# Design

## Introduction

Design is the prime steps in SDLC, which helps to transform the requirement into some appropriate form and moves the direction from problem sphere to solution territory.

Design is broad in itself and contains so many categories. Here I am going to take only an appropriate UML design, which is suitable for my project.

### Structural Diagrams

Structural diagram considers one of the important parts of the UML diagram. The main objective of this diagram is to represent the static part of the system.

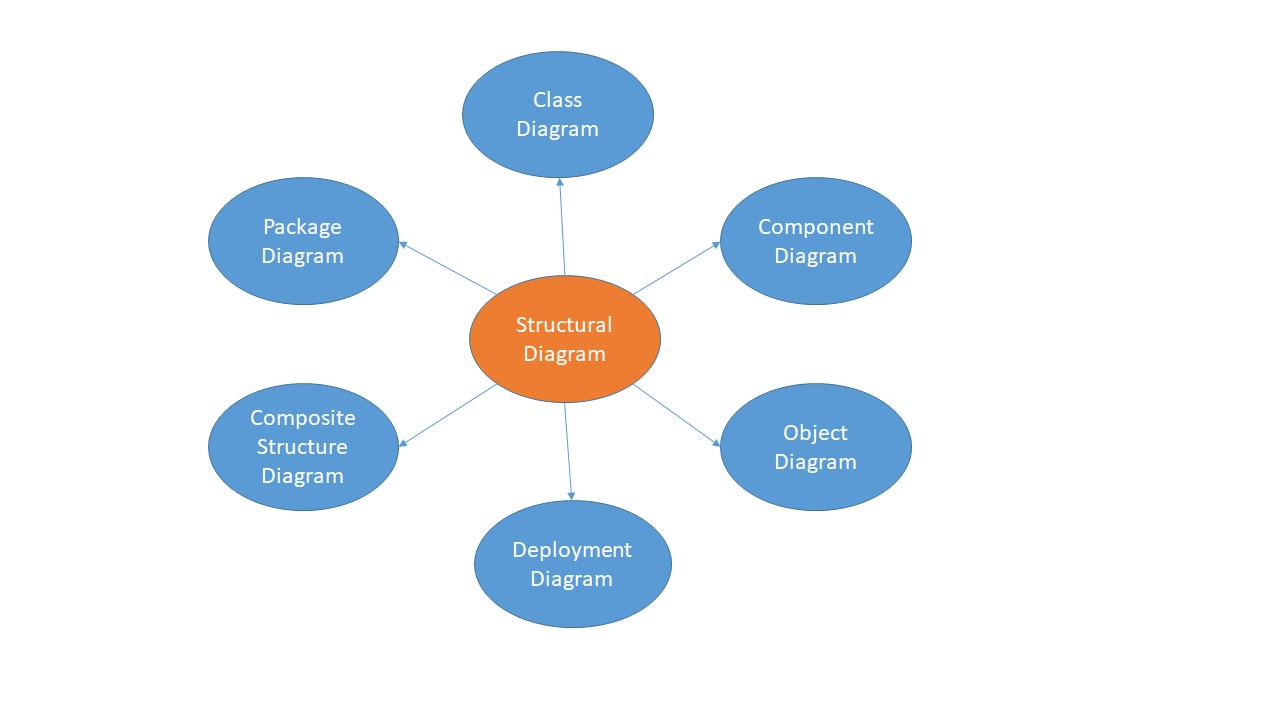


Figure Structural Diagram

#### Class Diagram

##### Introduction

For the development purpose, a class diagram is the central chunk of any object-oriented solution. The main objective of the class diagram is to show the classes in a system, attribute and their relationship.

##### Justification

* The best approach to model the static view of the system.
* It is the only model that can directly be mapped with the object-oriented language.
* Helps to perform both forward and reverse engineering.
* A fundamental base for the component.

##### Final Class Diagram

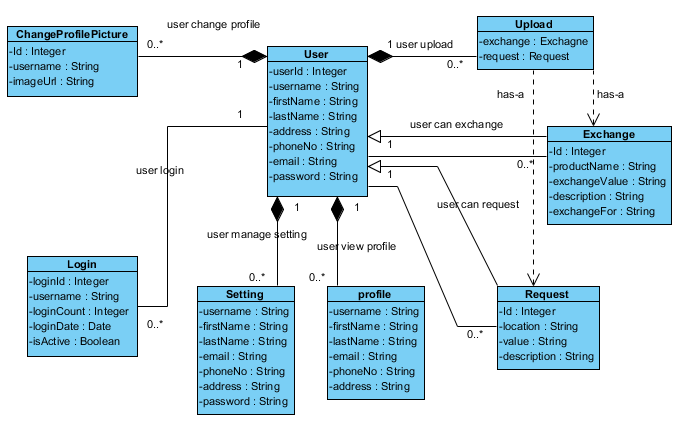


Figure 2 class diagram

## Notation Used

|  |  |  |
| --- | --- | --- |
| Name | Notation | Description |
| Association |  | In general, association connects the relationship between classes. It uses the verb phrase, which reflects the real world problem domain. The cardinality can be express as one-to-one, one-to-many and many-to-many. |
| Inheritance |  | Also known as generalization. Show the relationship between the subclass and the superclass means the subclass inherits the features of superclass. It is represented by the “is-a” relationship. |
| Realization |  | Generalization is the relationship between the blueprint class and the object accommodate its corresponding implementation level information. For e.g.:- the relationship between interface and implementing classes. |
| Dependency |  | A special type of association, which shows that an object of one class might use an object of another class. Any change in one class might reflect on the other. |
| Aggregation |  | A more generalized type of association, which represent the part of the relationship between the classes. For e.g.:- class2 is a part of class1 means object of class2 and class1 have separate lifetimes. |
| Composition |  | A more generalized type of aggregation in which object of class2 live and die with class1. None of the class can stand by itself. |

