

# **INVENTORY MANAGEMENT APPLICATION**

**A Thesis Submitted  
In Partial Fulfillment of the Requirements  
for the Degree of**

## **MASTER OF COMPUTER APPLICATIONS**

**by**

**Surbhi Mittal  
(1900290149098)**

**Under the Supervision of  
Mr. Ankit Verma**

**(Assistant Professor)**

**KIET Group of Institutions Ghaziabad**



**to the**

**Faculty of Computer Application  
DR. APJ ABDUL KALAM TECHNICAL  
UNIVERSITY LUCKNOW  
(Formerly Uttar Pradesh Technical University)**

**JULY, 2021**

## **DECLARATION**

I hereby declare that the work presented in this report entitled “Inventory Management app with Salesforce”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

**Surbhi Mittal**

**190029019098**

**MCA**



**(Candidate Signature)**

## **CERTIFICATE**

Certified that **Surbhi Mittal** (1900290149098) have carried out the project work having **“Inventory Management app with Salesforce”** for **Master of Computer Applications** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Surbhi Mittal**

**Roll No. 1900290149098**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Signature of Internal Examiner**

**Signature of External Examiner**

**Dr. Ajay Kr. Shrivastava**  
**Head, Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

## **Abstract**

Salesforce is a cloud-based customer relationship management (CRM) software that accelerates business relationships and can transform the working lives of the team. Marc Benioff developed it in the late 1990s and now it has been announced as the world's most innovative company for six consecutive years by Forbes Magazine. Unlike traditional CRM software, Salesforce is an internet service. It is available with just a sign-up and logs in through a browser, and it is immediately available. It is based on cloud computing, where the customers, without the need of installing any traditional software, can access the cloud, i.e., through the internet, for their business needs.

Inventory Management (IM) is the method of controlling and supervising the storage, utilization and ordering of components that an organization can track of their items it sells. It is the act of controlling and administering the quantities of products in the sale. For a business, an inventory is the main asset which represents an investment by the owner until the item is sold .

To demonstrate its functionalities of Salesforce, I created an application for inventory management. Here, In the inventory management, there are two parts: one is administration part, and another part is customer portal. The administrator manages the inventory and store operations, and the customer buys the products in the inventory through customer portal.

## Table of Contents

Declaration	ii
Certificate	iii
Abstract	iv
Table of Content	v
List of Figures	vi
<b>Chapter 1: Introduction</b>	<b>1-45</b>
1.1 Introduction to Cloud Computing	
1.2 Service Models of Cloud Computing	
1.3 Salesforce Basic	10
1.3.1 Introduction to Salesforce	10
1.3.2 The architecture of Salesforce	10
1.3.3 Salesforce Multiple View Controller	11
1.3.4 Salesforce Versions	12
1.3.5 System Overview	13
1.3.6 Technologies of Salesforce	14
1.3.7 Benefits of Using Salesforce	15
1.4 Inventory Management	<b>16</b>
1.4.1 Inventory Management	17
1.4.2 Development Area	17
1.4.3 Custom Apps	18
1.4.4 Objects	19
1.4.5 Fields	19
1.4.6 Relationships	20
1.4.7 Tabs	22
1.4.8 Securing and Sharing Data	23
1.4.9 Page Layouts	23
1.4.10 Visual Force	24
1.4.11 Apex	24
1.4.13 Reports and Dashboards	25
1.5 Functionalities	<b>26</b>
1.5.1 Inventory Administration	26
1.5.2 Customer Portal	35

## **Chapter: 2 Literature Review**

2.1 Abstract.

2.2 Introduction.

2.3 Related Work.

2.4 Design (GUI, analysis, implementation, evaluation).

2.5 Results and Decisions.

2.6 Conclusion.

2.7 References.

## List of Figures

Figure 1: Cloud Computing .....	7
Figure 2: Service Models of Cloud Computing .....	9
Figure 3: Service Models .....	9
Figure 4: Multi-tenancy.....	10
Figure 5: Salesforce MVC .....	11
Figure 6: Different Versions of Salesforce .....	12
Figure 7: Salesforce Mobile App .....	13
Figure 8: Benefits of Salesforce .....	16
Figure 9: Inventory Management System .....	17
Figure 11: An example of a Master-Detail relationship.....	21
Figure 12: Relationships between objects .....	22
Figure 13: Schema of Inventory Administrator .....	26
Figure 14: Product Approval Process .....	32
Figure 15: Workflow of Customer Portal .....	35

# CHAPTER 1

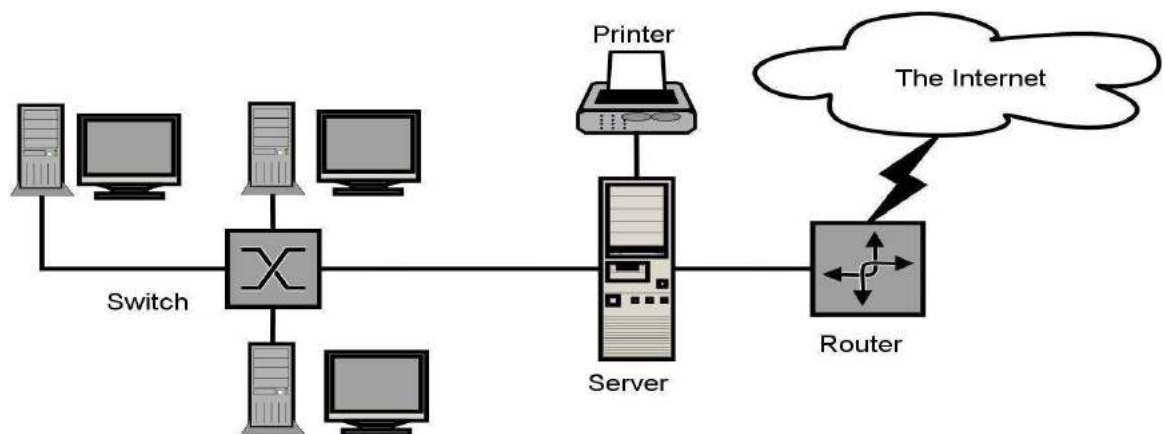
## INTRODUCTION

Cloud computing is usually defined as storing and managing the data over the cloud, rather than a local server. Cloud computing is easy to understand. All applications are developed and run in the web browser. Using the internet connection, users and developers will have access to whole applications thus eliminating the complexity and overhead of the maintain environment.

Unlike traditional business applications which are complicated, expensive and need experts to install, run, update and secure, cloud computing can be accessed anywhere with an online connection. In traditional systems, the entire infrastructure must work together. For such type of seamless interaction, and for the smooth run of the system, a constant maintenance is always required. With cloud computing, there is no necessity to invest money in acquiring and supporting hardware and software infrastructure, thus decreasing the potential cost for users and developers.

The main impact of cloud computing is on the responsiveness of IT systems. With the cloud computing environment, we can add users and developers instantly, and the applications can be deployed rapidly into

the cloud which reduces the user request response time. As the complexity of the of the internal systems is removed, the organization can speed up the entire IT process.



**Figure 1: Cloud Computing**



## **1.2 Service Models of Cloud Computing**

### **Software as a Service (SaaS)**

The end user can access the application which is developed by the provider on a cloud framework. The developed applications are available from different customer devices through interfaces like a web program or a program interface. Cloud infrastructure, servers, networks, storage and operating systems cannot be managed or controlled by the customer.

Cloud application services represent the increasing cloud market. Software as a service utilizes the internet to deploy the applications overseen by the vendor and whose interface will be able to access on the customer side. Many of the applications developed using SaaS will run in a web browser by using some plugins. There will not be need of any download or establishments.

The major applications which are developed using SaaS are healthcare related applications, client relationship administrations, incorporate email, and collaboration. Some of the costly ventures which are not able to considered as software vendors started using SaaS to get the upper hand and gain income.

### **Platform as a Service (PaaS)**

The Client can deploy onto the cloud infrastructure developed by the customers with the help of libraries, tools, services and the programming languages which are supported by the client. The underlying infrastructure of cloud and storage, servers, network or operating systems cannot be managed or controlled by the Customer.

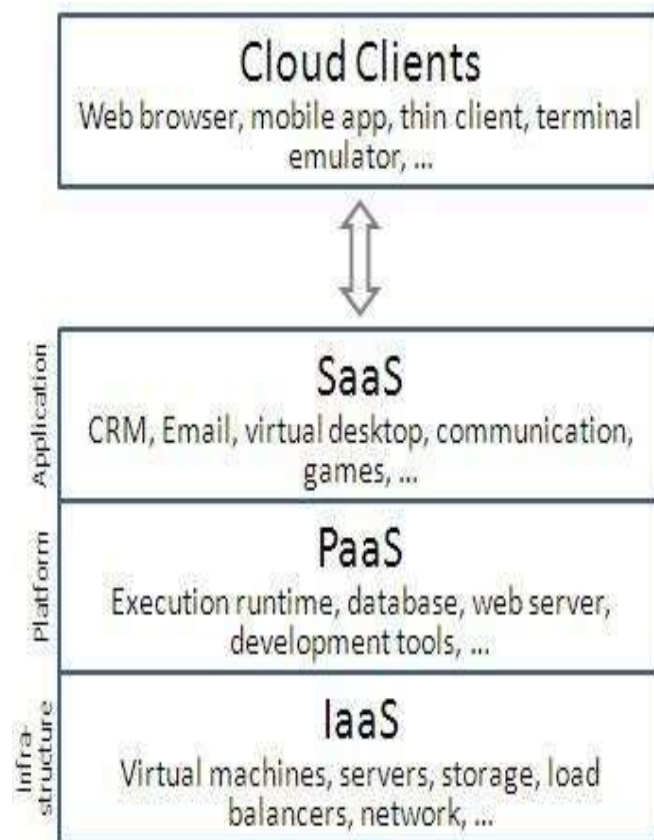
Cloud platform services can be utilized for applications and their advancements when cloud segments are given to programming. Developers can be able to pick the structure using PaaS where the applications can be expanded to create or modify. The testing and deployment of applications become easy and fast if the PaaS is used .

