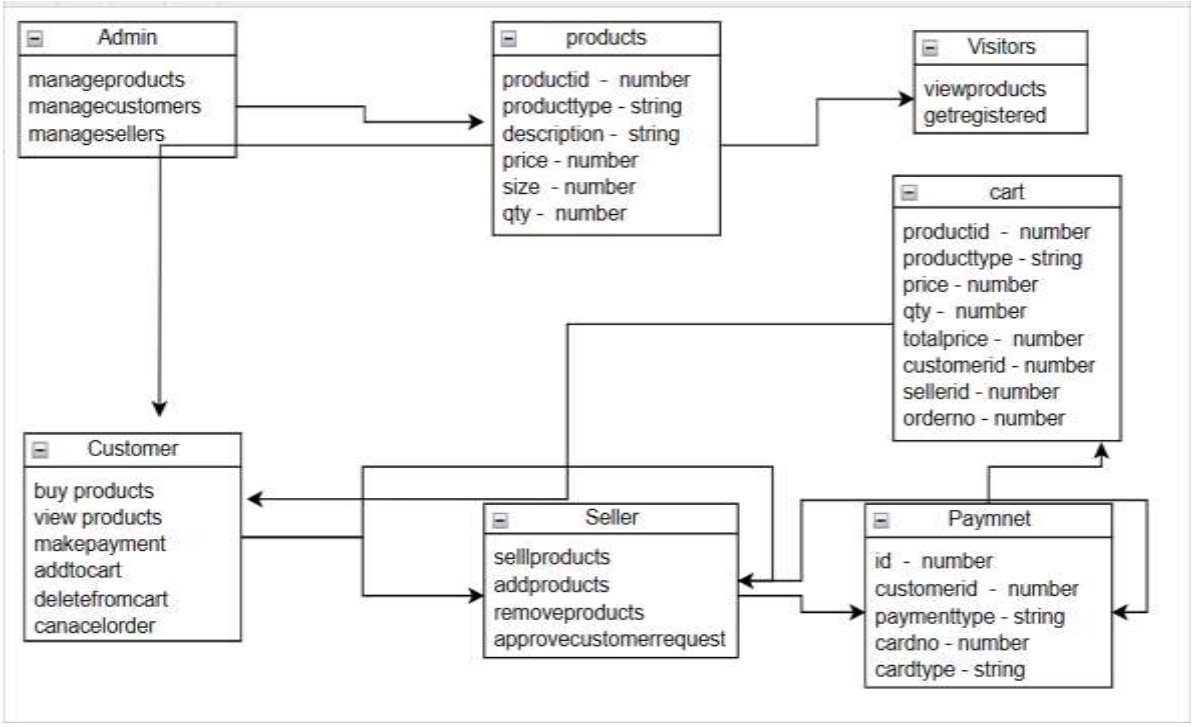
3.1 Use Case Diagram



3.1 Class Diagram

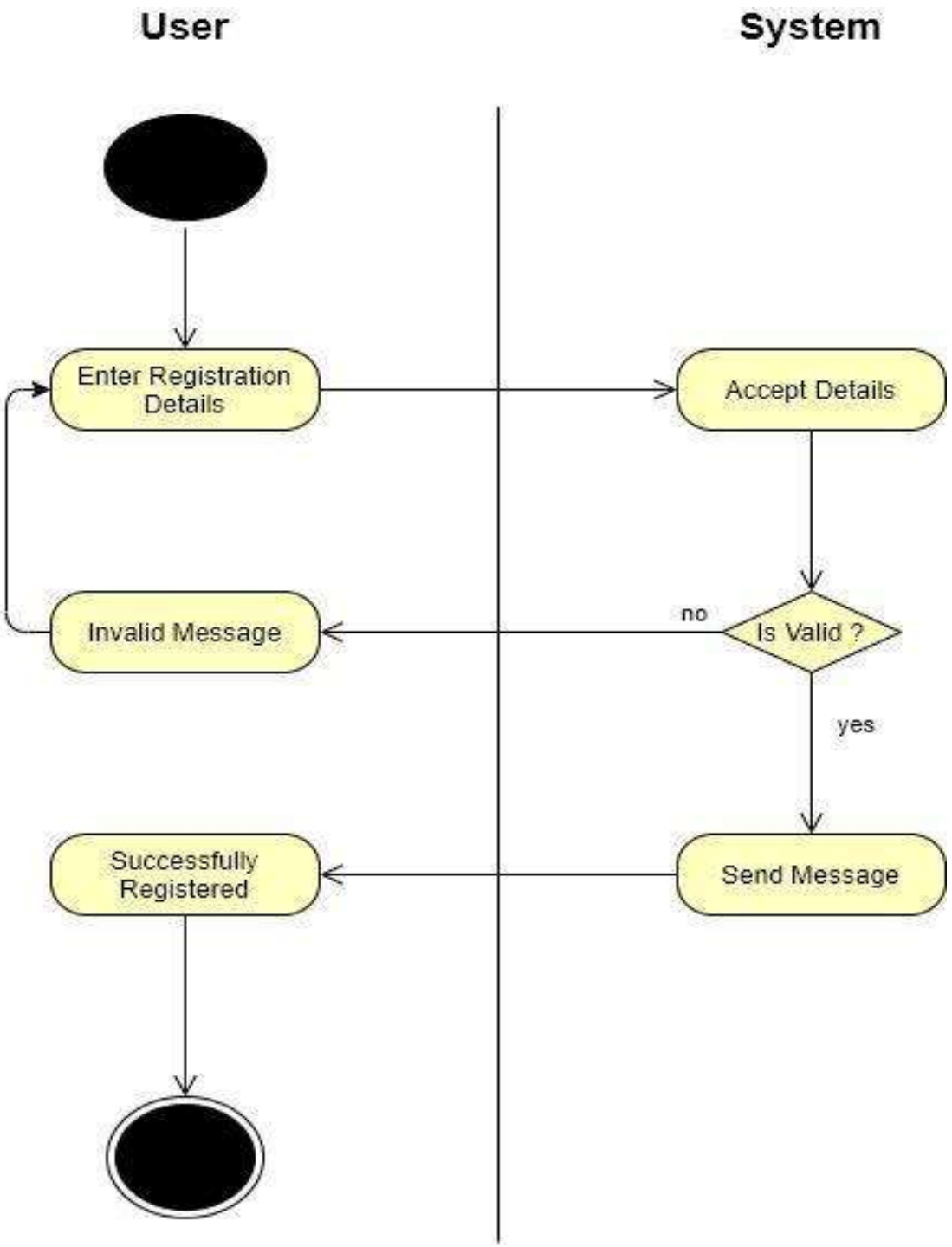


3.4 Interaction Diagram

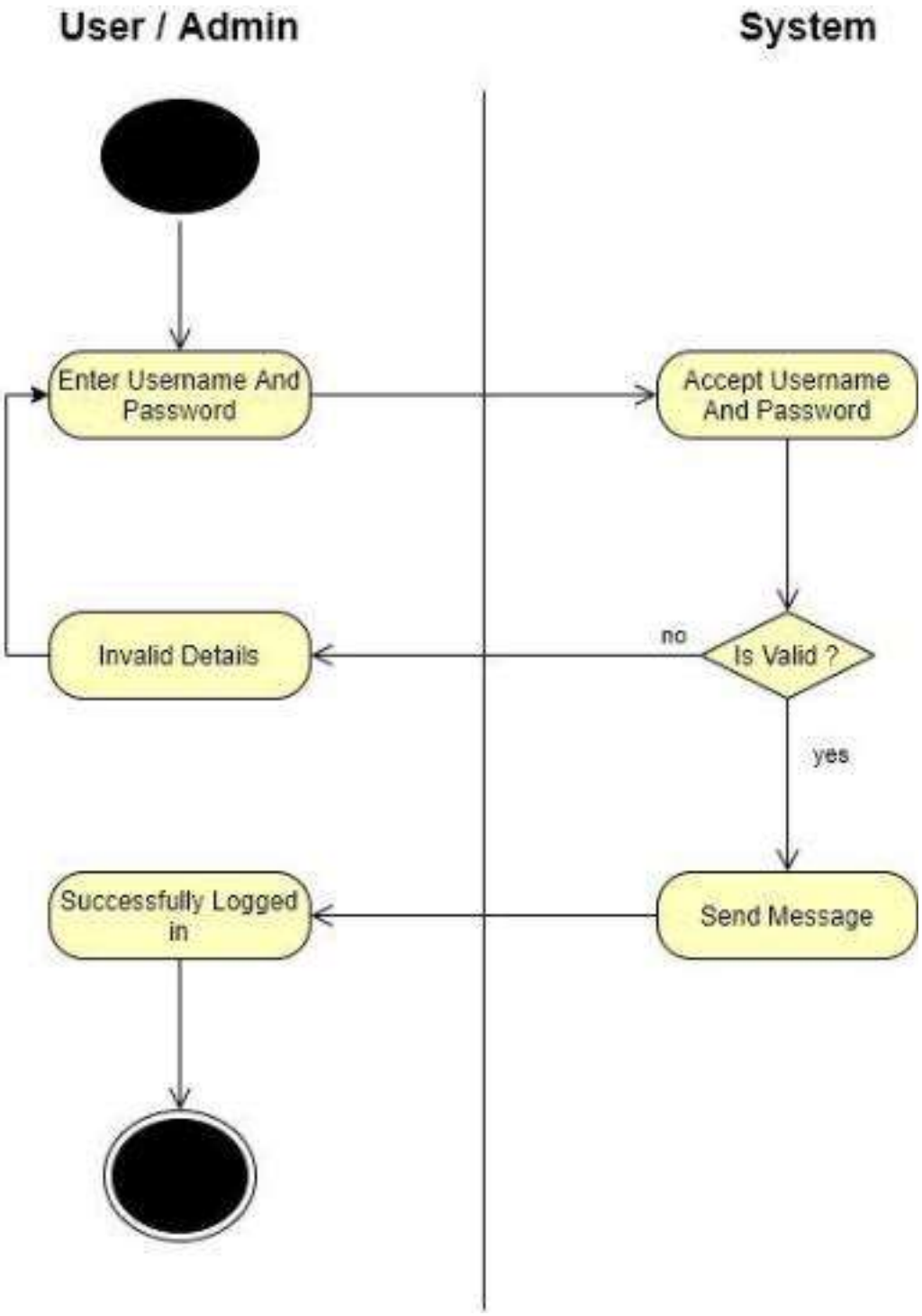


3.5 Activity Diagram