##### Data Flow Diagram and Description

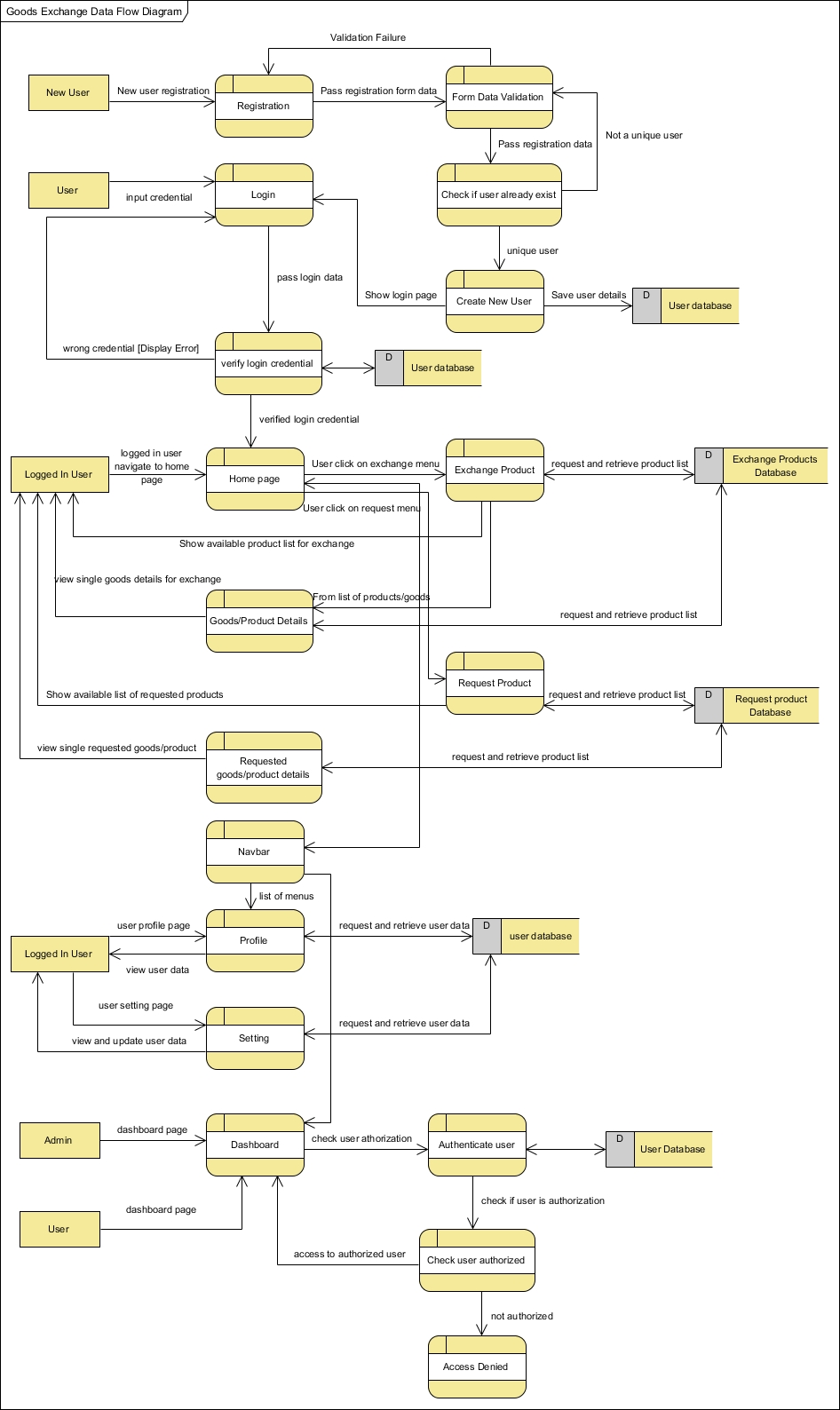


Figure Data Flow Diagram

##### Description of Diagram

Data Flow Diagram commonly known as DFD is used to visualize the flow of data in a business information system. It transfers data from the input to the storage during the system processing.

The above data flow diagram visualizes the flow of data of a proposed system. the above diagram has clearly shown how new user registration takes places and how existing user authenticates, authorize and login into the system. Beside this, a diagram also shows how a user can perform CRUD operation and exchange items or request items. Both user and admin role has been shown that describe how user and admin can perform their respective task.

##### Notation Used

|  |  |  |
| --- | --- | --- |
| Name | Representatoin | Description |
| Process |  | A process takes data as input, execute some steps and produce data as output. |
| External Entity |  | Objects outside the system being modeled and interacted with processes in a system. |
| Data store |  | Files or storage of data that store data input and output from the process. |
| data flow |  | The flow of data from process to process. |
| bidirectional data flow |  | The flow of data that flow both from and to process. |

Figure DFD Notation

## Behavioral Diagrams

The behavioral diagram is a dynamic aspect of the system, which helps to visualize the dynamic features of the system.

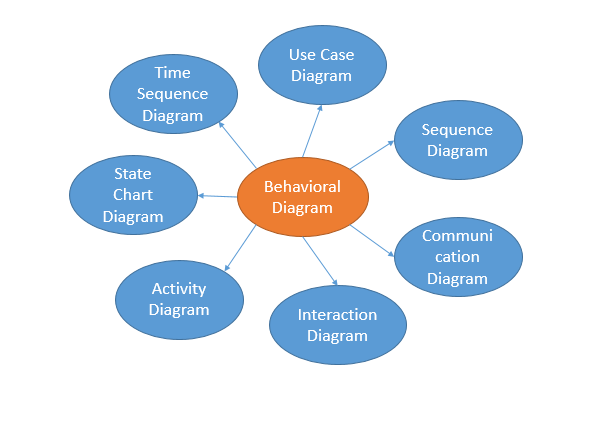


Figure Behavioral Diagram

### Activity Diagram

#### Introduction

Activity diagram considers one of the important aspects of the system in order to capture the dynamic behavior. Activity diagram is like a flow chart that shows the flow of one activity to another.