Enterprise PaaS gives a self-service portal to programming engineers for overseeing computing infrastructure from information technology operations. Scalability, Software as a service enablement and multi-occupancy can be acquired by the applications using PaaS. The coding fundamental measure will be decreased for the enterprises using PaaS and the application will be converted to a hybrid model.

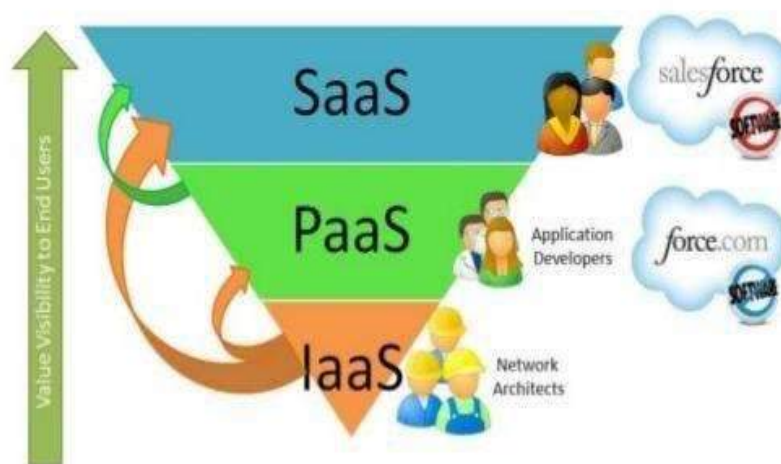
### **Infrastructure as a Service (IaaS)**

The customer can arrange systems, storage, processing and other essential computing resources in which the client can run and send arbitrary software like applications and operating systems. The hidden cloud infrastructure is not controlled or overseen by the customer but rather he can control over operating systems, storage, servers and network.

Cloud infrastructure services, referred to as IaaS, are models beneficial for overseeing and observing remote data center frameworks such as organizing, processing, stockpiling and networking services. With the help of IaaS, clients will get utility billing and power benefits.



**Figure 2: Service Models of Cloud Computing**



**Figure 3: Service Models**

## 1.3 Salesforce Basic.

### 1.3.1 Introduction to Salesforce.

Salesforce is one of the world's prime cloud computing companies and number one on-demand customer relationship management (CRM). Salesforce does not need any software installation or hardware or any infrastructure like servers. All we need to access Salesforce is the internet. This empowers even the most non-techie individuals to be able to use the system and configure it as per their needs.

Established as Salesforce.com 2(SFDC) and its customer relationship management (CRM) service and then divided into different sectors like sales cloud, service cloud, community cloud, analytics cloud, data cloud, marketing cloud, app cloud, and so on.

Since Salesforce coordinates well with all the platforms and supports all major OS and mobile devices, it is anything but difficult to utilize Salesforce outside of the workplace, thus helps to improve productivity.

### 1.3.2 The Architecture of Salesforce

Salesforce has a multi-tenant Architecture. Multi-tenancy is the fundamental technology utilized as a part of the cloud to share its resources safely and cost effectively. It's much the same as bank services where various tenants cost-efficiently share a common infrastructure yet safely and with most protection from other tenants. A cloud utilizes multi-tenant infrastructure to share its assets safely among different applications and occupants (organizations, associations, and so on) that use the cloud. Some clouds utilize virtualization-based architecture to confine occupants; others utilize custom software architecture to take care of business. The multi-tenant outline of a cloud service can dramatically affect the application delivery and the profitability of IT organization.

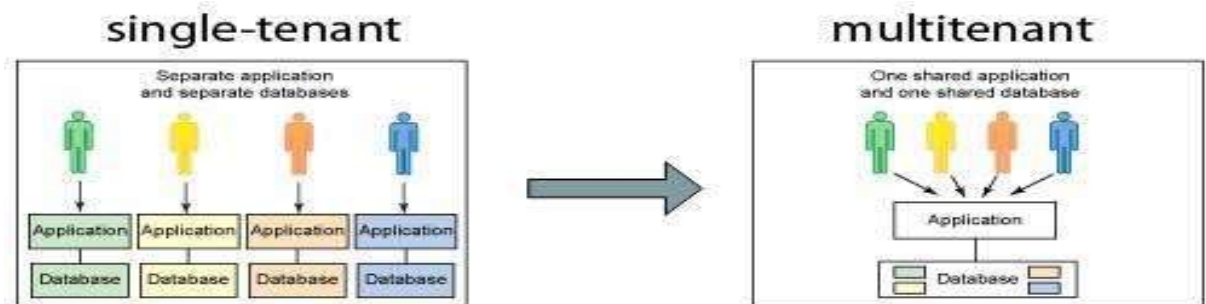


Figure 4: Multi-Tenancy

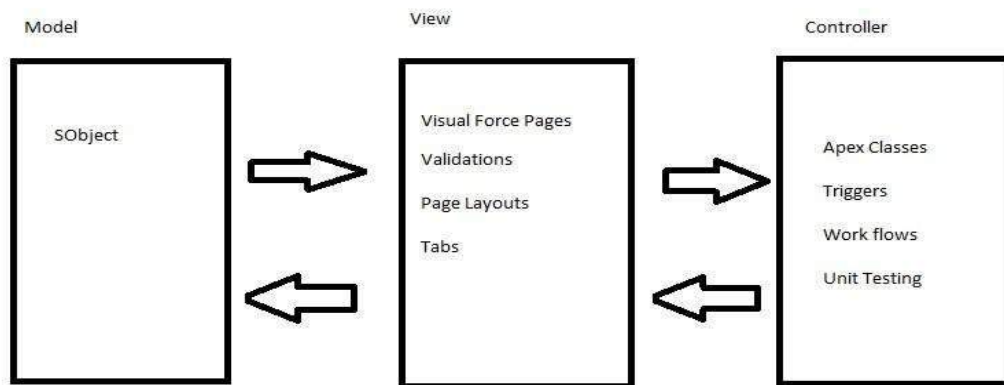
### 1.3.3 Salesforce Multiple View Controller (MVC)

MVC is a design pattern which separates business logic from interface logic i.e. it separates the graphical interface displayed to the user with the code that manages the user action.

In Salesforce, using SFDC visual force, we can write VIEW pages which are very similar to java servlets page(JSP) pages. Each visualforce(VF) page is correlated with a controller. The controller and model classes can be written using Apex language. In SFDC, controller part comprises of workflows, triggers, Apex classes and model layer comprises of fields, relationships, objects and View layer comprises of Tabs, page layouts, VF pages.

SFDC MVC mainly consists of three modules namely Model, View and Controller.

- 1) Model: Here we represent what schema and data the Salesforce used for the system representation and Subjects are a model, as every entity is mapped to some subject in Salesforce.
- 2) View: Here we represent how data and schema and visual force are used to present data to users.
- 3) Controller: Here we use controllers and interface actions to perform actions when the user interacts with visual force



**Figure 5: Salesforce MVC**

### **1.3.4 Different editions of Salesforce.**

Salesforce offers different editions of its products and services depending on business needs.

1)Professional Edition: This edition offers full CRM functionality.

2)Enterprise Edition: This edition offers advanced CRM customization and administration tools along with functionalities of Professional edition.

3)Unlimited Edition: Along with functionalities offered by Enterprise Edition, Unlimited Edition even offers full mobile access, premier support, unlimited custom apps and more.

4)Developer Edition: This edition allows developers to extend the Salesforce system, integration with other applications and develop new applications and tools.