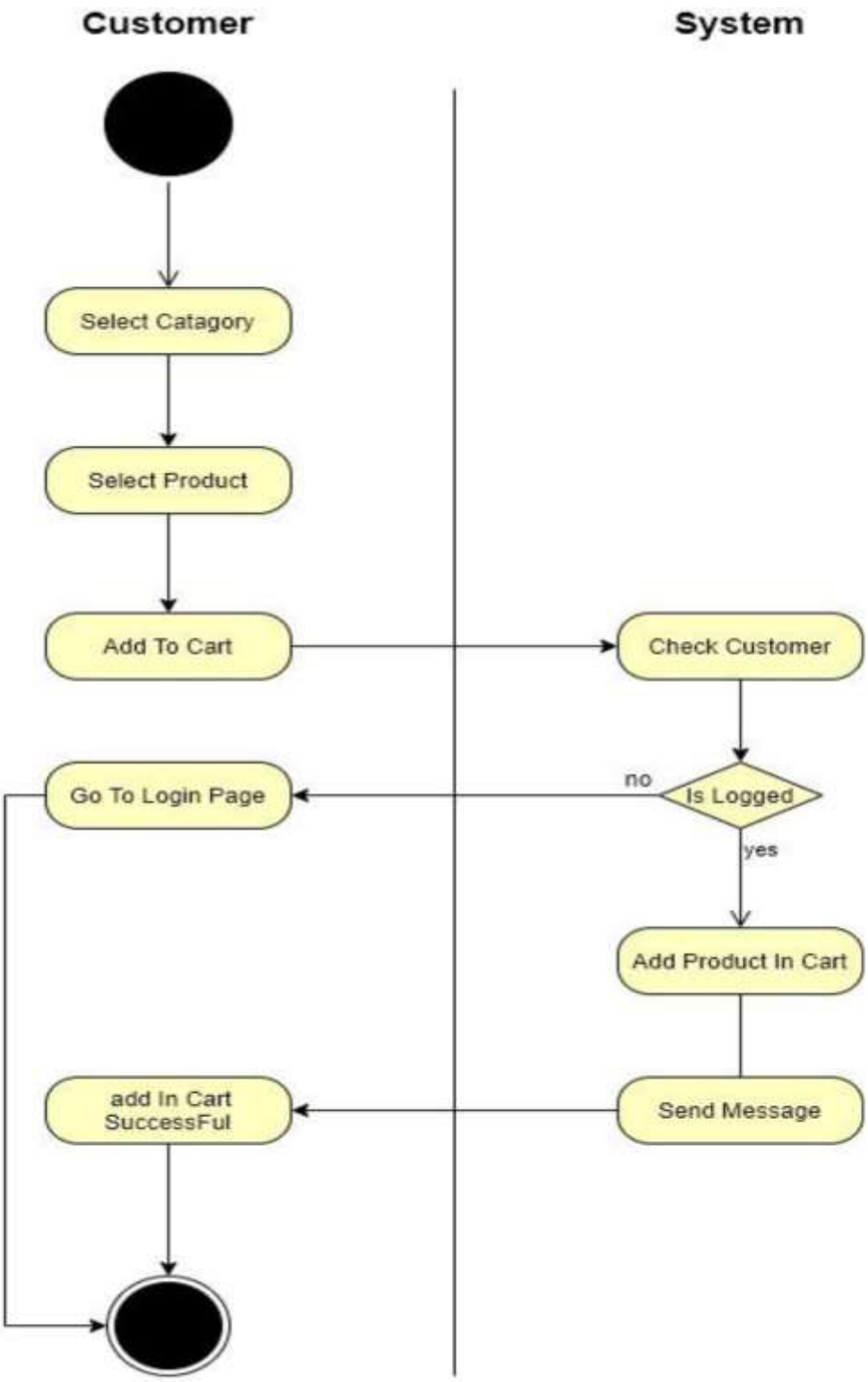
Registration



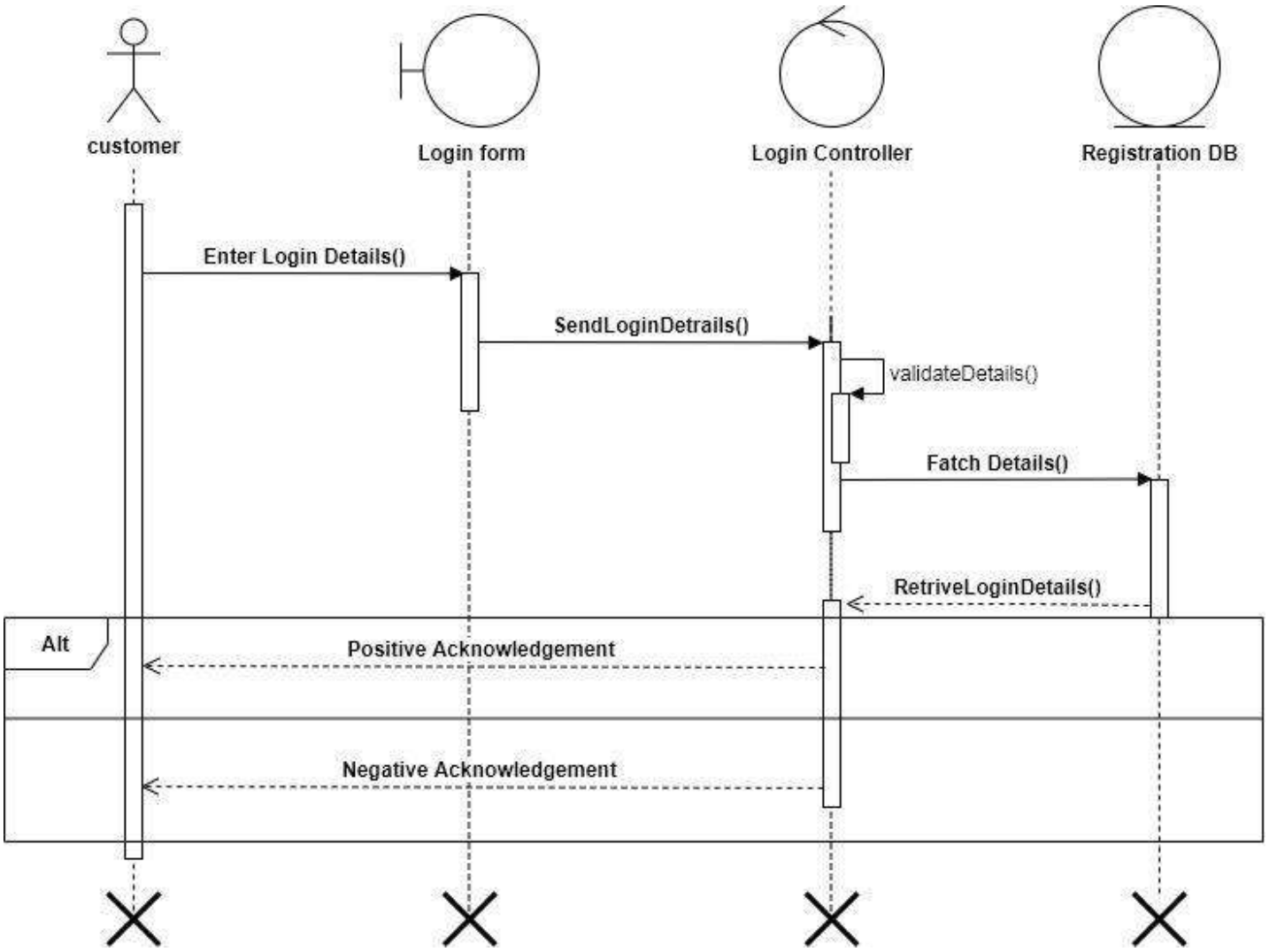
3.6 Login


User / Admin
System
Enter Username And Password
Accept Username And Password
Invalid Details
no
Is Valid ?
yes
Successfully Logged in
Send Message