##### Justification

* It is used to draw the activity flow of a system.
* This diagram is a graphical representation of activities and shows the flow of message between the activities.
* The different components help to deal with all type of flow control and allow to think functionally.

##### Activity Diagram

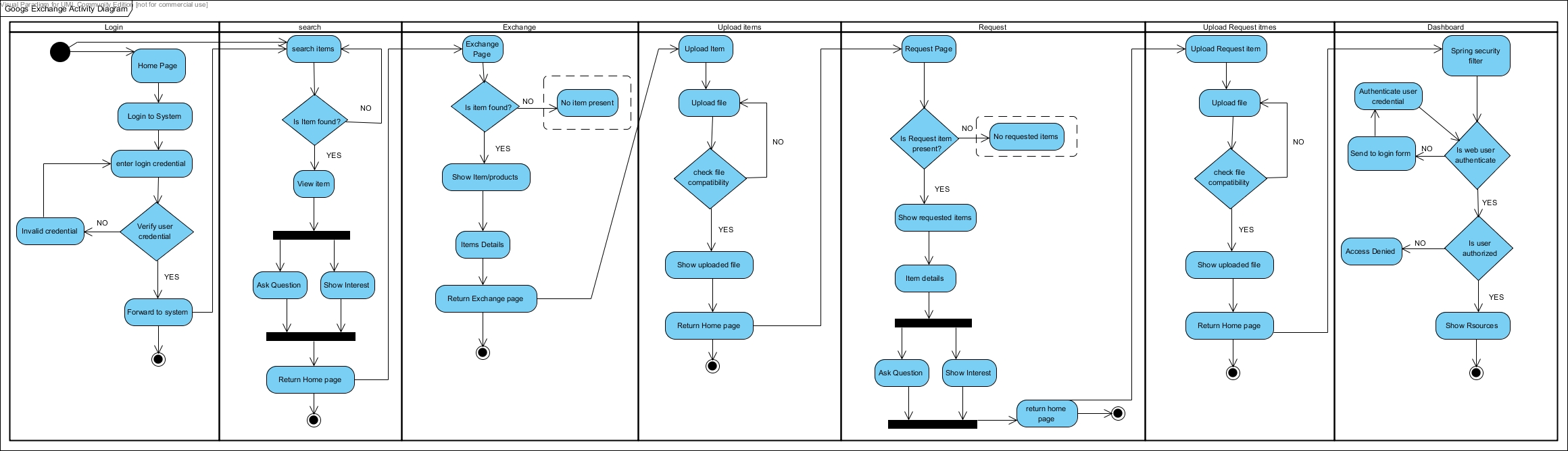


Figure 6 Activity Diagram

##### Description of Diagram

Activity diagram is an advanced version of flow chart it maps the flow of one activity to another activity. It is an important diagram of UML, which show the dynamic behavior of a system.

Here, in the above diagram a user initializes the process by login in into the system. The login process has several steps such as validate and check user credential. User can directly jump into the search process but in order to access the full functionality of a system, the user must be login, which is clearly shown in the diagram. After the successful login user can exchange a product, request product and upload items/products. To get access into the dashboard to manage the content of the system user must have authorization, which is clearly shown in above diagram.

## Notation Used

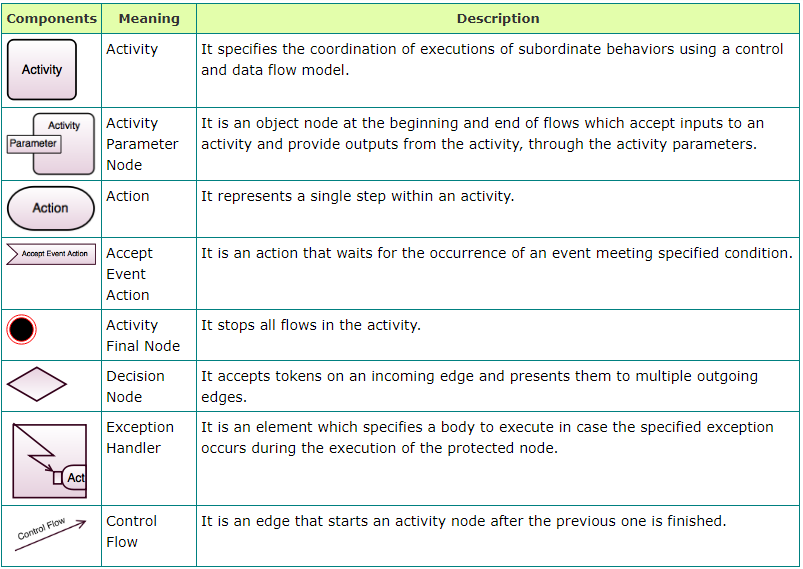


Figure 7 Activity Diagram Notation

#### Sequence Diagram

##### Introduction

A sequence diagram is a sequential interaction between objects. It provides an overview of how the object is function and in what order it function. This diagram helps to understand the requirement of the new or existing system.

##### Justification

* Sequence diagram help to visualize how objects are interacting in the system.
* The time sequence of the object can be captured.
* Sequence diagram show a message sequence chart.
* It consists of so many components that help to visualize how an overall system are interacting.

