UTA Marketplace

Market Place for students of The University of Texas, Arlington

CSE-5324-003-SFWR ENG I ANLY, DSGN, TESTING

Group Name: SCRUM
Group 14

Date: 10/04/2022

Team Members:
Manali Gandhi: 1002084355
Gaurav Taneja: 1001955801
Divya Darshi: 1002090905
Akash Upadhyay: 1001957955

PROJECT DESCRIPTION

The project is the solution to a major problem happening around the university to the students living in on-campus housing or nearby areas. A lot of students come every year to UTA and shift to the on-campus housing or the area near the university but don’t find enough affordable stuff to manage their livelihood like beds, bed-frame, electronic items, gadgets, books, transportation, study desk, chairs, dining table, furniture, wardrobe, etc. In such cases few students who are lucky enough, buy items for themselves whereas some don’t get anything, so we have decided to design an app that will let the students who have some old yet usable stuff with them and do not use themselves, add those items to the app and let any buyer see it or buy it.

The project will be a real-time solution for a real-time problem. It will be built using Android Studio's XML and Java for the UI part, and MySQL will be utilized to maintain the database and gather all the data. The backend will be developed in python and the front end and back end will be connected using the Django framework.

The app will essentially function as a classified marketplace where users can purchase and sell a variety of goods online. The software is designed with user-friendliness in mind so that anyone may use it without having any trouble. The user must log in to the app at the very beginning, and if they don't already have one, they must make one. The user must log in with his password after providing all his information on the initial sign-in. The app's whole body of work is displayed on the home screen in the form of sliding photos. Along with these photographs, he can also see numerous things that have been submitted by various app retailers and are recommended for him. Additionally, he can access a specific product category by selecting it from the home screen. Users have access to every product's detail as well as the ability to interact with the vendor. To view products later, he can add them to his favorites. When a product is ready, he can use our payment channel to purchase it.

Users can also upload products if they want to sell something. All he must do is click the + button below, select the category of the product he wants to sell, and add details such as make, model number, age, year of registration, cost, photos, and location. .and submit it. If buyers like your product, they can chat with you and purchase your product.

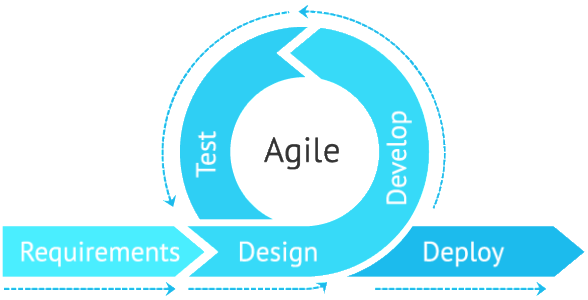
OVERALL DESIGN APPROACH

The basic design approach used to develop an app is defining the problem, identifying the requirements, generating and evaluating concepts, modifying concepts, choosing a concept, developing a functional prototype, and demonstrating it.

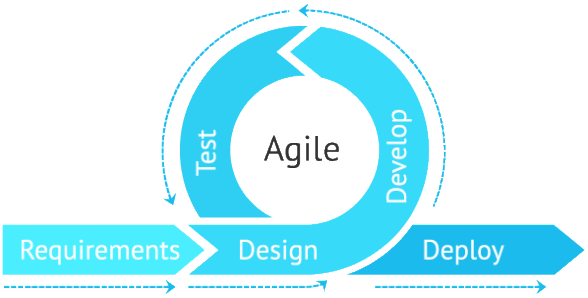
In software engineering, we have n number of designing models to follow the steps and design accordingly. Depending upon the size of the project and the time we have to develop it, we choose the model. While designing this app we came across a few software engineering models like the waterfall model, prototype model, V-model, agile model, Spiral model, etc. and out of all these models we decided to choose an agile model as it is the most reliable and friendly model to use.