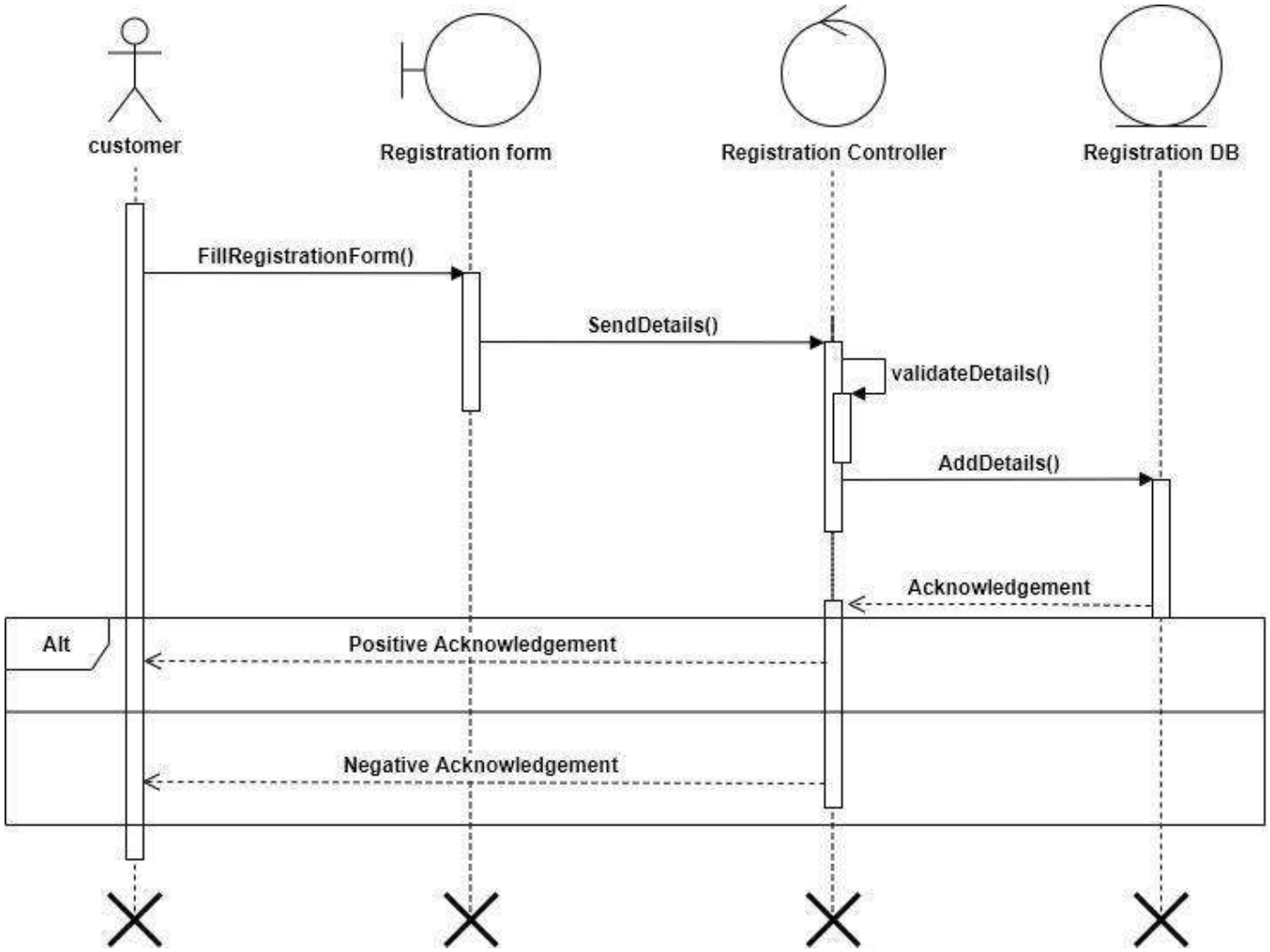
Select Product



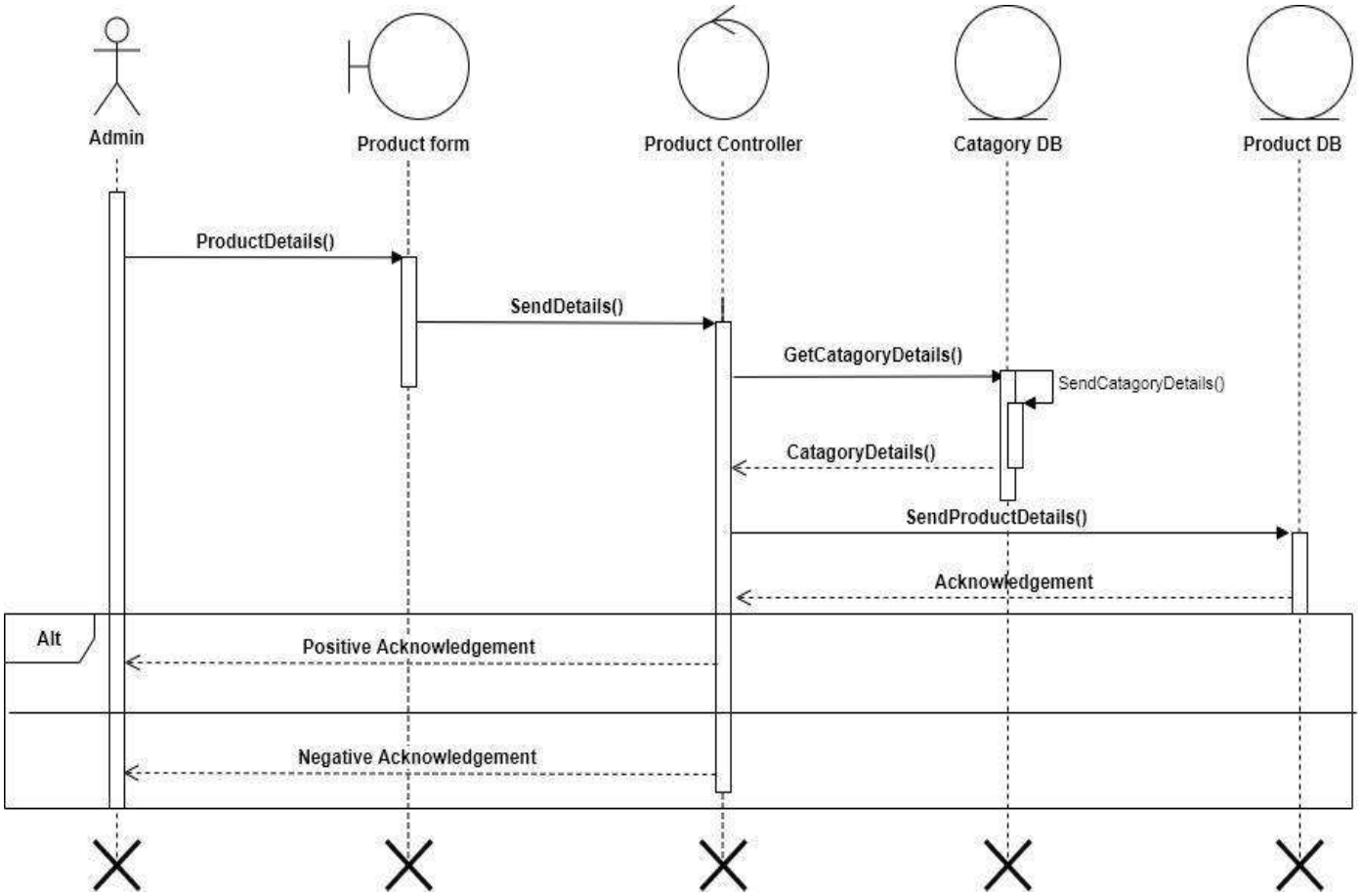
Interaction Diagram for Login:-



Interaction Diagram for SignUp:-



Interaction Diagram for Product:--



3.7 Data Dictionary

- The Data Dictionary can be specifically defining as an exhaustively organized list of all data elements that are pertinent to the system with precise, rigorous understanding of inputs and outputs and the components of stores along with all the constraints and intermediate calculations.
- In other words, a data dictionary is a catalogue – a repository of element in a system. Here in a data dictionary one can find list of all the elements composing the data flowing through a system. The major elements are data flows, data stores and process. The data dictionary stores the details and description of all these elements.

Importance of Data Dictionary:

- To manage the details in large system.
- To communicate a common meaning for all system elements.
- To document the features of a system. To facilitate analyst for the details in order to evaluate system requirements
- Following is the list of Tables which are used in my project. Consider Following Data Dictionary which denotes tables detail.

1. User table - this table contains the information of the users

Column Name	Data Type	Constrain	Description
Userid	Int	Primary Key	It is unique key for user
firstname	String(255)	Not null	It is First Name of user
lastname	String(255)	Not null	It is Last Name of user
username	String(255)	Not null	It is User Name of user
password	String(255)	Not null	It is Password of user
email	String(255)	Unique key	It is email of user
image	String(255)	Not null	It is Image of user
views	number	Not null	View user data
saved	String(255)	Not null	Save user data
isactive	Boolean	Not null	User active or not
Strip_customerid	String(255)	Not null	Payment user id
CreatedDate	date	Not null	Show created date
updatedDate	date	Not null	Show updated date
loginID	String(255)	Not null	Show login id

2. SubCategory table - this table contains the information of the sub-categories

Column Name	Data Type	Constrain	Description
id	String(255)	Primary Key	SubCategory id
Cat_id	String(255)	Foreign Key	Category id
Name	String(255)	Not null	Name of category
isActive	beelean	Not null	Isactive or not

3. Product table - this table contains the information of all the products

Column Name	Data Type	Constrain	Description
Categoryid	String(255)	Foreign Key	Category id
Sub_cat_id	String(255)	Foreign Key	Sub category id
Brand	String(255)	Not Null	Brand name
Size	String(255)	Not Null	Display size
description	String(255)	Not Null	Show item detail
price	number	Not Null	Show price
Discountprice	number	Not Null	SHOW Discountprice
Yourearning	number	Not Null	Your erning
condition	boolean	Not Null	Show condition
images	String(255)	Not Null	Show image
wishlist	String(255)	Not Null	Wishlist list show
name	String(255)	Not Null	Name of product
color	String(255)	Not Null	Color of product
Is_active	boolean	Not Null	Isactive or not
makeoffer	boolean	Not Null	Show make offer
createddate	Date	Not Null	Show create date