##### Sequence Diagram

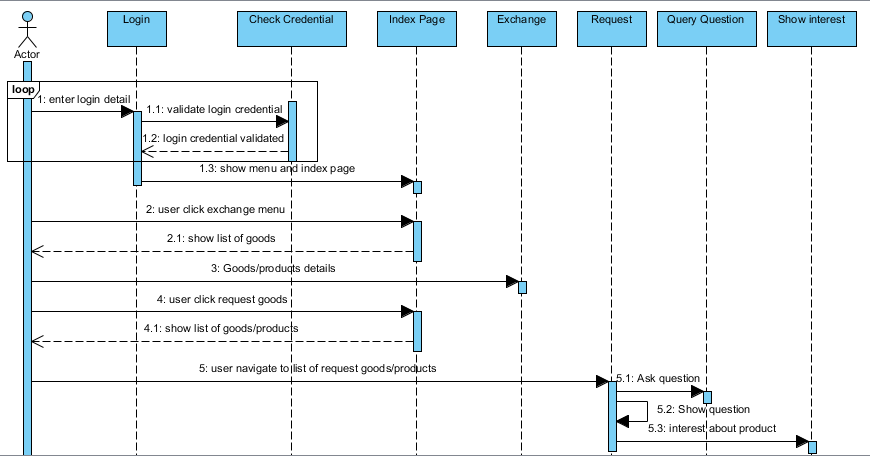


Figure 8 sequence diagram-1

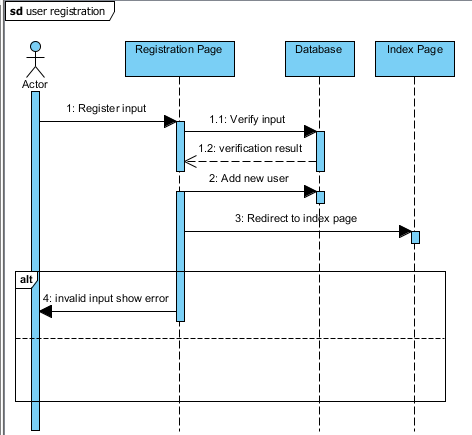


Figure 9 Sequence diagram-2

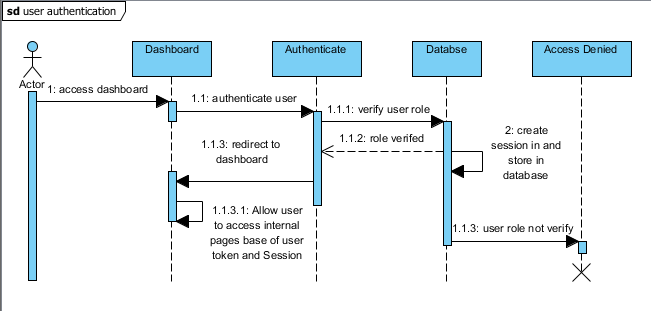


Figure 10 Sequence diagram-3

##### Description of Diagram

One of the important behavior diagrams is a sequence diagram, which shows the details of how the operation is carried out. It captures the interaction between the object in the context of a collaboration.

Here, in the first sequence diagram a user is represented by an actor and the whole sequence describe how user login and access the system full functions. At first, the user is shown a login form where user input the login credential and a credential is validated. If the login details are wrong then the login process is looped until user input the right credential. The looping process is surrounded by a loop frame. After a successful login user can perform a different function such as exchange, request products and ask a question. Similarly, in a second sequence diagram, a new user registration process has shown. User input required information into the fields and the field validated, if there is no error the user information will be stored in database. If there exist error that is validation error then the user in the registration process is not ended and a validation error populates which is surrounded by an alternative frame. Finally, in the third sequence diagram, a user authentication process is shown where an actor try to access into the system dashboard, which is only, authorize to admin. Here, in the authorization process, a user is authenticated by verifying user role from the database. If the user role is verify to get access into the dashboard then it will redirect to the dashboard by creating the user session and if user role is not verify to access to the dashboard then it will redirect to the access denied page or block the user to get access into the dashboard and terminate the process.