The basic steps of the agile model are depicted in the picture below

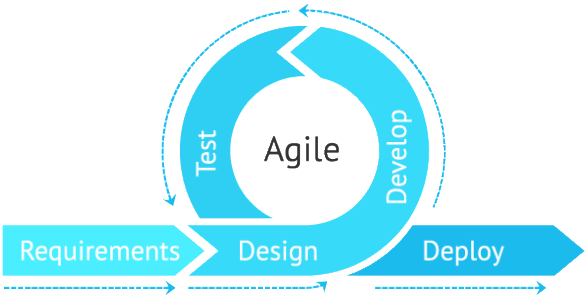
Iteration - 1



Iteration - 2



Iteration - 3



SYSTEM DESIGN

The section describes the system study, analysis, and design of the system and process flow diagram. Every project is required to achieve a set of objectives, considering several conditions like, it should be easy to use, workable, attractive, and user-friendly. The purpose of this project is to develop a platform or a system where students can search for some old yet reusable goods in their nearby area, this system will be developed considering the conditions stated above i.e., easy to use, workable, user-friendly, and so on. It helps in effective and efficient order management. With each shoot, the collection becomes clear, simple, and meaningful.

The system will be designed in such a way that each time a user can be either buyer or seller or both. After each step, there is a certain condition and the user cannot move further without completing that step.

Let us understand the system design in more detail using a flow diagram:

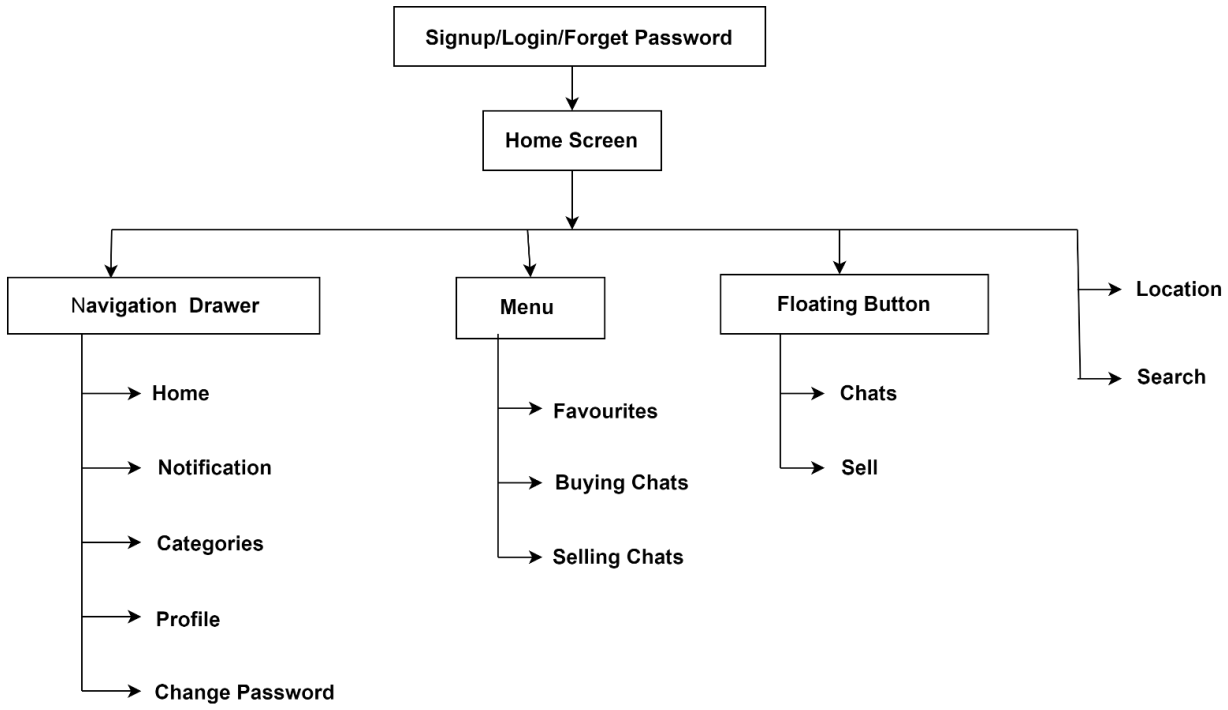


Fig: Flow diagram to understand system design

Every component performs some specific tasks and is a group of some related functions. The figure above shows the basic architecture of the app. Each component shares its information using an interface here home screen is the component and all other modules inside the home screen are its sub-components they are connected through an interface to each other.