SalesforceIQ Starter	Professional	Enterprise	Unlimited
Out-of-the-box CRM for up to 5 users	Complete CRM for any size team	<b>MOST POPULAR</b> Deeply customizable CRM for your business	Unlimited CRM power and support
<b>TRY FOR FREE</b>	<b>TRY FOR FREE</b>	<b>TRY FOR FREE</b>	<b>TRY FOR FREE</b>
Automatic data capture Customizable sales tracking for one list Intelligent follow-up reminders Contact auto-complete Shared address book Sent-email notifications Seamless collaboration Smart mobile apps Chrome extension Webinars and live trainings	Account and contact management Opportunity tracking Lead management Task and event tracking Customizable reports and dashboards Mobile access and administration Chatter – company social network Outlook Side Panel and sync Role permissions Case management Campaigns** Quotes and orders** Collaborative forecasts** Mass email** Sales Data*** Sales Cloud Engage***	Get all Professional features PLUS Workflow automation Enterprise territory management Profiles and page layouts Custom app development Integration via web service API Salesforce Identity Salesforce Private AppExchange Report history tracking** Approval automation** Sales Data*** Sales Cloud Engage***	Get all Enterprise features PLUS Unlimited customizations Unlimited custom apps Multiple sandboxes Additional data storage 24/7 toll-free support Access to 100+ admin services Unlimited online training Sales Data*** Sales Cloud Engage***

**Figure 6: Different Versions of Salesforce**

### 1.3.5 System Overview

Salesforce.com allows administrators to configure and design systems for complex implementations. Salesforce combines the power of configuration and custom development in its platform i.e. Force.com Platform. This platform user can make use of custom code, workflows rules, approval processes to implement their business logic and they can integrate the data with other applications, generate reports and do the analytics within no time. The Salesforce CRM model is used in organizations for interactions like emails, meetings, events with customers and also for prospects like sales, marketing, and support.

With Force.com, we can run business in the mobile using the Salesforce1 app. We can build and optimize the apps for mobile using HTML 5 and UI framework and it supports all devices with just one code base. Salesforce1 downloadable app can be installed from the App store or Google play on a mobile device.

The Salesforce1 Mobile App Includes Menu Items, Fields, and Links

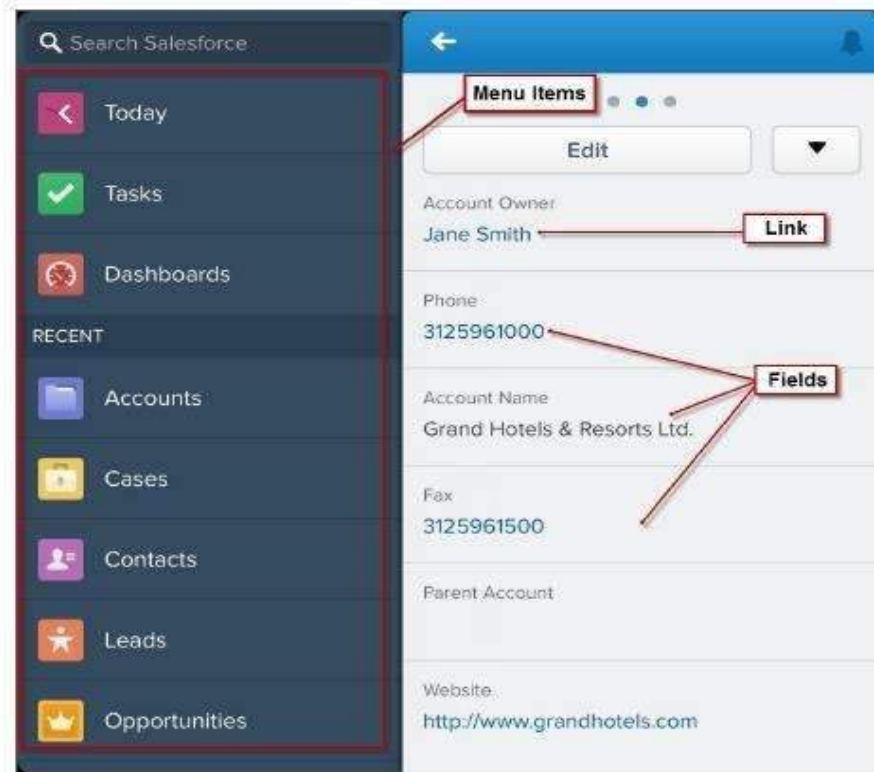


Figure 7: Salesforce Mobile App 1

Salesforce even has an app marketplace called AppExchange where we can find pre-built business applications. AppExchange offers thousands of verified and secured apps built by others or we can develop our own app and sell it here. It's quite similar to the App store and the play store.

Due to its ease of access, ease of use, minimum licensing/proprietary issues, and per user cost, Salesforce becomes a power system from small to large scale industries.

### 1.3.6 Technologies of Salesforce

#### 1) Apex

Salesforce has a programming language called Apex. It is a case-insensitive, mostly typed object-oriented programming language with syntax identical Java with curly brackets and dotnotation syntax. Apex is used to run programs and procedure in Force.com such as links, buttons, record insertion and so on with visual force custom controllers.



## 2) Visualforce

Visualforce(VF) is a framework for the Force.com platform with tag-based markup language identical to HTML. With the help of Visualforce, custom pages can be created for mobile apps and desktops with the help of with other front-end technologies like HTML, CSS, jquery, and JavaScript. With the Visualforce standard and custom controller features, we can build our own business logic in Apex.

## 3) Lightning

Lightning is a component-based framework for the Salesforce1 mobile app which is built on an open source Aura framework. With the lightning framework, responsive applications can be built easily. The apps build on the Lightning framework is sold or brought on AppExchange.

Lightning App builder for Salesforce is a tool for quick application advancement of responsive web interfaces. This interface takes into account distinctive screens to be assembled given lightning segments. This can be layouts as formats for records or particular applications .

## 1.3.7 Benefits of Using Salesforce

### 1) Invest in innovation, not infrastructure

With Salesforce, we can focus on business rather than the back end as Salesforce add new features and automatic upgrades three times a year. These boundary-pushing work made Salesforce one of Forbes' Most Innovative Companies.

### 2) Don't stress about the data security

		
<b>Application Services</b>	<b>Network Services</b>	<b>Infrastructure Services</b>
Manage password policies, app and data access, and field-level security.	Control access with built-in HTTPS, authentication, and network security controls.	Run apps securely, with automatic scalability, backup, and recovery.



### 3) Make Salesforce work the way we want

Salesforce can be customized to the core to be more agile and productive and the apps can be developed with interfaces with point and click to high-end platforms.

### 4) With AppExchange, find Prebuilt applications in minutes

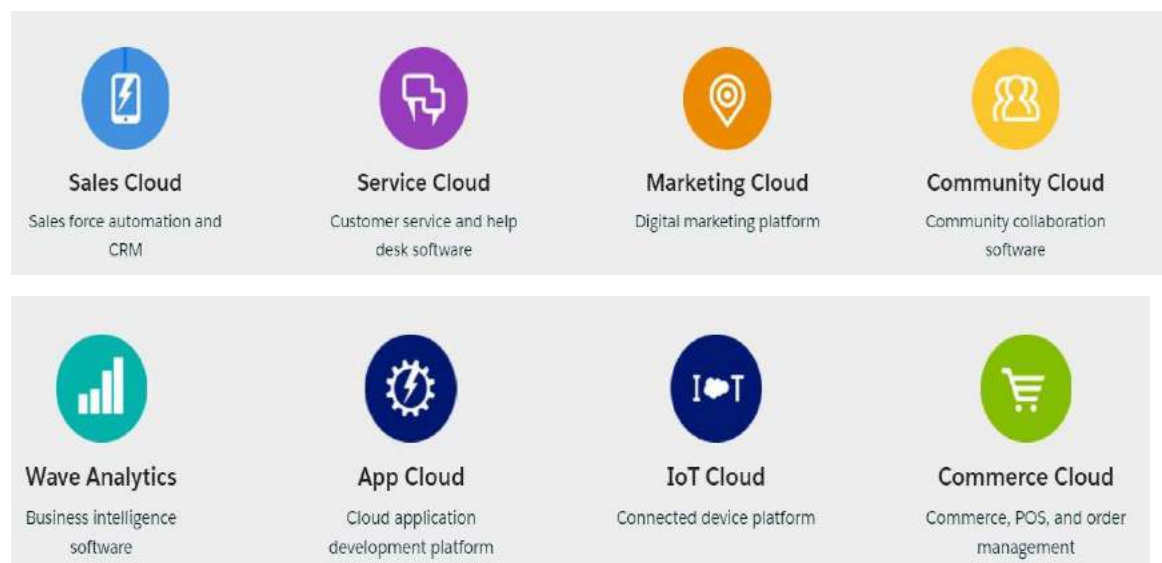
With industry-specific third party apps, Salesforce can be spread to every division and corporation. These apps are installed by millions of people and are reviewed by thousands. So with trust on the apps, we can focus on extending the business.

### 5) Work on one platform where everything works together

With Salesforce APIs, core technologies and third party tools, we can connect and manage data from any system and from anywhere.

Salesforce works in Desktop, Mobile Devices, and iPad too

Salesforce even have:



**Figure 8: Benefits of Salesforce**

## 1.4 Inventory Management

### 1.4.1 The Inventory Management

Inventory Management is the process of arranging, storing and ordering the items in the inventory. It is also the process of controlling and observing the products that are finished and available for sale. For a business, an inventory is the main asset which represents an investment by the owner until the item is sold.

Inventory Management helps us to provide production of an item, its sales and provides a service level for the customer at low cost. The largest item for some of the companies is their inventories. If there is any problem with the inventory, then it will lead to the failure of the business. So we have to manage the items in the inventory very carefully.