Created_id	String(255)	Foreign Key	User id

4. PaymentCheckout table - this table contains the information of all the payments done by customers

Column Name	Data Type	Constrain	Description
Categoryid	String(255)	Foreign Key	Show categoryid
userid	String(255)	Foreign Key	Show userid
Amount	Number	Not null	Amount of payment
Bank_id	Number	Not null	Bank id show
Is_payment_suucees	boolean	Not null	Payment success

5. Wishlist – this table contains the information of wishlist ofproducts

Column Name	Data Type	Constrain	Description
Userid	number	Foreign Key	userid
Product_id	number	Foreign Key	Product id
createdDate	date	Not null	Created date

6. StateSchema – this table contains the information of the states

Column Name	Data Type	Constrain	Description
Stated	number	Primary Key	Stated id
Name	String(255)	Not null	Name of state
Countryid	number	Foreign Key	Country id
Statecode	number	Not null	State code

7. Notification – this table contains the information of notificationof the customers

Column Name	Data Type	Constrain	Description
Name	String(255)	Not null	Name of notification user
Title	String(255)	Not null	Title of notification
Description	String(255)	Not null	Decription of notification
Price	Number	Not null	Price of product
Type	String(255)	Not null	Type of product
Userid	Number	Foreign Key	User id
Productid	Number	Not null	Product id
productOwner	String(255)	Not null	Name of send notification user

8. Makeoffer – this table contains the information of offers of theproducts

Column Name	Data Type	Constrain	Description
sellerid	number	Foreign Key	Seller id
Notificationid	number	Not null	Notification id
Customerid	number	Foreign Key	User id
Productid	number	Foreign key	Product id
Is_offer_active	boolean	Not null	Active status

9. Cities – this table contains the information of cities

Column Name	Data Type	Constrain	Description
Cityid	number	Primary Key	Cityid
Name	String(255)	Not null	Name of city
Stateid	number	Foreign Key	Show State id

10. Category – this table contains the information of cities

Column Name	Data Type	Constrain	Description
catid	number	Primary key	id of category
Isactive	boolean	Not null	Isactive or not
Name	String(255)	Not null	Name of category

11. BankDetails – this table contains the information of bank detailsof the customers

Column Name	Data Type	Constrain	Description
userid	Number	Foreign Key	User id
Accountholdername	String(255)	Not null	Card holder name
Instution_code	Number	Not null	Idec code
Transit_code	String(255)	Not null	Transit code
Account_number	Number	Not null	Account number

12.Address – this table contains the information of address of all theusers

Column Name	Data Type	Contrain	Description
Userid	Number	Foreign Key	User id
firstname	String(255)	Not null	Firstname of user
Lastname	String(255)	Not null	Lastname of user
Address	String(255)	Not null	Address of user
City	String(255)	Not null	City of user
State	String(255)	Not null	State of user
Country	String(255)	Foreign key	Country of user
State_code	number	Foreign key	State code
Country_code	Number	Foreign key	Country code
Phone	Number	Not null	User phone number