Login/ Signup/ Forget Password

As with every other application, our app will initially have a login/signup screen with forget password functionality. On clicking the forget password button, the app will prompt the user to enter his registered

mail id and we will send a link to that mail address to reset a new password for the account. There will be a hidden button for admins to log in. Admin cannot be registered through the app but users- buyers and sellers can be registered using the signup page.

Home Screen

On successful login, the user is redirected to the home screen where he has different options to explore the app. On the top, there will be a navigation panel where the user can find their full address and can also change it whenever he wants to use the navigation panel. We will also have a dynamic set of items displayed on the home screen like the feed we have on Facebook. The user can see and add items to a favorite or view the item from there itself and buy them or chat with its specific seller.

On the top left, there is a navigation drawer from where a user can navigate to different screens like home, favorites, categories, my profile, change password, and logout. The categories option classifies various categories like- cars, bikes, mobile phones, laptops, properties, or furniture. Further, we have subcategories like car hardware or car tools, mobile accessories or mobile phone, laptop accessories or laptops, home decor or student's furniture, etc.

On the top right there will be three dots known as a menu which will contain the following options: Notifications, buying chats, and selling chats. This menu option will provide easy access to data to the user and help him to organize his chats based on the type he has signed in.

On the bottom right corner, we will have two floating buttons. One for adding new items to the feed by the seller and one for opening all the chats.

Favorites

Once the user clicks on the heart of an item the item is added to the favorites list and when he clicks again the item is removed from the favorite list. The items selected as favorites can be viewed on the home screen itself and the user can also see his favorites using the navigation panel on the left. This feature allows user to access all their liked products in one place.

Sell Floating button

The sell button is for the sellers who wish to sell any item from their belongings. The sell button prompts the seller to choose between 6 options of selling which item. That six options will be a car, bike, mobile, laptop, furniture, and property. When the user selects any of the options, a specific form opens asking for every single detail of the product and uploading the image and address, the costing, etc. Once the details are filled and the item is uploaded, it is available in the home screen feed of the user and the user can find it by specifically going to that category of product.

Location

We will be integrating navigation using google maps to add address flexibility for the users. Using this feature the user can easily find his exact location and that location will be used to upload a new selling item to the feed.

Search

The search functionality will allow the user to easily look for items he wants and see a list of all such items in the database. This functionality also helps in the accessibility feature of the app and the user does not have to hustle in finding what he wants

REQUIREMENT

Functional Requirements:

Administration Module

- Admin can provide a username, email, password, and admin account Created.
- After logging in, there is a dashboard where admins can see the number of customers Register the number of products sold, and the number of orders placed.
- Administrators can add/delete/view/edit products.
- Administrators can view/edit/delete customer data.
- Administrators can view/delete orders.
- Administrators can change the status of an order (order is pending, confirmed, in preparation for delivery, Delivered)

Users Module

- Users can view/search products after logging in.
- Users can also sell their products.
- Users can also add/remove products to the cart.
- When a user tries to purchase a product, they must log into the system.
- You can place an order after creating an account and logging into the system.
- The payment is successful when the user clicks the Pay button.
- Users can view order details by clicking the Orders button.
- Users can see the status of each order (pending, confirmed, shipped).

Non-Functional Requirements:

Specifies the quality features of the software system. they judge software systems based on their responsiveness, usability, security, portability, and other non-functional criteria Critical to the success of any software system.

- ***Availability***:

  Systems need to stay up and running every day, everywhere.

- ***Accuracy:***

  The system needs to be fine-tuned to ensure more accurate results and calculations.

- ***Usability:***

  The system should provide a user-friendly interface and tooltips for improvement Respond effectively and independently.

- ***Security***:

  The system should be able to provide security against external injection multi-layer security systems. Also, the implementation of the user login function ensures that the system is Protected from unauthorized persons.

- ***System Performance:***

  Response times are very good for the work given. the system Support multi-user environments.

- ***System Reliability:***

  The system is very reliable and produces all updated Information in the correct order. Data validation and verification are done at every stage of the activity.

Interface Requirements:

This section describes the necessary hardware components and software requirements. Effective and efficient operation of the system.