Inventory control helps in reducing the inventory cost by increasing the chance of providing the product promptly to the customer. It is one of the parts of the Inventory Management.



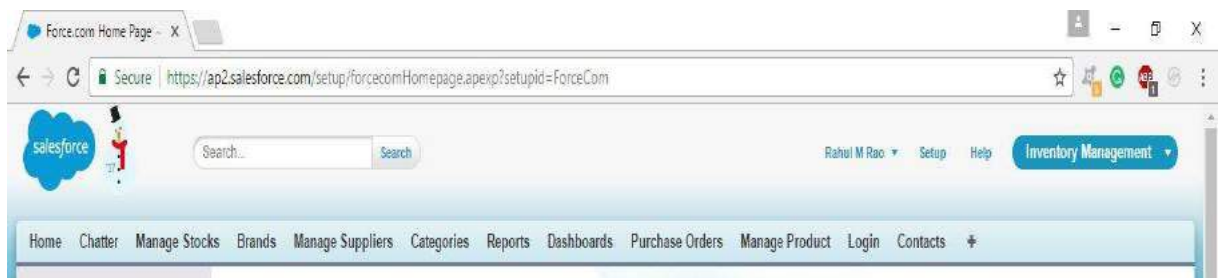
Figure 9: Inventory Management System

### 1.4.2 Development Area

As Salesforce is a cloud-based platform, all it needs is a browser and internet connection. Once a developer or user of force.com platform logs in into salesforce.com, he sees his name on the top right corner of the page, and beside that, there is a setup button which we use mostly while developing an app.

- Open a browser and enter [www.login.salesforce.com](http://www.login.salesforce.com)
- Enter Username and Password

It will take us to the Salesforce customized company domain and it will look like as shown below.



**Figure 10: Salesforce Home Page**

After the Setup button in the header, we can see the Help button which will give detail information about anything in this platform and lastly we can see a drop-down list showing Inventory Management, where we can see the list of apps available in this domain. By selecting the Inventory Management app, we can see all the data/ modules related to this app along with some predefined standard data.

### 1.4.3 Custom Apps

In Salesforce, creating an App is just a point and click away, to create a new app click on Setup and locate Create, then click on Apps and then on New. As simple as shown below, enter the name of the app as Inventory Management after clicking New.

Setup   Create   Apps   New

Salesforce provides some predefined apps like sales which consist of standard objects and standard fields. Now, we are creating our custom app which will consist of custom objects and custom fields to hold the data.

### 1.4.4 Objects

Objects in Salesforce are database tables with information. The primary object in the Salesforce data model represents accounts/companies and organizations involved in the business, such as customers, partners, and competitors. A record is similar to a row in a database table.

Objects already created by Salesforce are called standard objects. Objects we create in organization are called custom objects

Created the custom object Brands in the following way,

- Click on Create Objects Custom Object.
- Enter the Label name as Brand, Plural name as Brands which be used as Tab name and Object name as Brand which will be used as API name. By default, the custom objects are stored with \_\_c to access through API.
- Enter the record name Brand Title which is a mandatory field while creating a custom object and select its data type as Text.
- Next, check allow reports, activities and track field history check box fields, which are basically for availing report creation, assigning tasks to the user for a particular Brand record and tracking the history of all field edits and updates.

In similar fashion, For the Inventory Management App, six custom objects were created.

1. Brand: Tracks all the brands of products in Inventory
2. Category: Tracks the categories of products in the Inventory
3. Manage Supplier: Tracks the details of the suppliers of the products.
4. Manage Products: Tracks the records of the products in the Inventory
5. Manage Stock: Tracks the record Stock in the Inventory
6. Purchase Order: Tracks the record of the Purchases at the point of sale.

The objects can be accessed by tabs in salesforce.com and menu items as in the salesforce1 platform which is a mobile platform.

#### 1.4.5 Fields

Salesforce has provided many inbuilt fields called standard fields like “Name,” “Owner,” “Created By,” “Last Modified By” etc. Every object in Salesforce has a set of standard fields that may be applicable for capturing data for that type of an object.

Salesforce also allows users to create new fields in the system to capture additional information. These fields are called as custom fields. Salesforce supports a variety of datatypes for these fields like Text, Text Area, Rich Text Area, Number, Currency, Boolean, Email, Phone, etc.

Administrators can then configure how these fields should be set (visible/readonly/edit/mandatory).

Created some more custom fields to the object **Brand** in the following way,

🔗 Create Objects Brand Custom Fields & Relationships.

### 1.4.6 Relationships

The Force.com platform supports two types of parent-child relationships between objects. They are lookup relationships and master-detail relationships. These relationships connect objects with other objects. These both work like a foreign-key relationship in a relational database. These relations are the fields in the Salesforce. We can create them as we create custom fields.

**The lookup relationship** creates a simple relation between two objects. With this relationship field, we can navigate from records in one object to the related records in another object and can create one-to-one and one-to-many relationships. Lookup relationships are appropriate when a relationship between two objects is required in some cases, but not always. In scenarios like to relate multiple parent records to the child record and to reference commonly shared data, such as reference data, a lookup relationship is used.

**The master-detail relationship** is a powerful relationship which is based on a parent-child relationship. The object on which we create a master-detail relationship is the child, and the other one is the parent or master object which will be referenced as a field in the child object. The sharing settings of the child object can be taken from their master object. In a master-detail relationship, if we delete master object records, child objects also automatically deleted. With the master-detail relationship, we can create a roll-up summary field on parent objects.

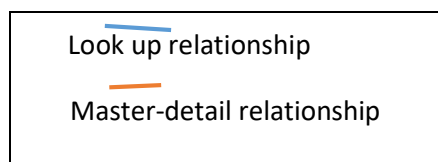
Example:

In Figure 11, charity is the master of events object and donation is the child object of events. So with the Master-detail relationship between them, we can see total money raised, in the charity object which is a roll up a summary of the total donations of the events object.



Figure 11: An example of a Master-Detail relationship.

In the Inventory Management app, there is a master-detail relationship in between brand and manage product object and in between manage category and manage product objects. The manage product object is the parent of both manage category and brand objects. There is look up the relationship between the delivery order and contact object and then the contact object with a purchase order. The manage the product object have lookup relationship between the purchase order and manage supplier object.



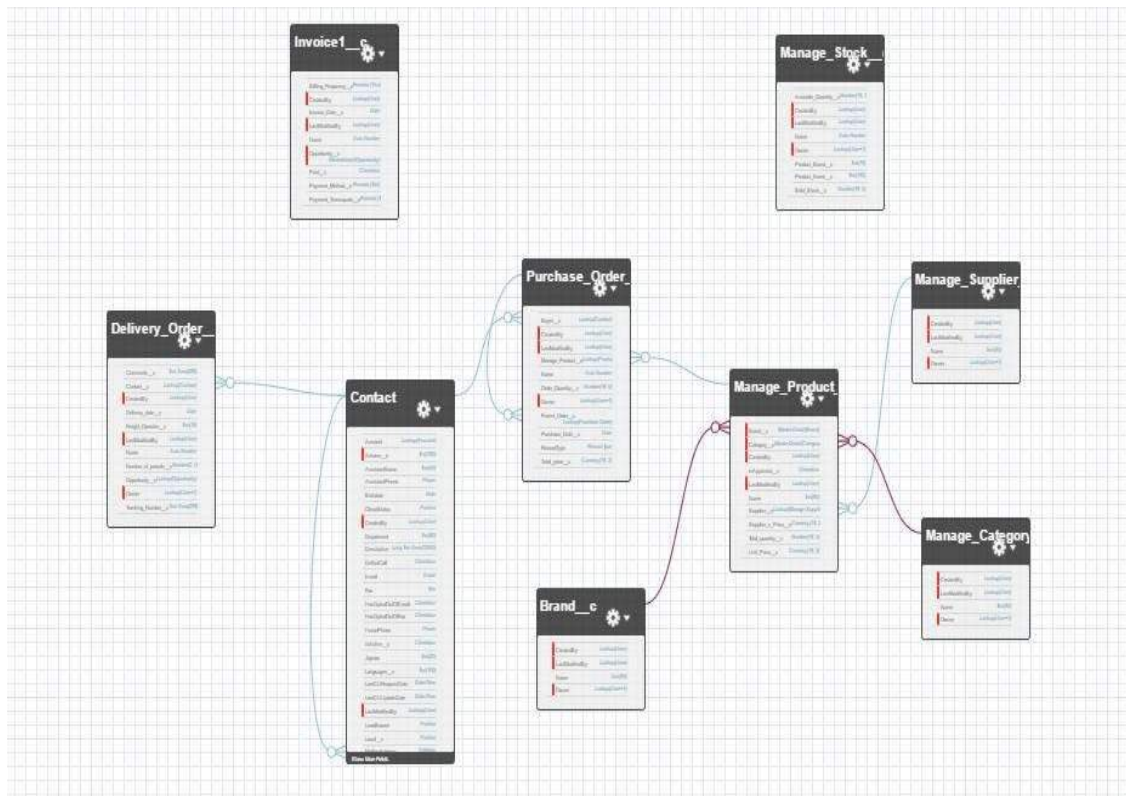


Figure 12: Relationships between objects

### 1.4.7 Tabs

Tabs are the primary way to access the main objects in the application for users in Force.com platform apps. Just like standard objects/fields Salesforce also has the standard/custom tabs. Tabs provide an interface to access records for that particular object. Most standard objects have their tabs exposed. Users can also configure which tabs they would like to see and can rearrange their order. Users can create custom tabs for the custom objects they create. Tabs can also be used to open custom pages and links.