13.County – this table contains the information of countries

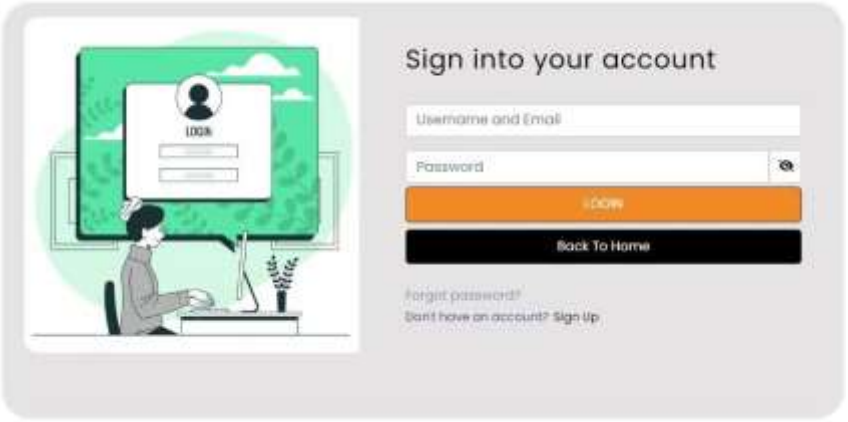
Column Name	Data Type	Constrain	Description
Country_id	Number	Foreign Key	Country id
Name	String(255)	Not null	Country name
Currency	String(255)	Not null	Currency
Region	String(255)	Not null	Region

14.ContactUs – this table contains the information of all the orders

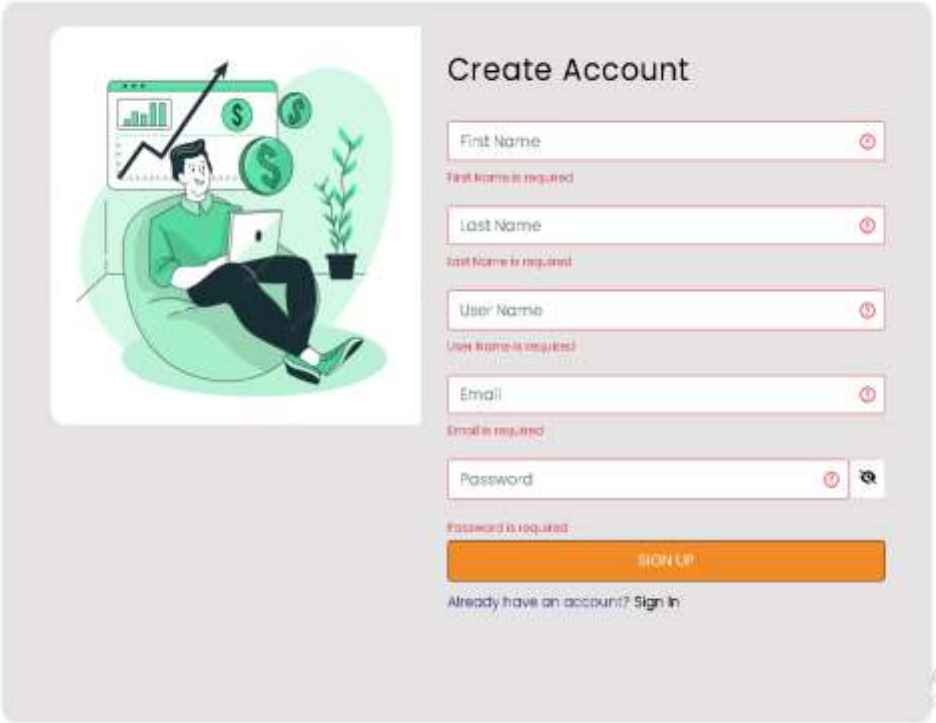
Column Name	Data Type	Constrain	Description
id	Number	Primary Key	Id of user
Name	String(255)	Not null	Name of user
Email	String(255)	Unique key	Email of user
Message	String(255)	Not null	Message of user