## Notation Used

|  |
| --- |
|  |

## Database Diagram

### Data Dictionary

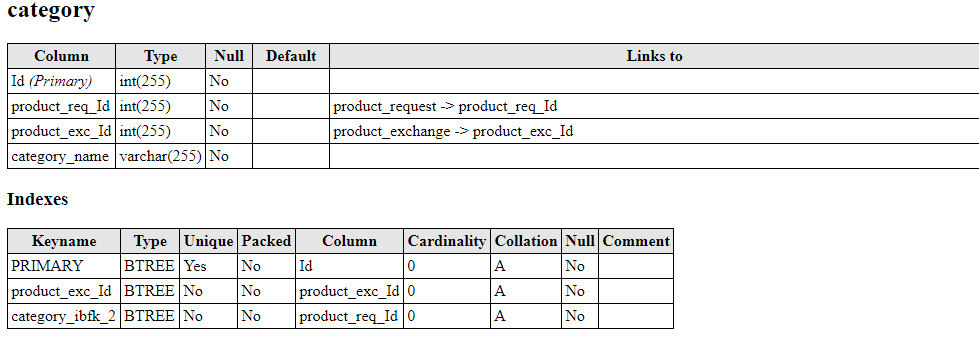


Figure 11 data dictionary table\_category

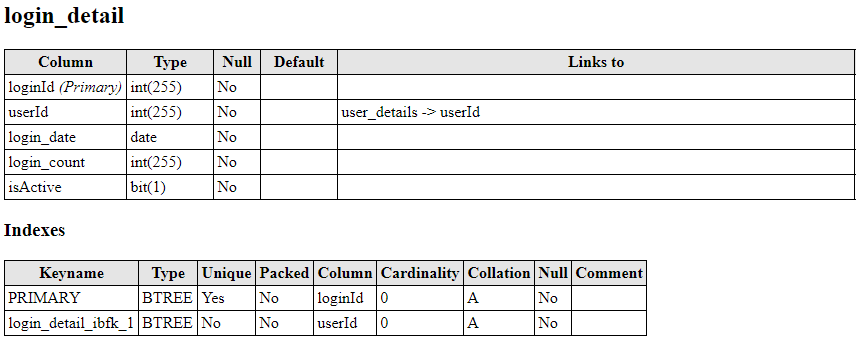


Figure 12 data dictionary table login\_detail

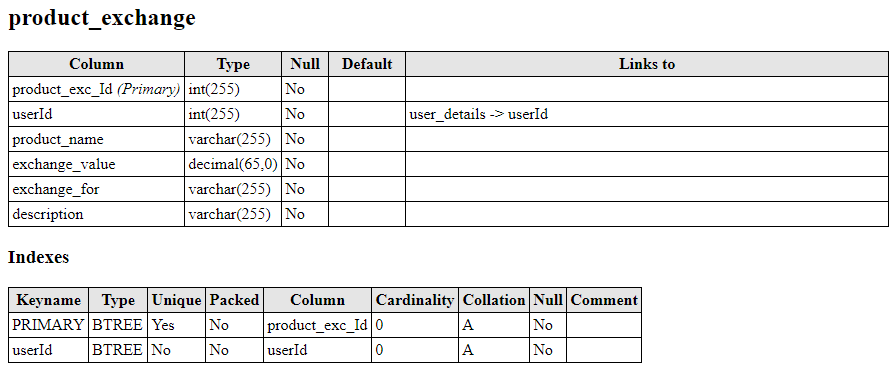


Figure 13 data dictionary table product\_exchange

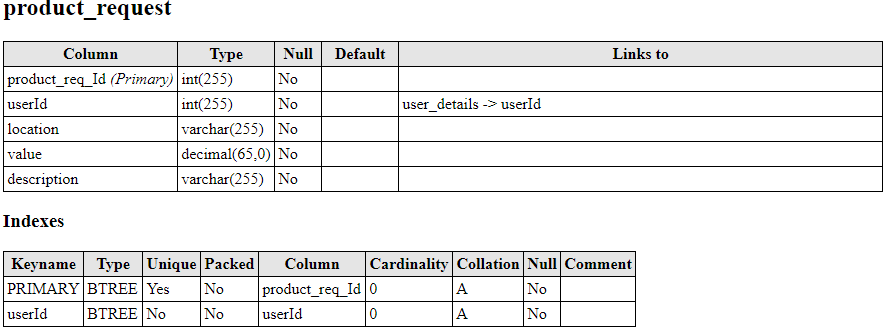


Figure 14 data dictionary table product\_request

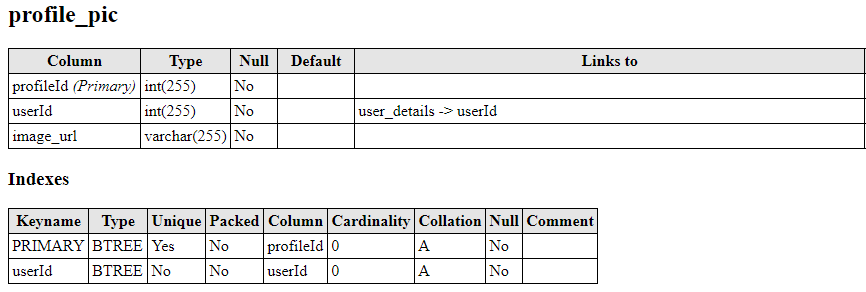


Figure 15 data dictionary table profile\_pic

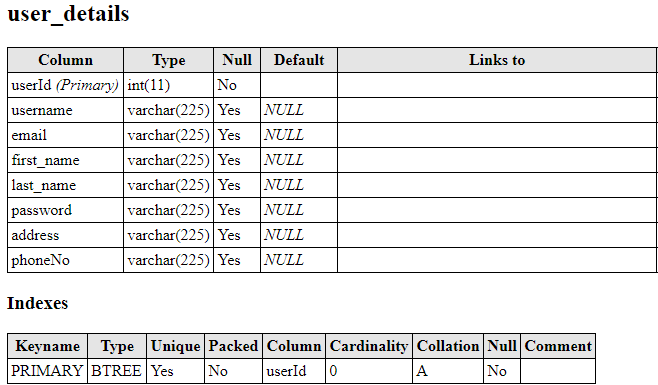


Figure 16 data dictionary user\_details

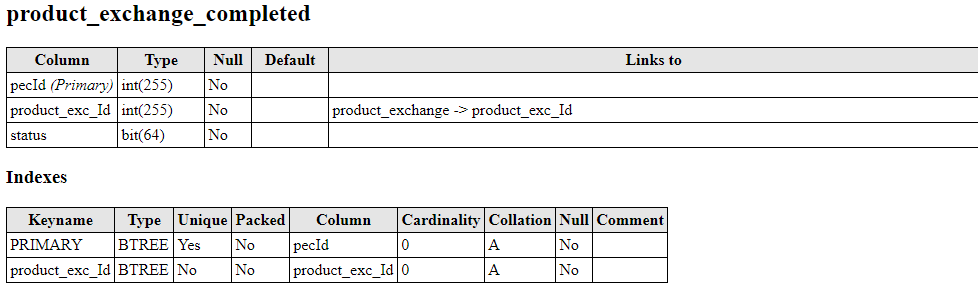


Figure 17 data dictionary product exchange completed

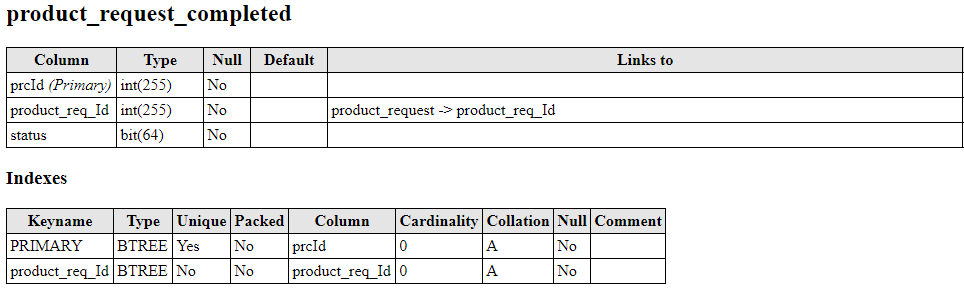


Figure 18 data dictionary product request completed

### ER-Diagram

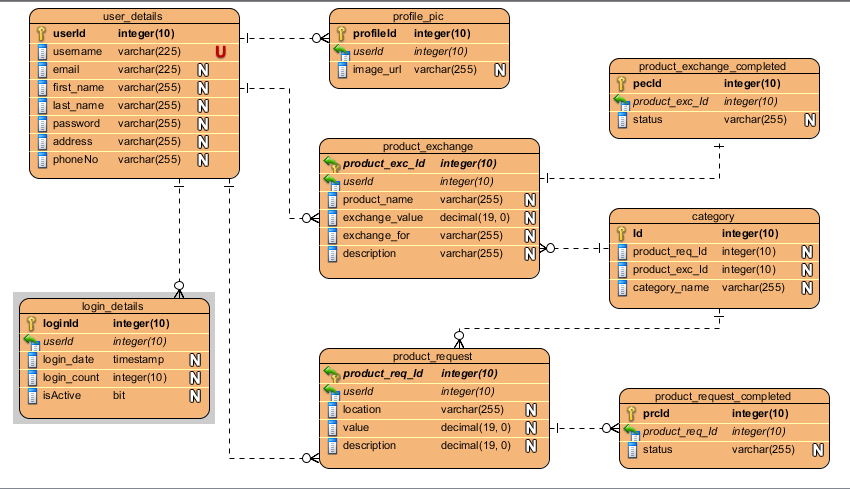


Figure 19 ER-Diagram

## Architectural Model

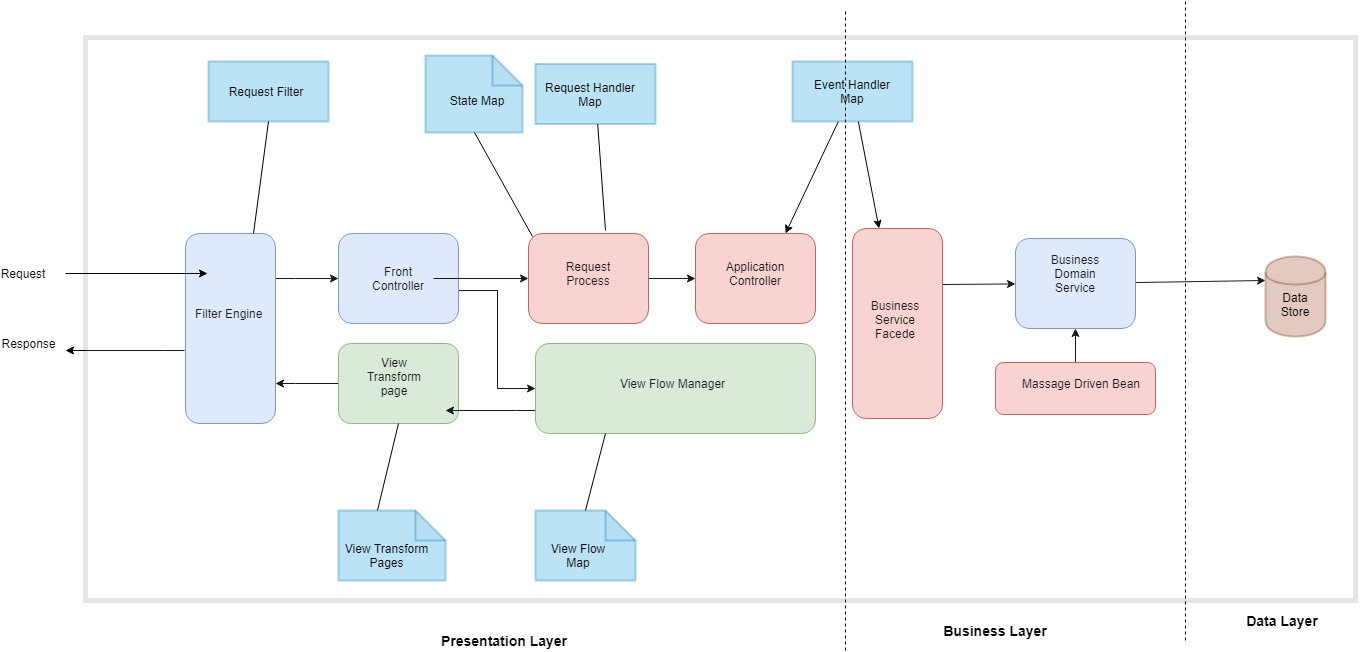


Figure 20 Architecture Diagram

## Description

Three-tier architecture is a three-way communication of client and server environment. One of the most popular architectures, which is widely used in web application. It provides a production-ready look for our application. The architecture is divided into three layers, which are presentation layer, business layer, and the data layer. The above architecture is an example of three-tier architecture used in java spring project and the diagram is described based on the spring project.

The very top-most layer is the presentation layer where user interaction with the application takes place. This layer contains classes and components required to represent the user interface. Mainly it consists of JSP/Servlet.

The middle layer is commonly known as the business layer where all the business logic is implemented. This layer coordinates the application, process commands makes logical decision and evaluations and perform the calculation. It also moves and process data between the two surrounding layers. This layer consists of Spring Services, Spring Transaction.