Created a custom Tab to the object **Brand** in the following way,

☑ Create ☑ Tabs ☑ Brands

The Custom Tabs that were created for the custom objects are Brands Tab, Categories Tab, Manage Stocks Tab, Manage Suppliers Tab, Products Tab, Purchase Orders Tab

The other Tabs that are used for visual force Pages Are Login Tab and Manage Product Tab



### **1.4.8 Securing and Sharing data**

Salesforce provides a wide level of security and sharing of data in an organization. Some of them are listed below.

1. Object Level Access: These controls access to objects in the Organization.
2. Field Level Access: These controls access to particular fields in an object.
3. Record Level Access: These controls access to the records of an Object.
4. Profiles: Every user has one profile and this profile is used to control many things like user permissions, object permissions, field permissions, app settings, tab settings, Visualforce page access, Apex class access, page layouts, login hours, and login IP ranges and so on.
5. Permission Sets: Every user can have one profile but can have multiple permission sets. Permission sets to control access of object permissions, field permissions, user permissions, tab settings, app settings and it also controls Apex class access and Visualforce page access.
6. Field Level Permissions: These controls access of a user to view, edit and delete of fields of an object.
7. Organization-Wide Default Settings : These are default controls set by the organization and these controls access to the records of an object.
8. Role Hierarchies: With this, the user who is above in the role hierarchy will have access to the records of their subordinates.
9. Sharing rules: These controls access to the records of an object but these makes exceptions to the organization-wide default.
10. Manual sharing: It overrides the all the restrictions to access the records. One can share records with specific users.

The profile and the user permissions restrict what the user can see. The modifications made in the desktop Salesforce site are reflected in the app even and mobile users to get their Organization's data without special configuration.

### **1.4.9 Page Layouts**

Page Layouts can be used to control the visibility of the custom fields, button and links on the Object record. We can rearrange the fields as per our convenience and by separate sections also. We can make a field to be visible, required or read-only field to control the access for different users, marks few fields as read only, mark few fields as mandatory, etc. Page layouts can be created in the following way,

Create   Objects   (Custom Object)   Page Layouts



#### **1.4.10 Visual Force**

Visualforce is a complete framework for making such UIs, empowering any interface outline and communication to be constructed and conveyed altogether in the cloud. The UIs that work with Visualforce can expand the standard Force.com stage look and feel, or supplant it with an impressive style and set of sophisticated interactions. Since Visualforce markup is rendered into HTML, designers can utilize Visualforce tags along with standard HTML, JavaScript, Flash, or whatever other code that can execute inside an HTML page on Platform. Visual Force pages can enable in the mobile by just checking the “enable the page for mobile apps” box field after creating it. With the CSS, the visual force pages can be optimized for the mobile devices and match the look of the Salesforce1.

For the Inventory Management app, more than 10 VF pages are created. The log in page, the home page, manage the product page, the insert product page, the invoice page and the payment page, are some of them.

#### **1.4.11 Apex**

Salesforce introduced Apex as the first cloud computing programming language. The syntax of Apex is quite similar to Java. It is particularly intended for building business applications to oversee Data and procedures larger ambience of the Force.com platform. The Apex lets the developers focus just on elements specific to their application by providing a productive approach to creating functionality and logic, leaving the rest of work for Force.com Platform.

Usually, for every interactive VF page, Apex Class is associated. So more 10 Apex Classes are used in the Application.

#### **Chatter**

Chatter is a standard Salesforce functionality which is very useful feed tracking system in Salesforce where users can post to feed, comment on feeds, shares the information, attach files and share. Also, the user can follow other users to get updates from that user. The user can follow a particular field of an object and can get updates whenever its value changed or updated. It reduces the mailing efforts for a user to update about things going on the organization. Also, create groups who are working on an assignment and keep an update or follow up with them by enabling chatter. It's like a social network application for the work environment to connect people with updates going in their organization.

## Reports and Dashboards

As we know a company deals with lots of data, to check the data record by record is a very lengthy and time taking process. A good application should provide the overview of the data at a glance. For this purpose, Salesforce provided a tool called Reports. Using reports, we can generate different formats of data resides in the company at a glance view. A dashboard is a place where we can put all reports together to give the overview of the statistics. For example, if the manager wants to know how many products are sold from the inventory, he can create reports and find these statistics. Salesforce supports four different formats of Reports which are,

- **Tabular Reports:** These give a simple list view of data with Grand totals. But these are not used to create groups of data and graphs.
- **Summary Reports:** These give the summary of the data like subtotals with row-wise also. These are used to create charts, graphs. These reports can be used in the dashboard.
- **Matrix Reports:** These give the groups of data both by row wise and column wise. These are the most time consuming to generate a report or to set up, but can also be used in the dashboard.
- **Joined Reports:** These allow to join the multiple reports of different types. These reports can also be used to create charts so that can be used in dashboards.

The Reports can be created as follows,

Create □ Apps □ Add Reports Tab □ Reports.

A Dashboard can show the reports in a visual format like Charts, Gauges, Tables, Metrics, or Visualforce pages. Force.com platform supports 20 components to be allowed in a dashboard for an organization. For example, the reports which we created before, like Pie chart and Vertical bar chart we can put up to 20 components in a dashboard. For the dashboard also we created a folder and saved all the dashboards under Inventory Dashboards folder and shared same as Reports folder. Dashboard can be created as,

Reports Tab □ Reports & Dashboards □ New Dashboard.

## 1.5 The Functionality of Inventory Management

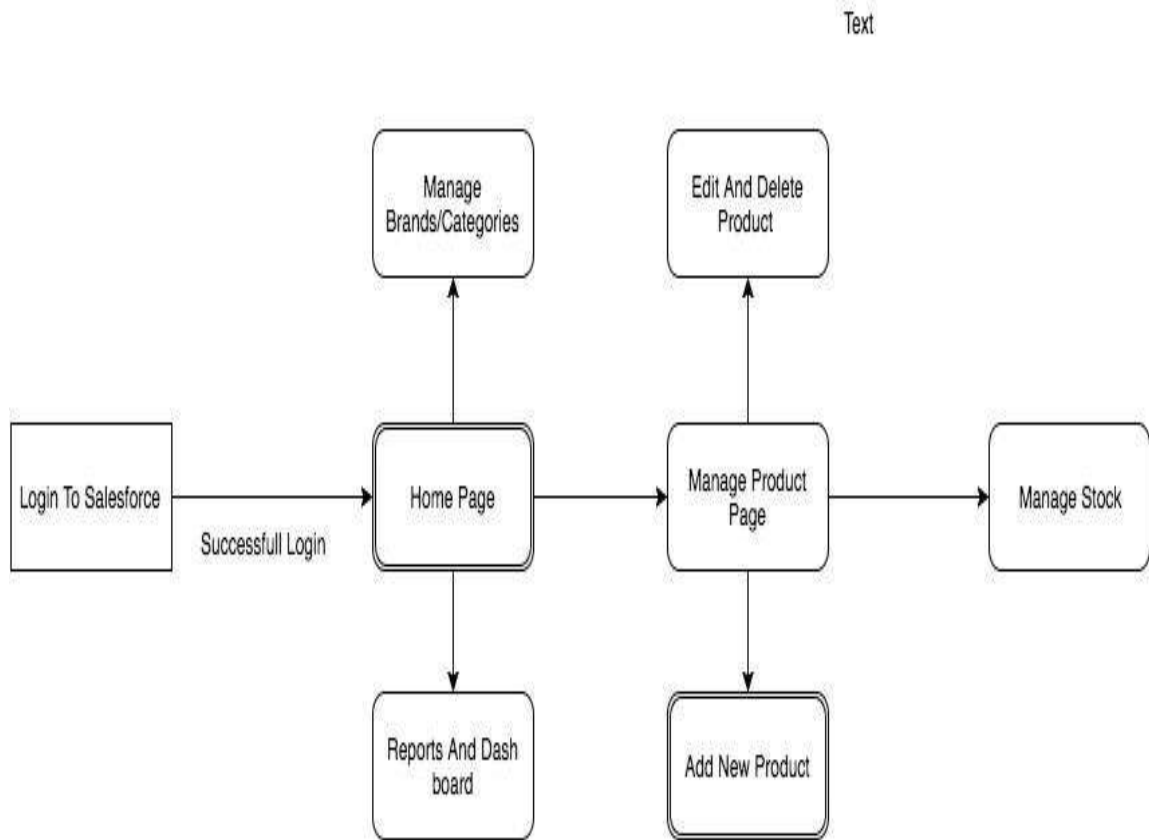
The Inventory Management Application can be divided into two parts i.e.

- 1) Inventory Administration
- 2) Point of Sale or Customer Portal

### 1.5.1 Inventory Administration

Inventory administration is carried out usually by store manager or store admin.