4 Development

4.1 User Sign-in



4.2 User Sign-up

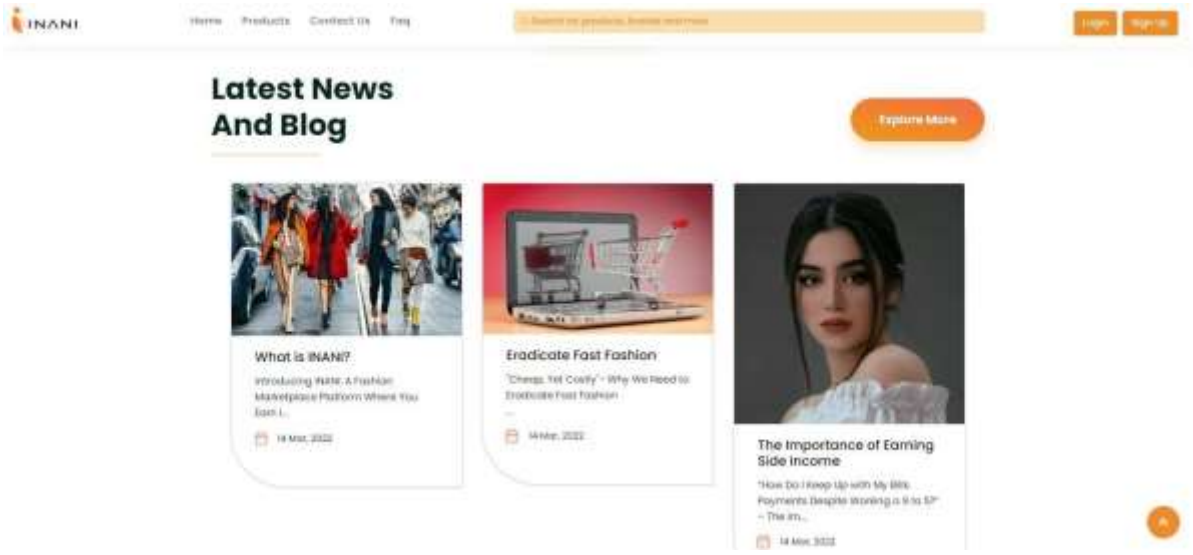


4.3 Home

4.4 About

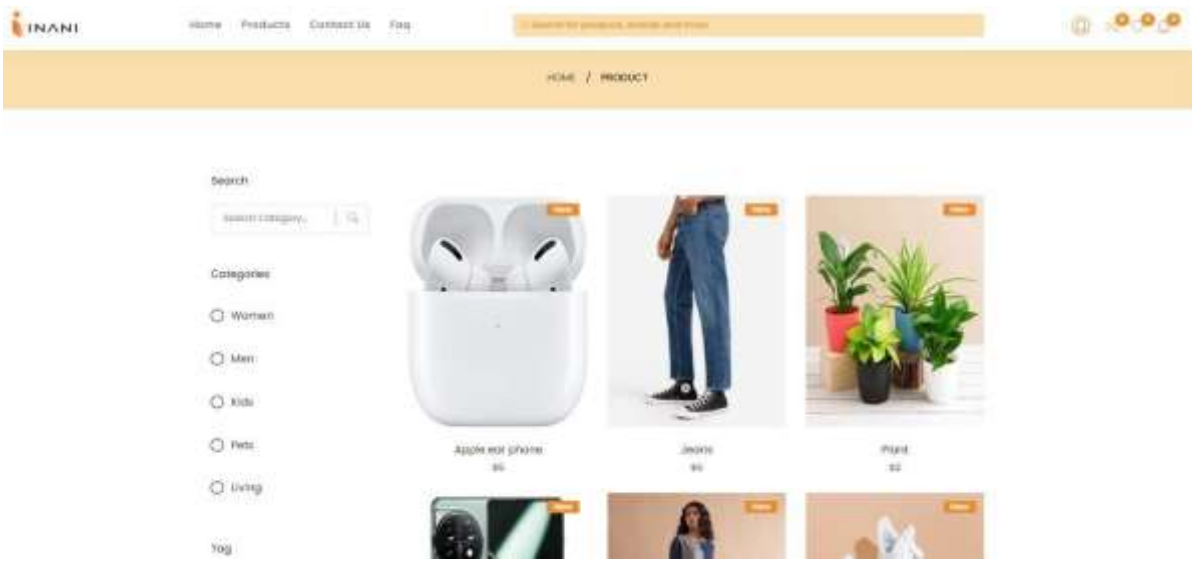
4.5 Buyer's information

4.6 Blog's information

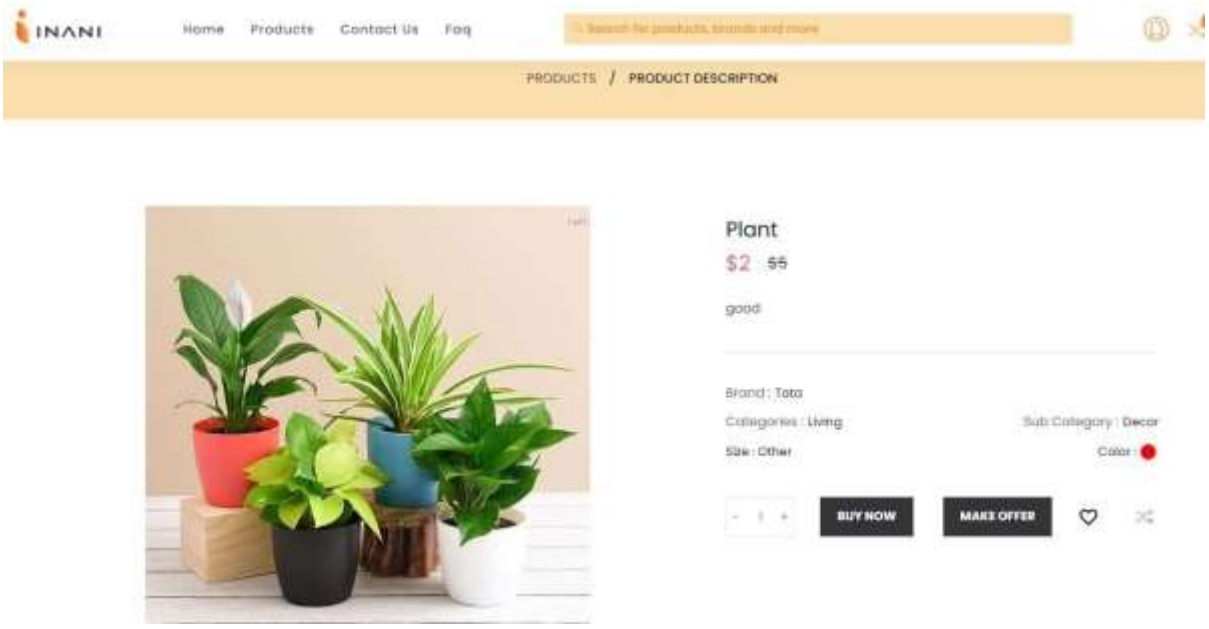


4.7 FAQ’s information

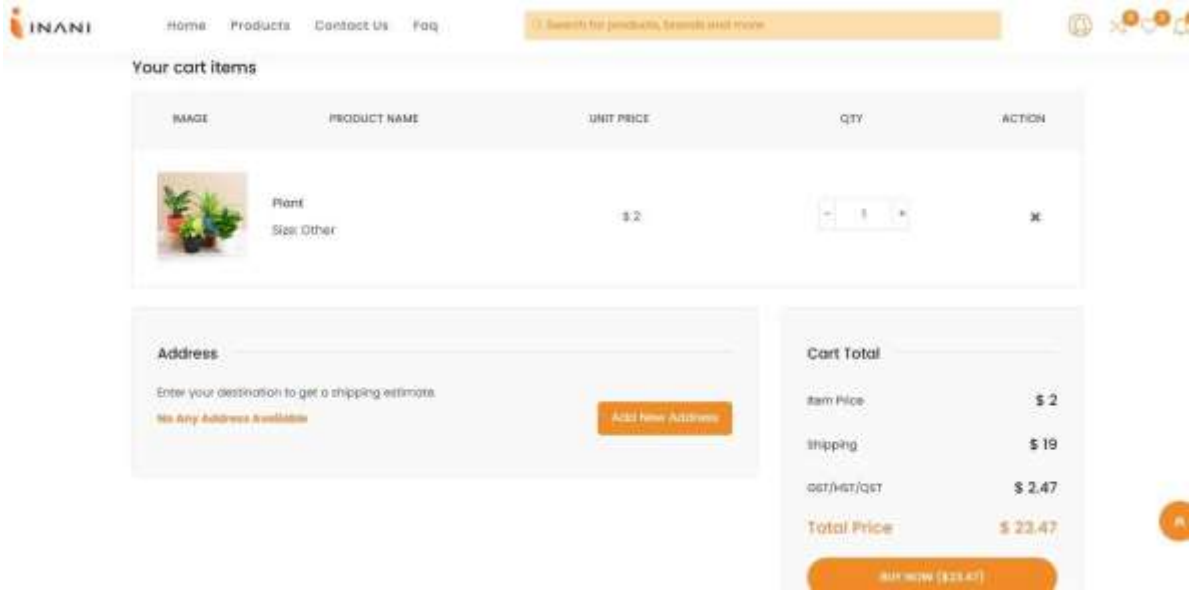
4.8 Product’s information



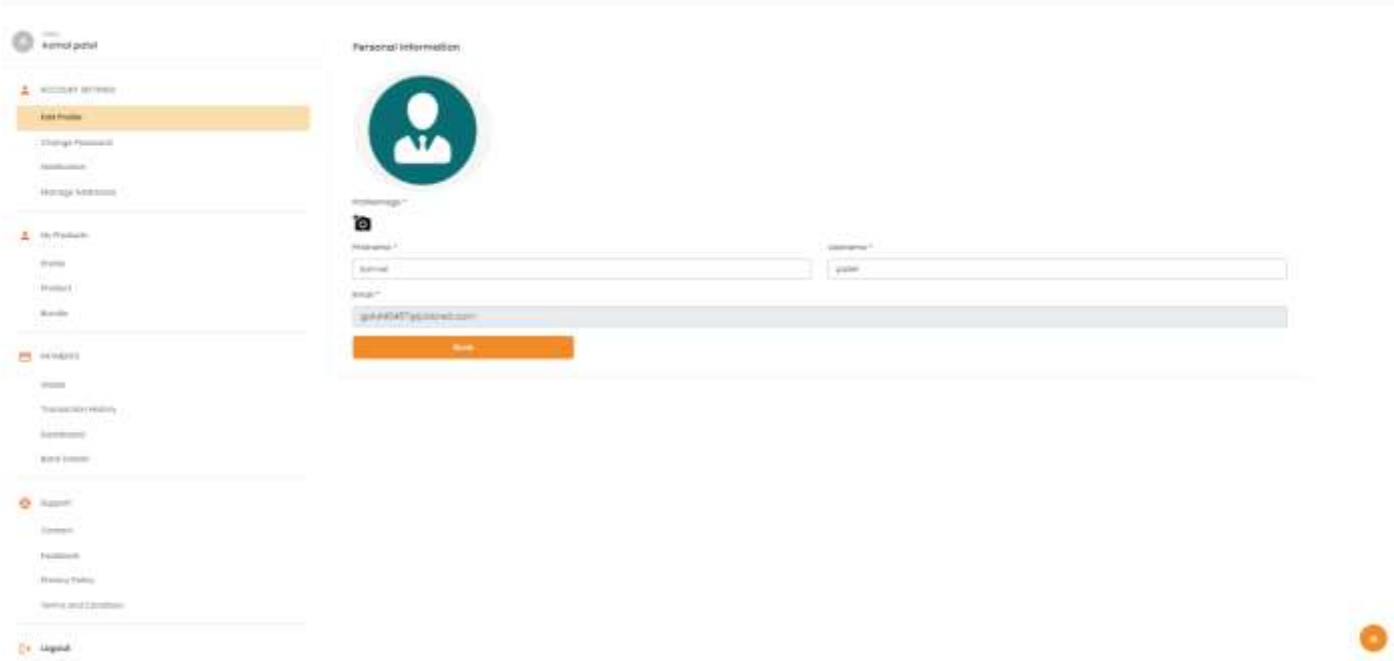
4.9 Product's Details



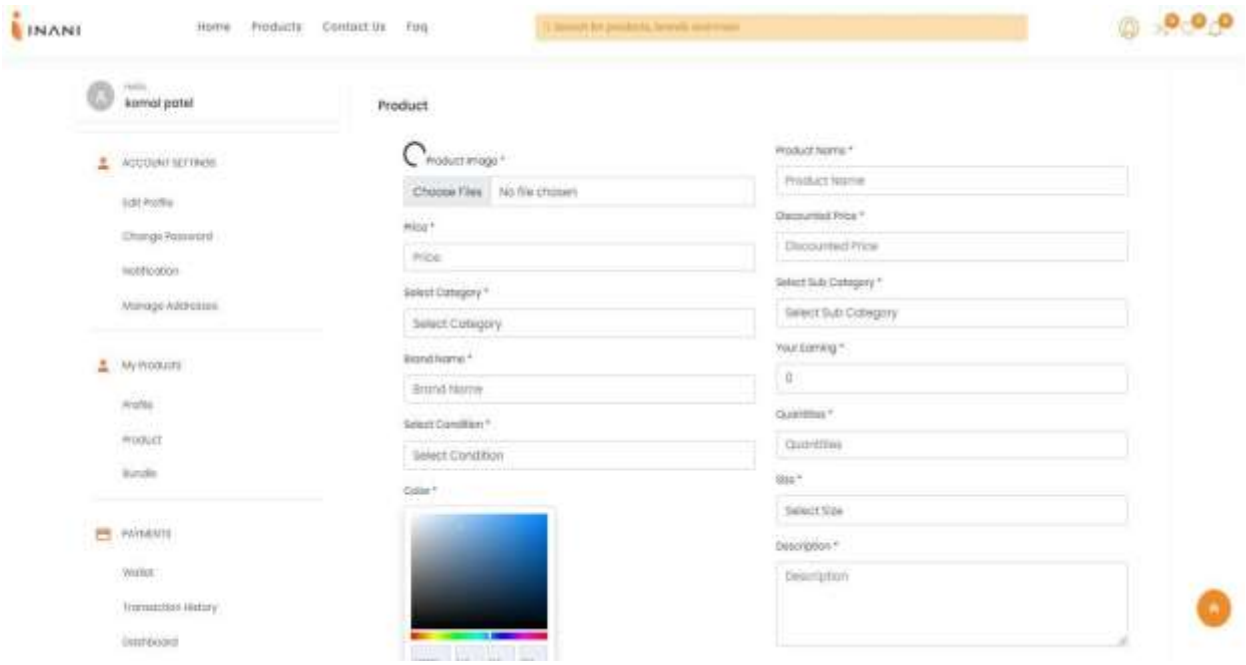
4.10 Selected Item



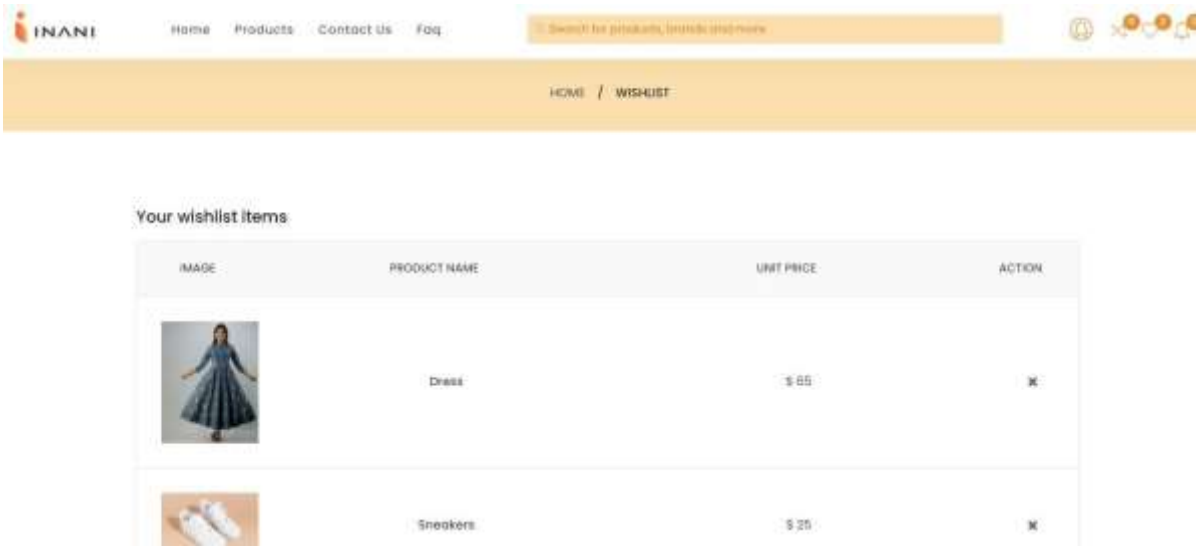
4.11 Profile



4.12 Upload Item



4.13 Wishlist

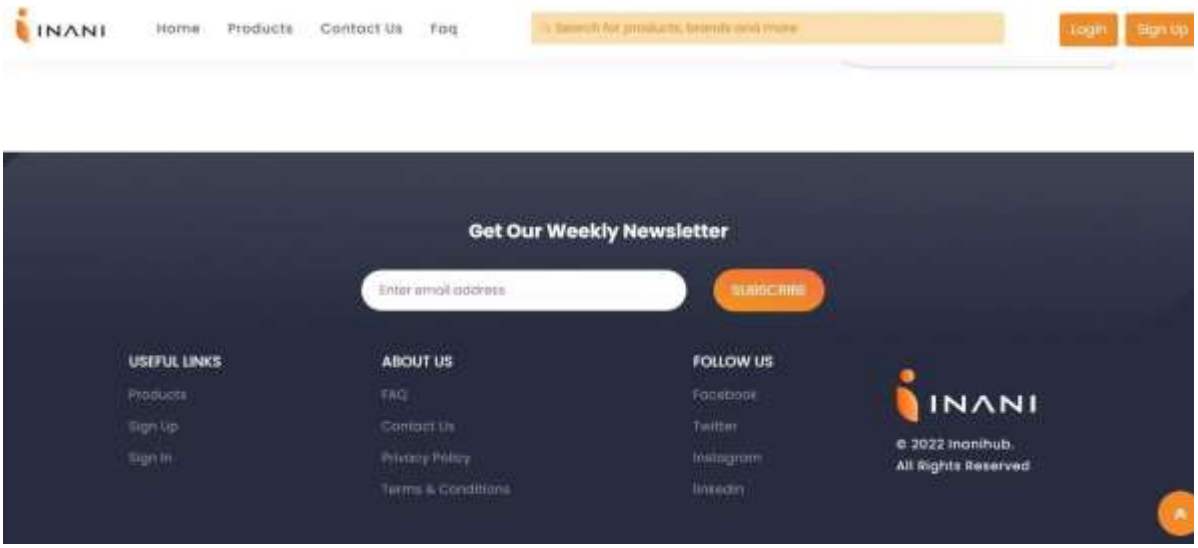


4.14 Notification

4.15 Contact US

4.16 Categories

4.17 Footer


INANI
Home
Products
Contact Us
Faq
Login
Sign Up
Get Our Weekly Newsletter
Enter email address
SUBSCRIBE
USEFUL LINKS
Products
Sign Up
Sign In
ABOUT US
FAQ
Contact Us
Privacy Policy
Terms & Conditions
FOLLOW US
Facebook
Twitter
Instagram
linkedin
INANI
© 2022 Inanihub.
All Rights Reserved

5. Agile Documentation

5.1 Agile Project Charter

- Agile development and project management processes use a framework to adapt products quickly by shortening the release cycles of new features. In the context of retail, agile ecommerce is the rapid development of, and updates to, websites with new features or products, while reducing operational risks by improving code quality. In most cases, software development teams use either a Scrum or DevOps approach when working in agile development environments.