Hardware Interfaces:

Laptop

- Core i5 or i7
- Hard Disk 1TB
- Minimum 4GB RAM

Software Interfaces:

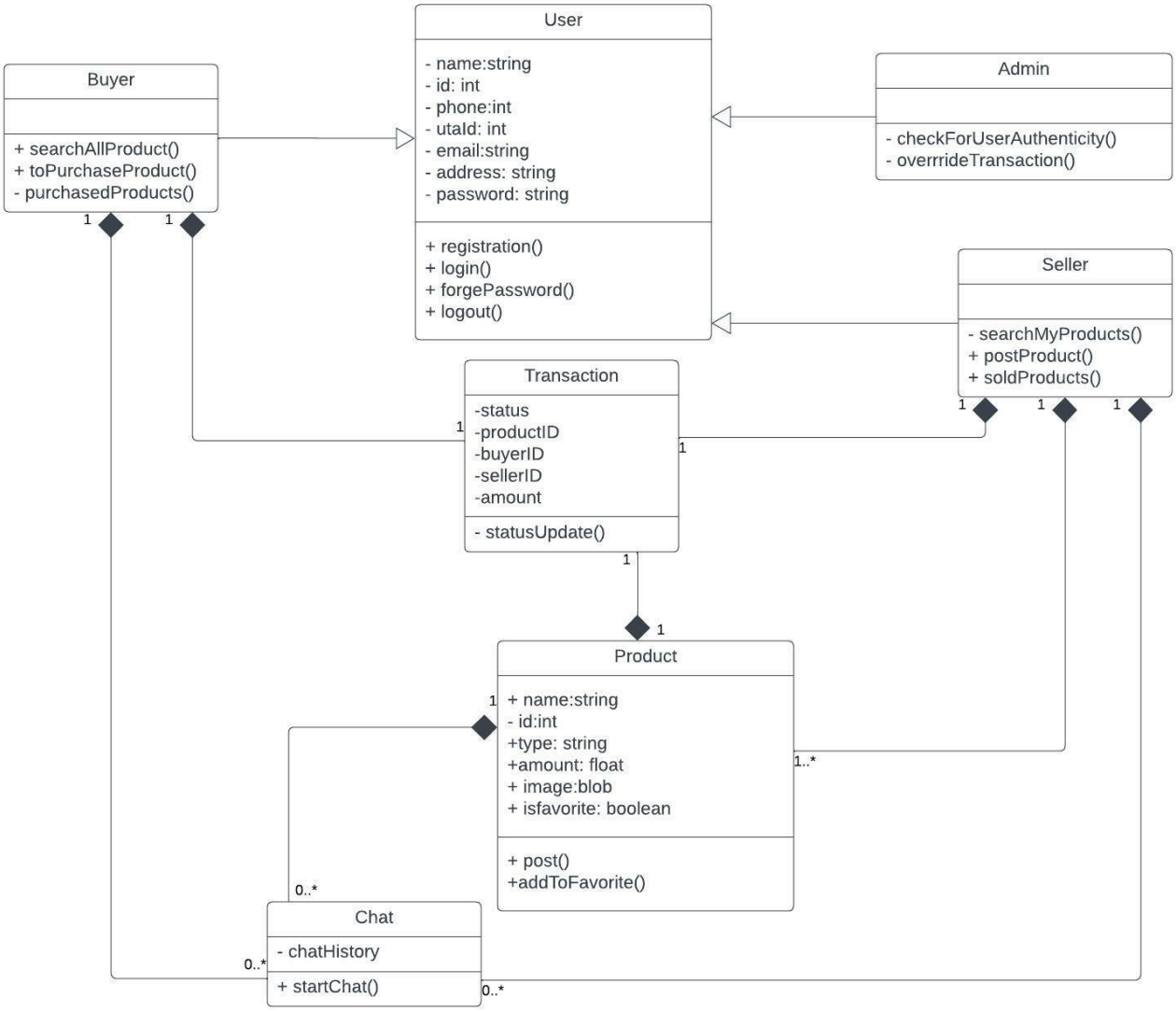
- Windows or Mac
- Android studio
- ADT Bundle XAMPP Server (Apache & MySQL)
- Java Development Kit

DOMAIN MODEL AND ACTOR SYSTEM INTERACTION MODEL

Domain modeling is a conceptualization process. It aims to identify important domain concepts, their properties, and relationships between the concepts. The result is portrayed in a diagram called a domain model.