The final layer is the data layer where information is store and retrieves from the database. This layer consist of classes and components require to handle database related operation. Mainly it consists of repository classes, data access object class and other data access logic.

## UI Model

### Prototype

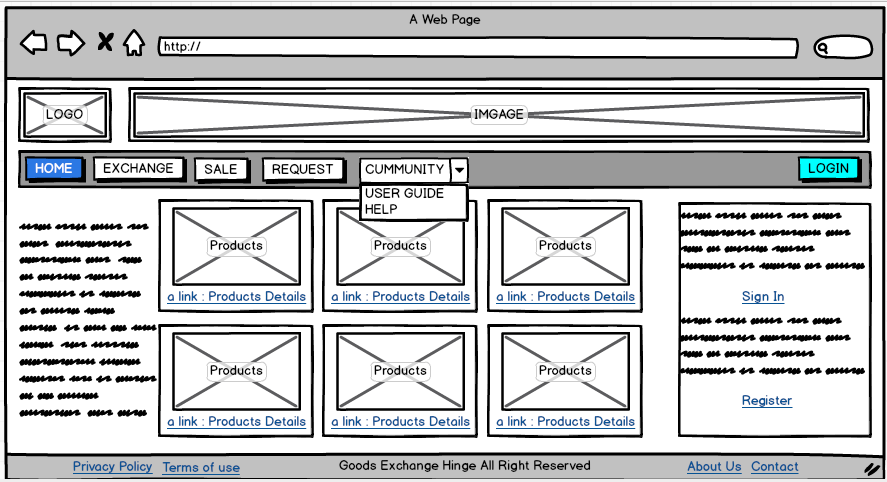


Figure home page

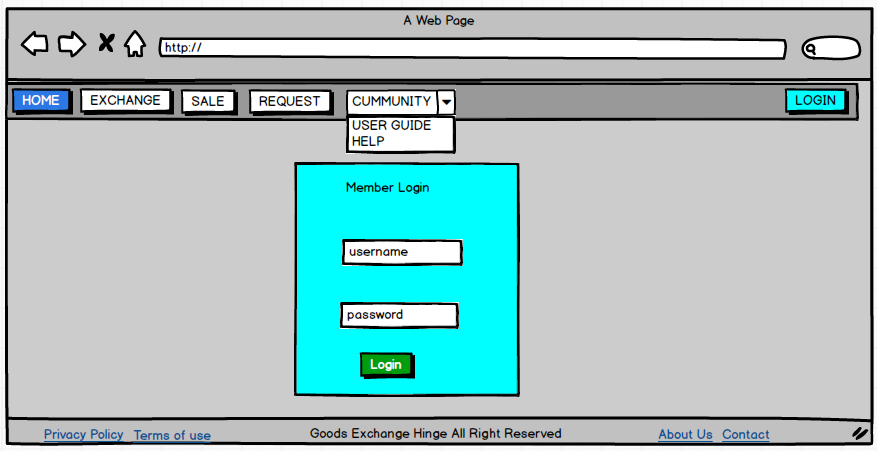