### Schema

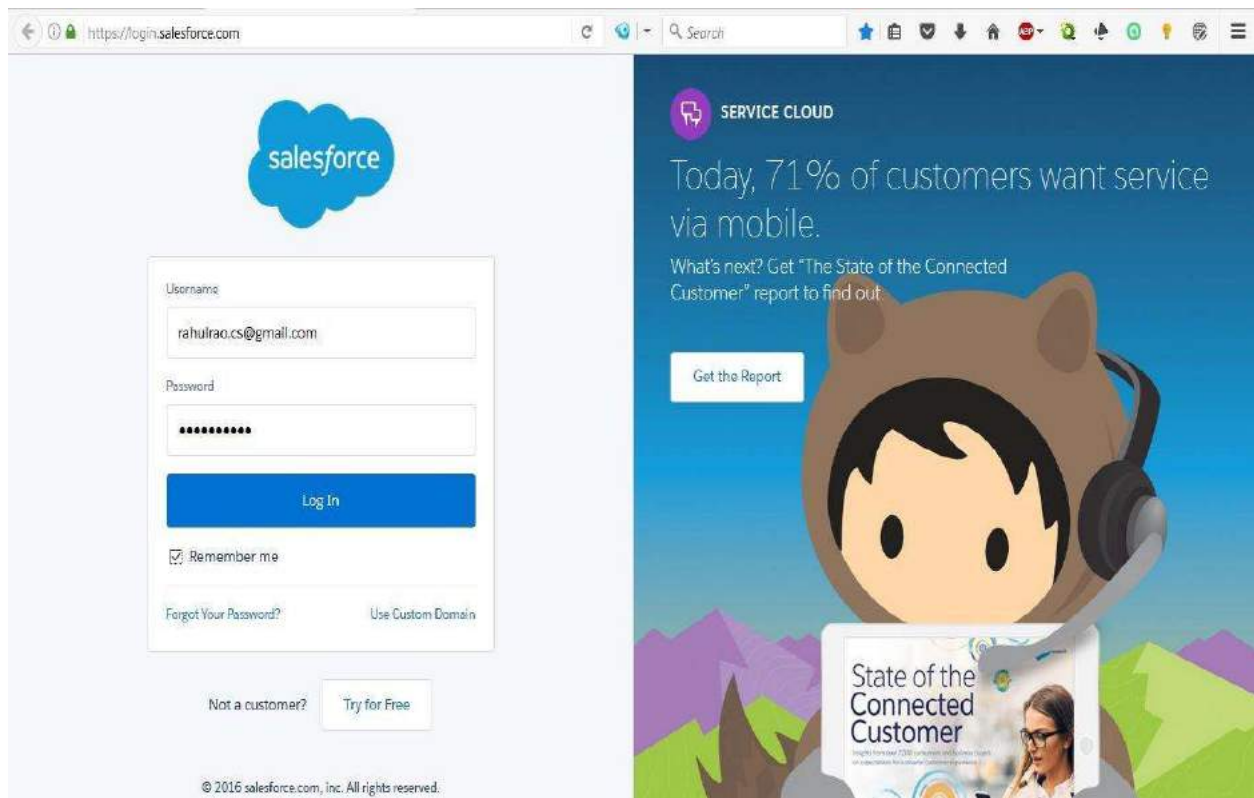


**Figure 13: Schema of Inventory Administrator**

The Manager can do the following functions.

The Store Manager can add, edit, delete and view the products in the database. The various functions that a manager can do include updating new brands and suppliers, generating reports (based on the category, brand, created date) and the access to see and edit the customer details and all purchase orders details.

To administrate his inventory, Manager needs to log into Salesforce with his credentials.



Once the admin logs into the Salesforce, the log in page is redirected to home page. In the home page, the manager finds Standard Page components like apps, tabs, menu and he will also find custom modules like calendar, task to perform, etc., In the home page, the manager can navigate between all the tabs, existing applications, and other components.

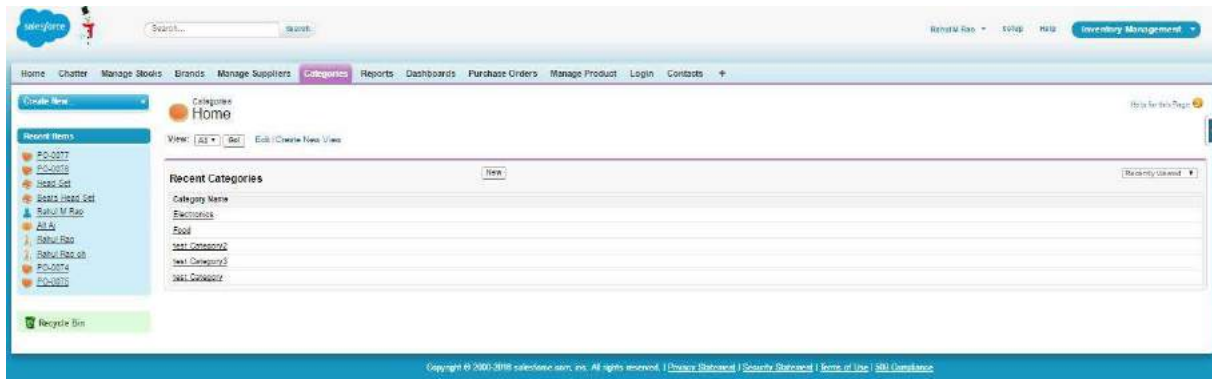
Home Page can be customized depending on the requirements. The layout of the page is fixed while the components like tabs can be customized. In the inventory management app, home page is customized.

In the customized home page, manager can view

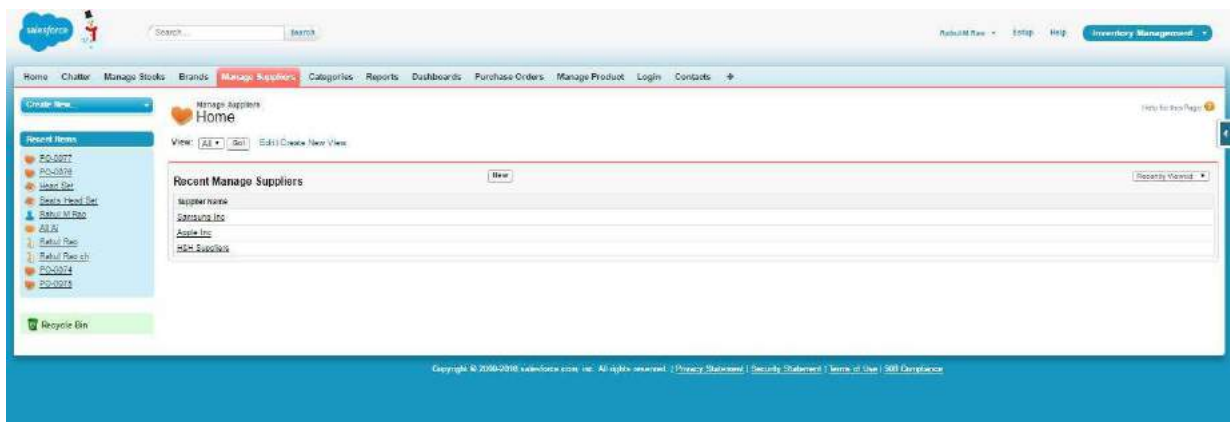
1. Calendar,
2. Tasks to perform,
3. Dashboards
4. Record need his approval
5. Recent Activities

In addition to standard page components like Tabs, Apps, etc.





The manage suppliers tab has access to add, view and delete the suppliers for products. Sorting filters like recently viewed recently created and recently modified are provided for this tab also.



Similar to above features, the manage product tab provides access to store manager on products in the database. Stocks can be viewed and managed in this page.

Secure | <https://cap2.visual.force.com/apex/ProductTablePage?sfdc.tabName=01r28000000fM7J> Go to Force.com

Add Product		Manage Stock		Delete		
Check To delete	Click to Edit	Product Name ↑	Brand	Category	Unit Price	Available Quantity
<input type="checkbox"/>	<a href="#">Edit</a>	Beats Head Set	Apple	Electronics	\$200.00	125
<input type="checkbox"/>	<a href="#">Edit</a>	Head Set	Samsung	Electronics	\$300.00	45
<input type="checkbox"/>	<a href="#">Edit</a>	Iphone	Apple	Mobiles	\$200.00	5
<input type="checkbox"/>	<a href="#">Edit</a>	Iphone	test brand	Mobiles	\$500.00	2
<input type="checkbox"/>	<a href="#">Edit</a>	Moto g3	Motorola	Mobiles	\$210.00	0
<input type="checkbox"/>	<a href="#">Edit</a>	Note	test brand	Mobiles	\$200.00	3
<input type="checkbox"/>	<a href="#">Edit</a>	Sony TV 42 in	Sony	Electronics	\$300.00	64
<input type="checkbox"/>	<a href="#">Edit</a>	TV 20 IN	Sony	Electronics	\$150.00	9

The store manager can add new products based on category, brand, and supplier. Other attributes like the quantity of the product, the unit price and the supplier price of the product are even added to the new product.