Class and Object:

A class is a type, an intentional definition of a concept. A class encapsulates its attributes and operations that characterize the instances of the class. An object is an instance of a class. The attributes and operations of a class are also cal1edfeatures of the class. A class is a type, be it a built-in type or a user-defined type. In the object-oriented paradigm, the predicate that defines a class is the set of attributes, and operations of the class Classes can be shown using either the compact view or the expanded view, we are using the expanded view. Attributes can be entered from an input device (like a keyboard) but objects cannot; objects are created by calling a constructor.



UC1: Registration and UC2: Login/logout

Actor: User	System: Marketplace
	0. System displays the registration page
1. User enters name, email-id, Uta-id, phone number, and password and clicks on the register button.	2. System validates the field and stores the provided data in the database. Redirects to the login page.
3. User adds his email and password and clicks on the login button.	5. System verifies the details and redirects the user to the homepage.
6. User clicks on the logout button.	4. System logouts the user.

UC3: Purchase item

Actor: Buyer	System: Marketplace
	0. System displays the login page
1. Buyer adds his email and password and clicks on the login button	2. System verifies the details and redirects the user to the homepage.
3. The buyer explores the item from the homepage and selects any item he wants to buy	4. System shows the details for a specific item.
5. a) Buyer can click on buy option b) Buyer can click on chat with seller option	6. a) System will redirect to the checkout page b) System will redirect to the chat page with a specific seller.

UC4: Sell an item

Actor: Seller	System: Marketplace
	0. System displays the login page
1. Seller adds his email and password and clicks on the login button	2. System verifies the details and redirects the user to the homepage.
3. The seller clicks on add item button	4. System displays the list of different categories of items
5. a) Seller can choose any category	6. a) System will redirect the seller to enter the item details.
7. Seller will click on the sell item button	8. System will add the item to the database and will be available for buyers to buy.