5.2 Agile Roadmap

An agile roadmap using quarters would look something like this:

Quarter	Q1	Q2	Q3	Q4
Outcome	Sign-up/in Module	Customers	Sellers	Visitors
Features	Cerate the Sign-up and sign-in module for all the users	Create all the facility related to the customer Create the facility for the customer's products Create the facility related to payment	Create all the facility related to the seller Create the facility for the seller's product Create the facility related to payment	Create all the facility related to the visitors

5.3 Agile Project Plan

- Agile planning is a project management style with an incremental, iterative approach. Instead of using in an in-depth plan from the start of

the project which is typically product related Agile leaves room for requirement changes throughout and relies on constant feedback from end users.

❖ Over a defined period of time, cross-functional teams work on product iterations and achieving OKRs (objectives and key results), organizing their work into backlogs that focus on delivering value. The ultimate goal of each iteration is to produce a working project.

5.4 Agile User Story

There are three different kinds of user stories that we typically encounter:

- Behavior-driven stories
- Granularity of a Story
- Reliability of a story

Once a Product Owner is able to recognize the type of story needed, it becomes easier for them to focus on the criteria that are important for the story acceptance.

- **Behavior-Driven Stories :**

These are stories where the user's actions or decisions are the focus. These stories typically have a number of scenarios to be considered.

For example: As a customer, I need to authenticate myself so that I can see my account details and past orders.

In this example, there are multiple possible scenarios:

1. User entered the correct credentials for authentication
2. User entered incorrect credentials for authentication
3. User realizes that they forgot the credentials
4. User realizes that they do not have an account, and wants to create one now

For each scenario, we need the acceptance criteria defined. Use simple language in an active voice to state what the product needs to do. Each scenario is expected to have a "WHEN" and a "THEN" phrase. If the scenario is dependent on a specific pre-condition, then the scenario would need a "GIVEN" phrase too.

GIVEN <situational pre-condition>

WHEN <user action 1> and <user action 2> and … <user action n>

THEN <product action 1> and <product action 2> and … <product action n>

The earlier user story now would look as follows: As a customer, I need to authenticate myself so that I can see my account details and past orders.

Acceptance criteria:

Scenario 1: Successful authentication

When the user enters the correct email address and password and selects “sign-in” CTA

Then route the user to My Account home page and display signed-in status on header

Scenario 2: Unsuccessful authentication

When the user enters an incorrect combination of email address and password and selects “sign-in” CTA

Then reset credentials fields and display an error message “Incorrect Credentials”

Scenario 2: Unsuccessful authentication

When the user enters an incorrect combination of email address and password and selects “sign-in” CTA

Then reset credentials fields and display an error message “Incorrect Credentials”

Scenario 4: Register

When the user selects the “register” CTA Then

route the user to the “registration” page

- **Granularity of a Story:**

Regardless of the type of story that is being written, there are two rules that govern the granularity of a story.

The first rule: The story needs to represent a meaningful product increment to the product user.

For example, if we are building a registration page, having two stories - one to display the form, and another to submit the form doesn't make sense. Each individual story in this case doesn't make any sense to the product user. We need asingle story that can display the form and accept form submission. If there are any optional independent sections on the form (say a loyalty registration section), we could have a separate story to cover just that.

The second rule: The story needs to be small enough for the team to implement in one sprint.

A story may need to be split into multiple stories depending on the team's experience and sprint duration. It's perfectly fine to wait until refinement or planning with the team before doing this splitting.

- **Reliability of a Story :**

It seems obvious, but it's worth stating: a story needs to be easily readable.Here

are some tips:

- Pay attention to spelling and grammar.
- Use numbered lists wherever possible. In cases where numbering doesn't make sense for a list, use bullets.
- Use indentation where applicable.
- When providing links to UX/UI artifacts, or other specifications as part of acceptance criteria, use plain text for the "hyperlink text" instead of the URL to avoid clutter.
- Use white space (line breaks) between sections or scenarios.

5.5 Agile Release plan

- Agile release planning is an approach to product management that takes into account the intangible and flexible nature of software development—as part of this approach, teams plan iterative sprints across incremental releases.
- This level of planning, combined with an iterative schedule to account for the dynamic nature of software, is what makes Agile product development so valuable.
- The iterative release schedule gives teams the space to make course corrections without derailing the entire project, while the detailed roadmap and focus on the planning stage ensure everyone is on the same page.

5.6 Agile Sprint Backlog

- A well-prioritized agile backlog not only makes release and iteration planning easier, it broadcasts all the things your team intends to spend time on including internal work that the customer will never notice. This helps set expectations with stakeholders and other teams, especially when they bring additional work to you, and makes engineering time a fixed asset.
- A sprint backlog is a list of work items your team plans to complete during a project sprint. These items are usually pulled from the product backlog during the sprint planning session. A clear sprint backlog prevents scope creep by clarifying exactly what your team will be doing and not doing during each sprint

5.7 Agile Test Plan

Types of Testing for E-commerce System :

Sr No.	Type of testing	Testing process

1	Browser compatibility	• Lack of support for early browsers • Browser specific extensions • Browser testing should cover the main platforms (Linux, Windows, Mac etc.)
2	Page display	• Incorrect display of pages • Runtime error messages • Poor page download time • Dead hyperlink, plugin dependency, font sizing, etc.
3	Session Management	• Session Expiration • Session storage
4	Usability	• Non-intuitive design • Poor site navigation • Catalog navigation • Lack of help-support
5	Content Analysis	• Misleading, offensive and litigious content • Royalty free images and copyright infringement • Personalization functionality • Availability 24/7
6	Availability	• Denial of service attacks • Unacceptable levels of unavailability
7	Back-up and Recovery	• Failure or fall over recovery • Backup failure • Fault tolerance
8	Transactions	• Transaction Integrity • Throughput

		• Auditing
9	Shopping order processing and purchasing	• Shopping cart functionality • Order processing • Payment processing • Order tracking
10	Internationalization	• Language support • Language display • Cultural sensitivity • Regional Accounting
11	Operational business procedures	• How well e-procedure copes • Observe for bottlenecks
12	System Integration	• Data Interface format • Interface frequency and activation • Updates • Interface volume capacity • Integrated performance
13	Performance	• Performance bottlenecks • Load handling • Scalability analysis
14	Login and Security	• Login capability • Penetration and access control • Insecure information transmission • Web attacks • Computer viruses • Digital signatures

5.8 Earned-value and burn Charts

Agile EV provides the data of **cost**, **performance** and **schedule** allowing to:

- Compare monetary expressions of value planned, earned and consumed at any given moment
- Calculate cost and schedule efficiency
- Make forecasts for the end-of-the-project time and cost

Earned value **(EV) measurement** and techniques, as methods for project **management monitoring, reporting, forecasting,** and **controlling have been developed** and **adopted over the past few decades** in software product development as well as traditional engineering projects.

6 . Proposed Enhancements

In Future we will add GPS system that user choose their location.

In future we will add the provide the Online Payment Facility.

In future we can make all language independent so that User can Easily access it.

7 . Conclusion

Online shopping brings us great convenience, but it also encourages irresponsible consumption habits like exploiting the advantages of free returns and expedited shipping.

These add on to the existing pool of environmental problems that we are dealing with global warming, wastes and pollution.

8. Bibliography

Websites - www.w3schools.com

www.javatpoint.com

www.geeksforgeeks.org

www.tutorialspoint.com

www.stackoverflow.com

8. Bibliography

Websites - www.w3schools.com

www.javatpoint.com

www.geeksforgeeks.org

www.tutorialspoint.com