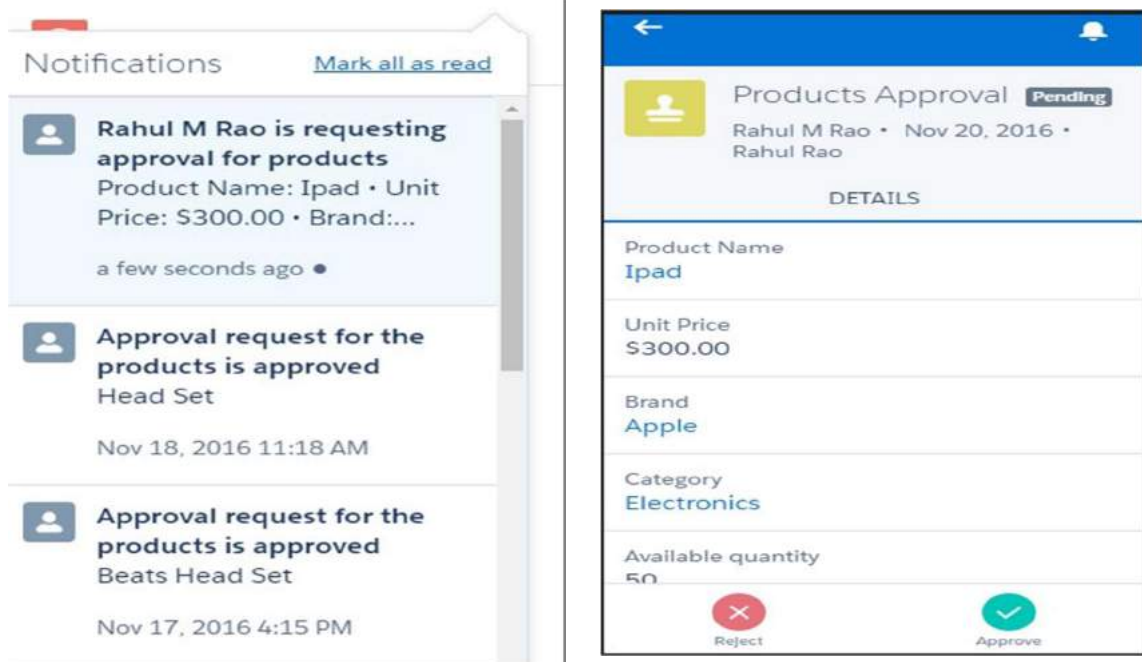


Product name	<input type="text" value="Ipad"/>
Unit Price \$	<input type="text" value="300"/>
Supplier's Price \$	<input type="text" value="200"/>
Quantity	<input type="text" value="50"/>
Select Brand	<input type="text" value="Apple"/>
Select Supplier	<input type="text" value="Apple Inc"/>
Select Category	<input type="text" value="Electronics"/>

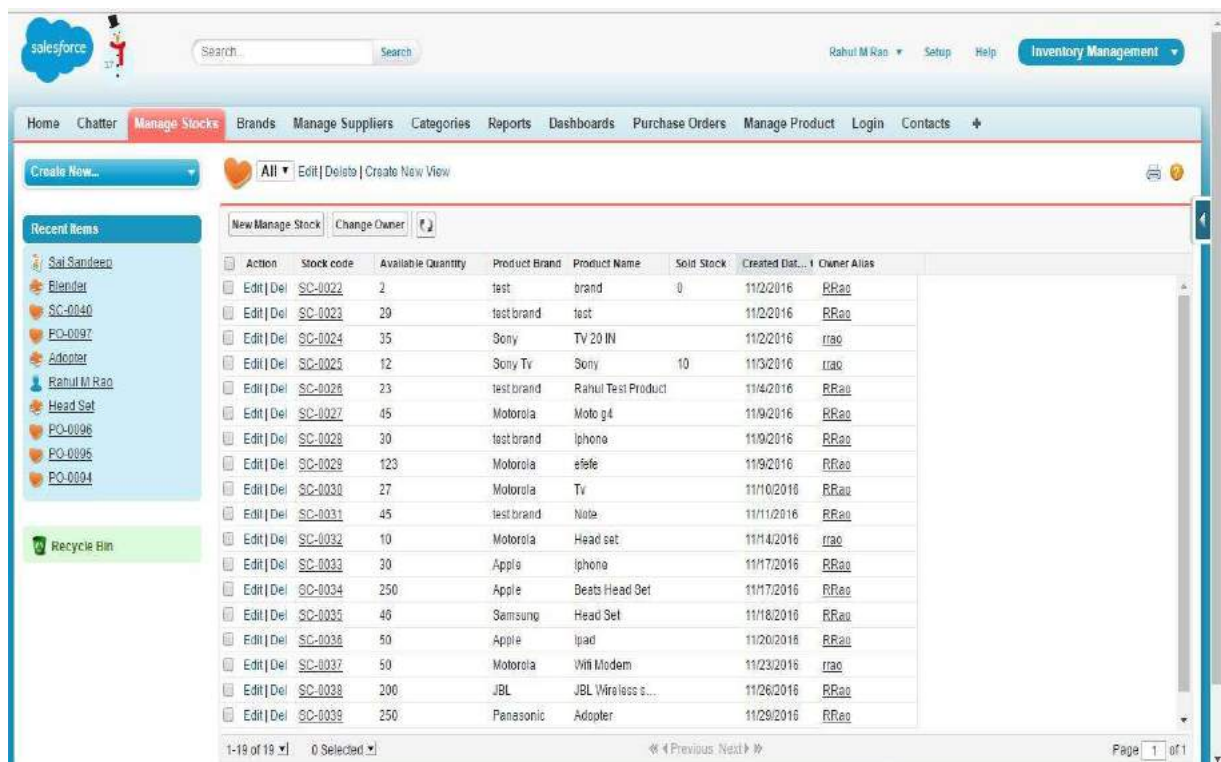


Only the store manager has master access to the database. Input fed by employees is only reflected in the inventory database on approval from a manager. The employee has to raise an approval request.

Salesforce App 1 View:



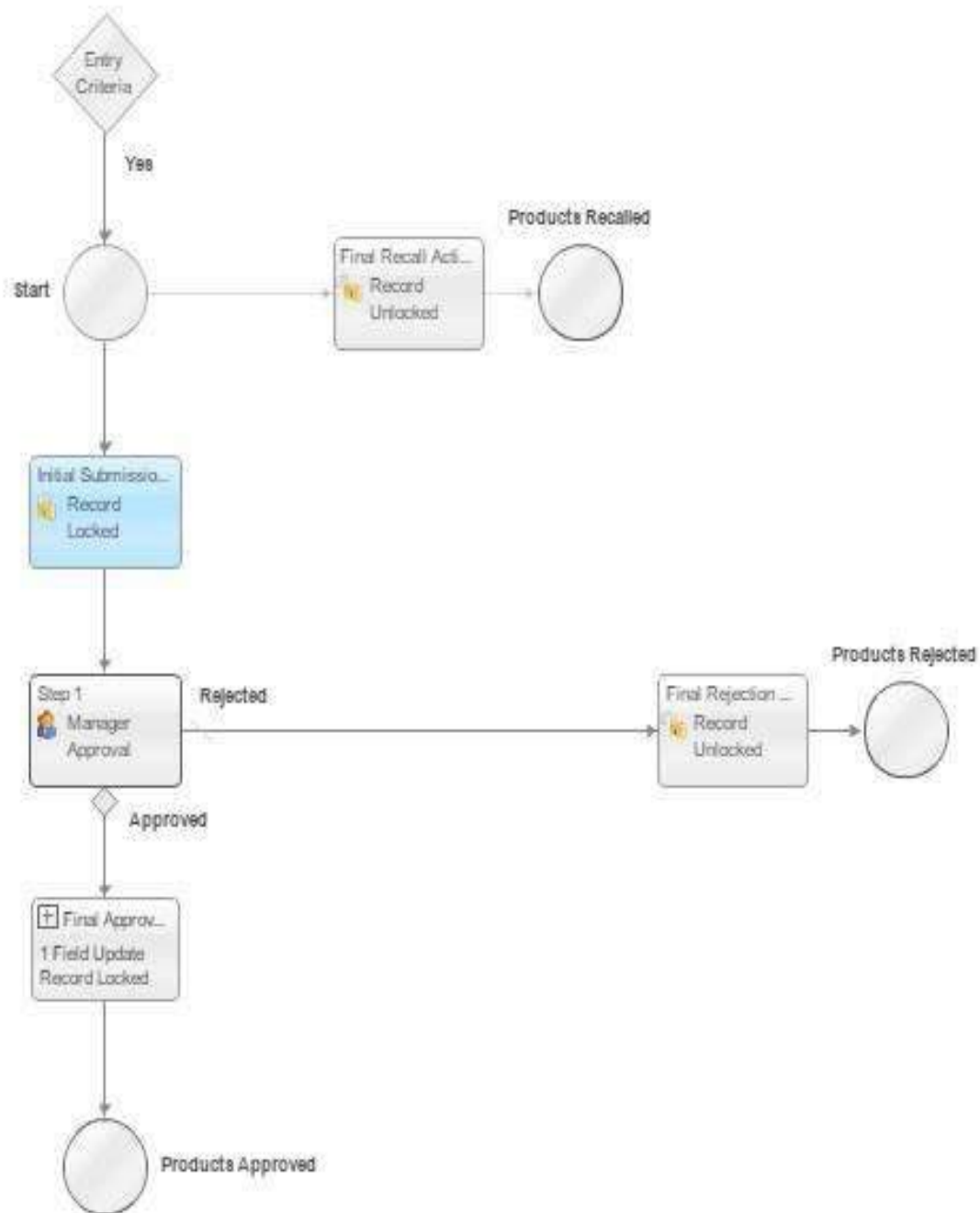
The Manager can reject or approve a product. If he approves the product, it is reflected in the stock.