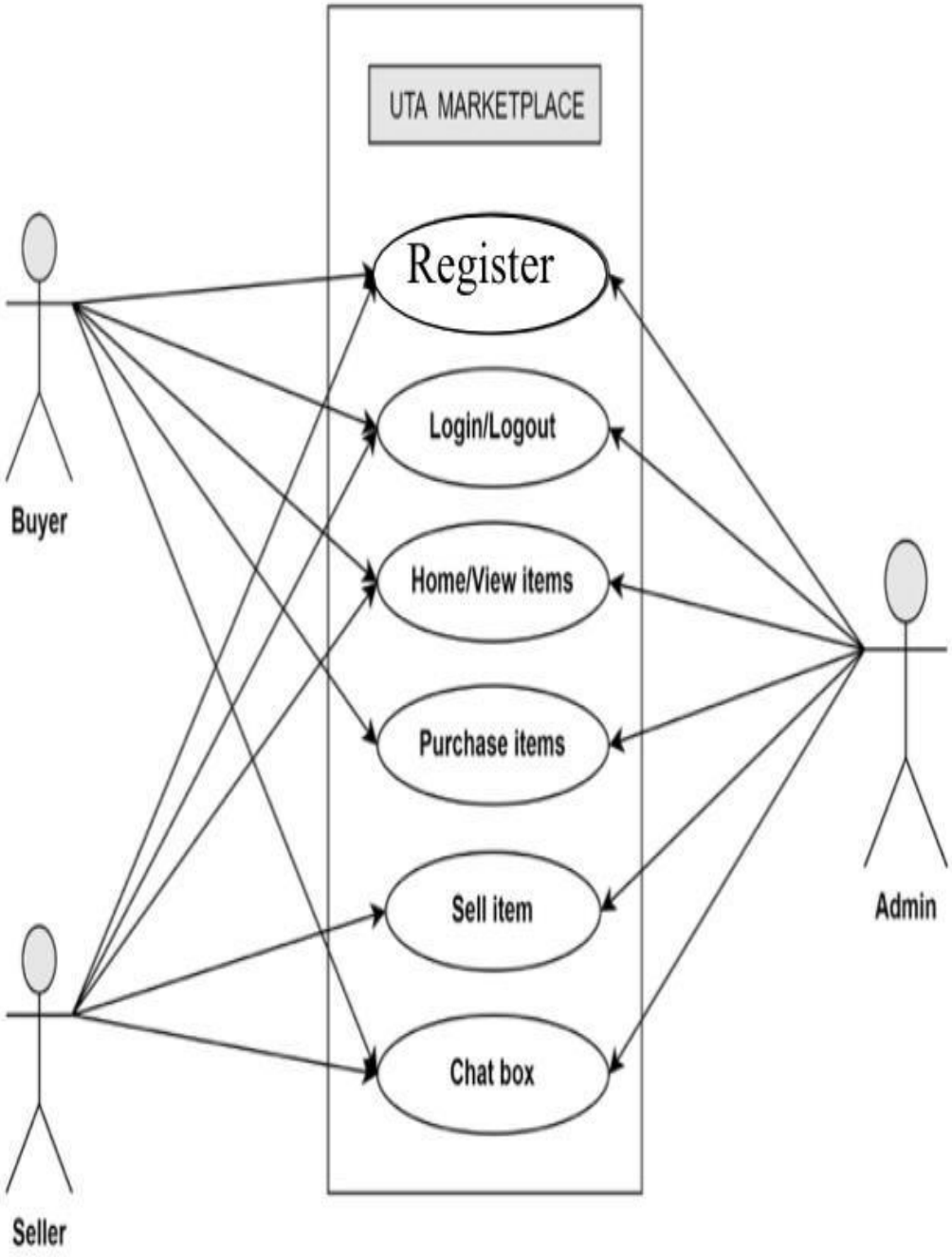
USE CASE

A Use Case Diagrams identify the functionality provided by the system, the users who interact with the system, and the mapping between the users and the functionality.

The main primary goal of Use Case diagrams include:

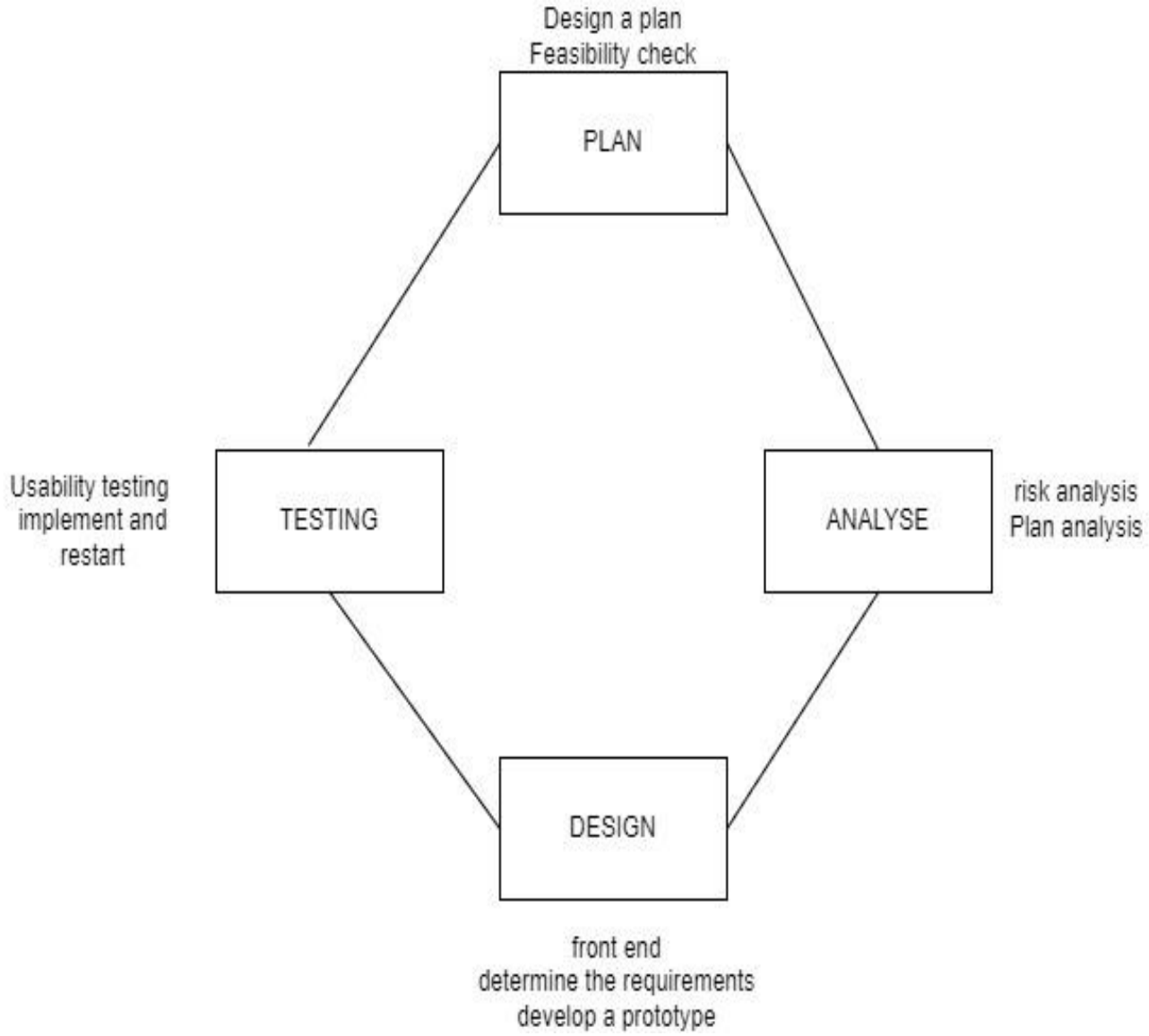
- Providing a high-level view of what the system does
- Identifying the users of the system
- Determining areas needing human-computer interfaces