Figure login page

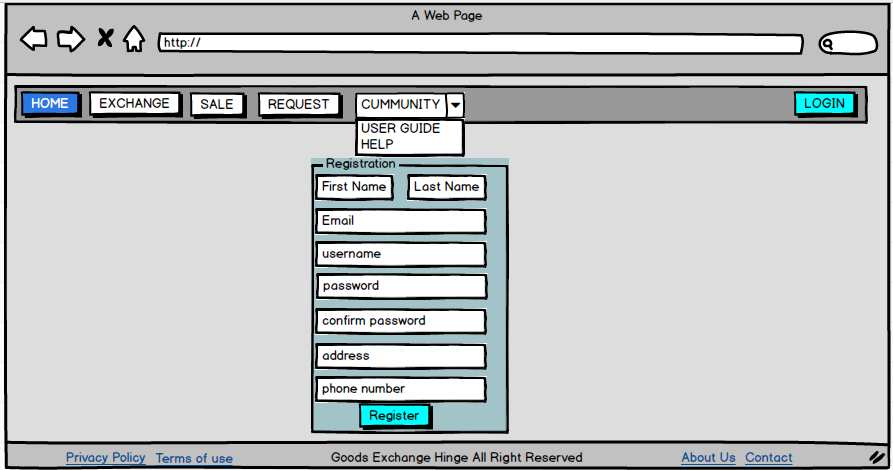


Figure registration page

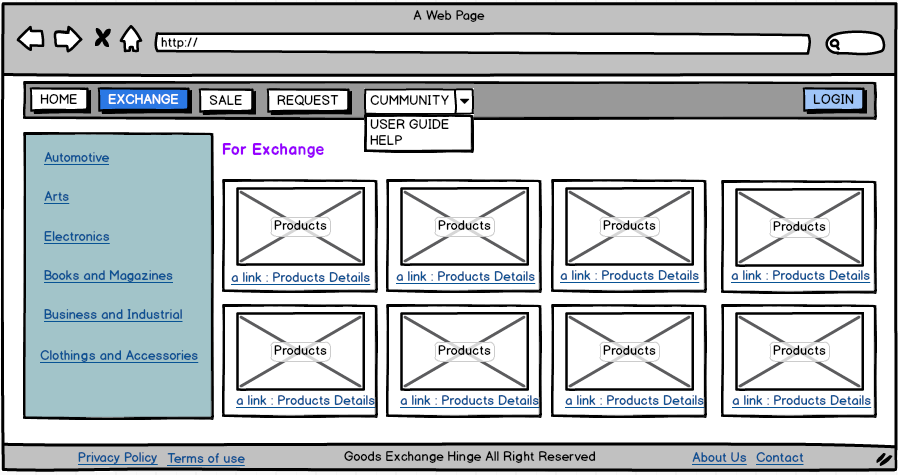


Figure exchange page

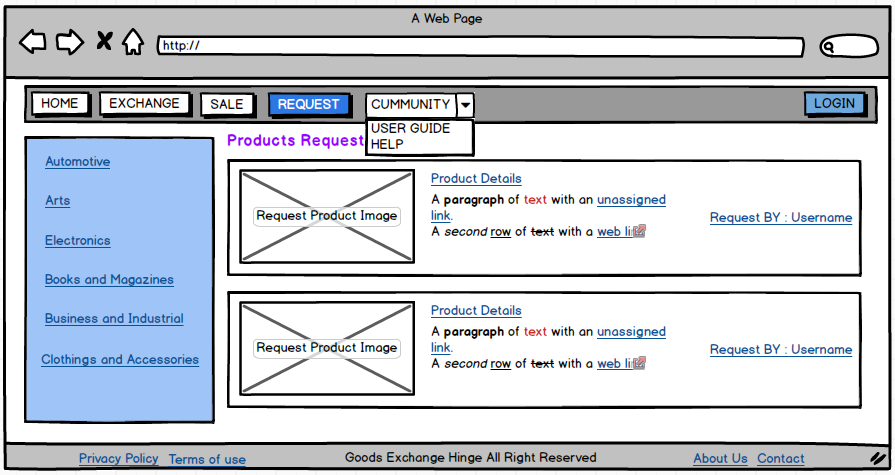


Figure request page

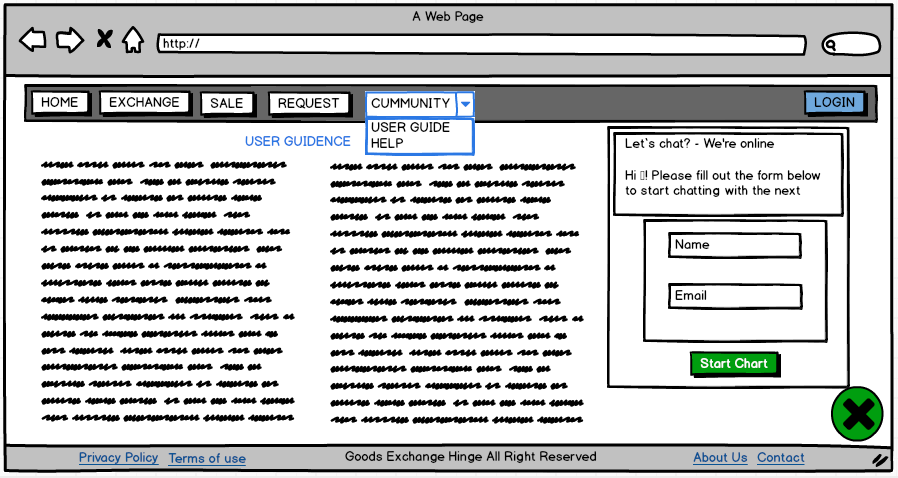


Figure community page

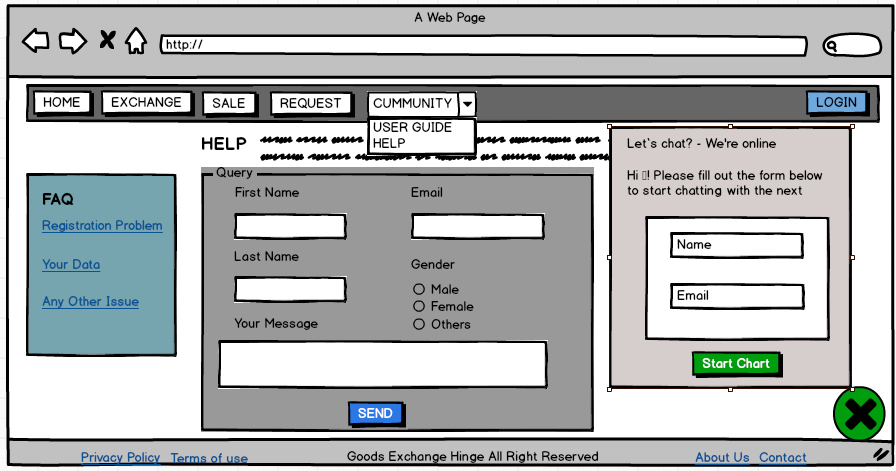


Figure help and chart page

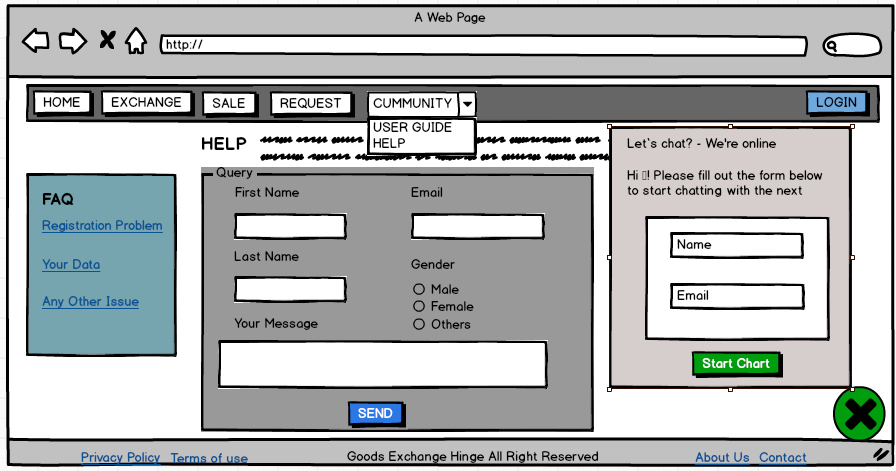


Figure dashboard page