High-Level Use Case Diagram (HLUC)



UCD (User-Centered Design)

UCD is an iterative design process in which designers focus on users and their needs at each stage of the design process.



RUCTM – Requirements Use Case Traceability Matrix

These are the requirements for our application. They have been categorized appropriately and are numbered for easy formation of RUCTM.
R1: Functional Requirement (Admin Module)
R2: Functional Requirement (Users Module)
R3: Non-Functional Requirement (Availability)
R4: Non-Functional Requirement (Accuracy)
R5: Non-Functional Requirement (Usability)
R6: Non-Functional Requirement (Security)
R7: Non-Functional Requirement (System Performance)
R8: Non-Functional Requirement (System Reliability)
R9: Interface Requirement (Hardware)
R10: Interface Requirement (Software)

Now, these are the Use Cases of our application. They have been numbered for simplicity and since this is a marketplace, these Use Cases are valid for both Buyers and Sellers.
UC1: Registration
UC2: Login / Logout
UC3: Purchase Item
UC4: Sell Item
UC5: Chat Box

	Priority Weight	**UC1**	**UC2**	**UC3**	**UC4**	**UC5**
R1	5	X	X		X	
R2	5	X	X	X		
R3	2	X	X	X	X	
R4	3	X	X			
R5	2	X				
R6	5	X	X	X	X	
R7	4					X
R8	3					X
R9	1	X	X			X
R10	1	X	X	X	X	X
Total		24	22	13	13	9

INCREMENT MATRIX - Project Planning with Use Cases

	Week 1	**Week 2**	**Week 3**	**Week 4**	**Week 5**	**Week 6**	**Week 7**	**Week 8**	**Week 9**	**Week 10**
UC1	X	X								
UC2			X	X	X					
UC3					X	X	X	X		
UC4					X	X	X	X	X	
UC5						X	X	X	X	X

References:

1. **Use Case diagram:** https://www.youtube.com/watch?v=zid-MVo7M-E
2. **Domain modeling:** Chapter 5 Object-oriented software engineering – David King
3. **Actor system Interaction Modeling:** Chapter 8 Object-oriented software engineering – David King
4. **Modeling and Design of Interactive Systems:** Chapter 7 Object-oriented software engineering – David King