## Approval Process

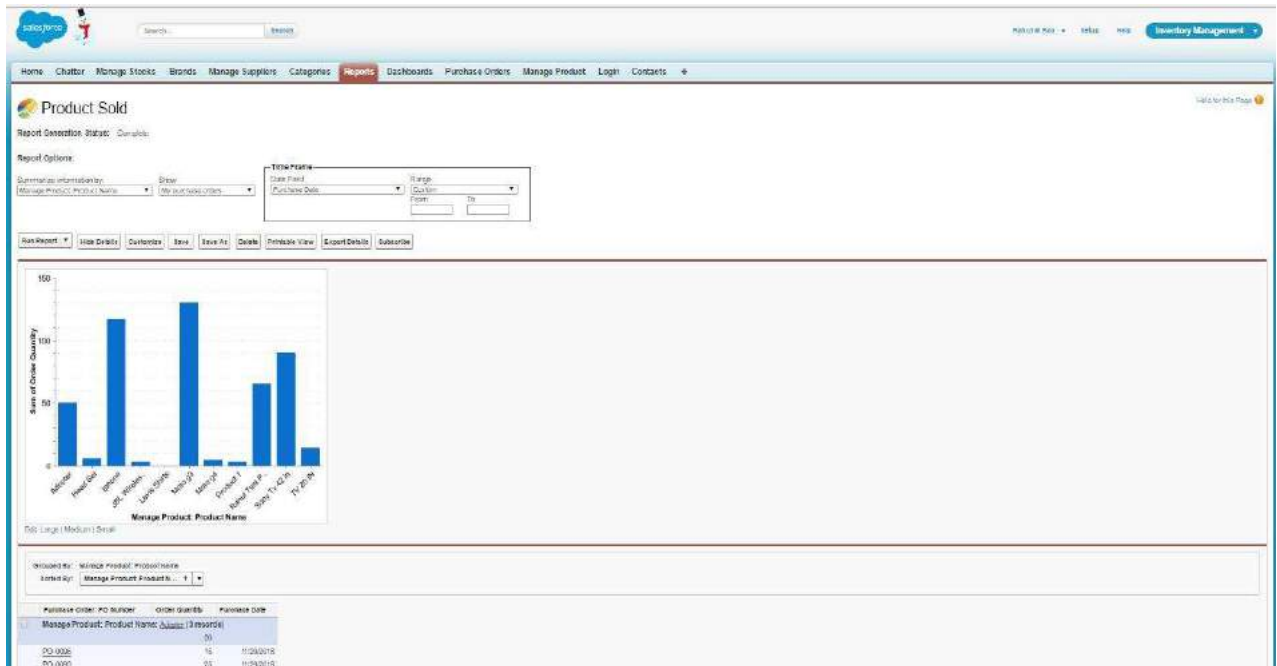
Every product must get the approval of store manager before it goes into the inventory. Below is the approval process flow.



**Figure 14: Product Approval Process**

The store manager can generate reports based on

- Categories ☐ Brands ☐ Time.
- Products sold



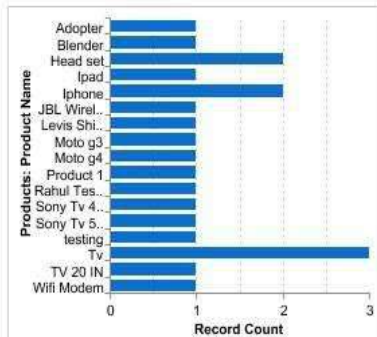
Data Generated from the reports can be viewed in dashboards in formats like

- Horizontal bar chart
- Vertical bar chart
- Line chart
- Pie chart
- Donut chart and
- Funnel chart

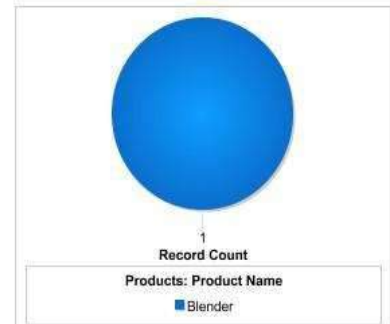
## My Products

Find a dashboard... Edit Clone Refresh As of Today at 5:40 AM

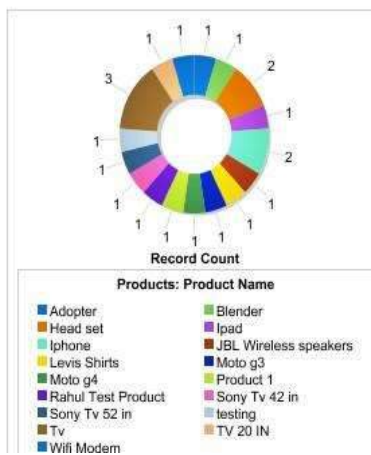
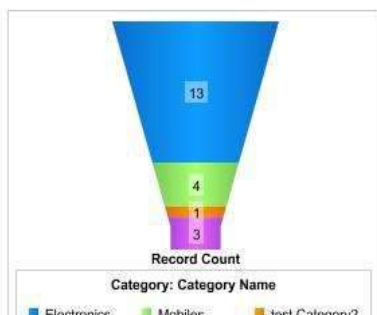
Products this Year



Products this Month



Products by Brand

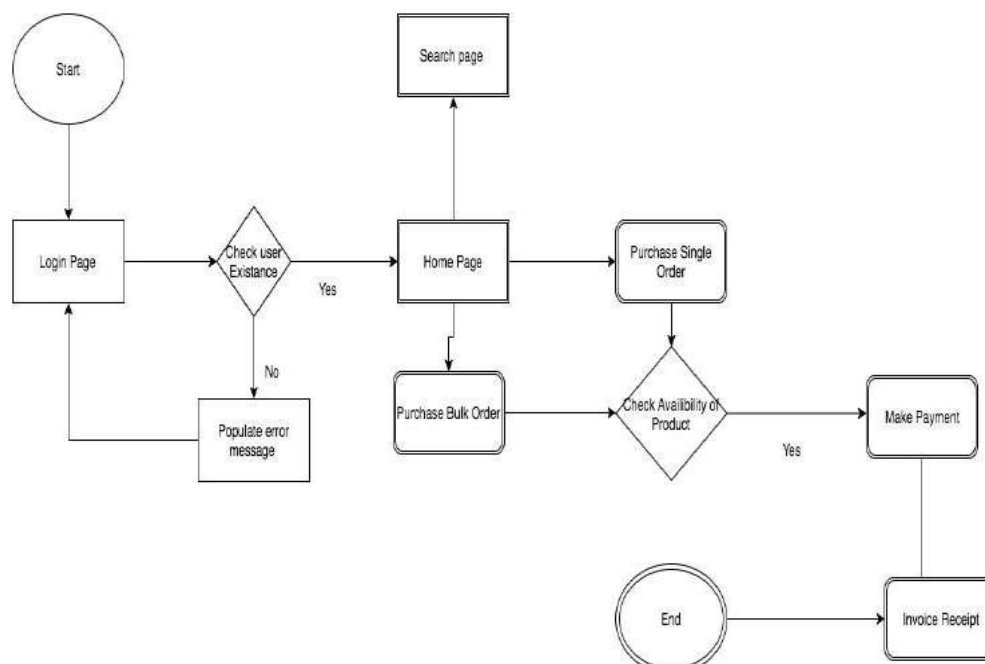


Products By Category



## 1.5.2 Customer Portal

The customer can perform following functionalities.



**Figure 15: Workflow of Customer Portal**

The customer needs to log in into the Salesforce. A new customer has to register with valid email id, full name, and security questions to get access to the user portal.

### Registration Page

**REGISTER TO PURCHASE PRODUCT**

**PRIMARY USER INFORMATION**  
**Your title:**  
  
**First name:**  
  
**Last name:**  
  
**Email:**  
  
**Confirm email:**  
  
**Password:**  
  
**Confirm Password:**  
  
**Security question:**  
  
**Avatar:**

**ORGANIZATION INFORMATION**  
**Organization type:**  
  
**Organization name:**  
  
**Primary phone:**  
  
**Country:**  
  
**Mailing street address:**  
  
**Mailing city:**  
  
**State:**  
  
**Zip:**  
  
**Supervisor name:**  
  
**Supervisor title:**  
  
**Supervisor phone:**

CANCEL

REGISTER


On successful registration, the new user receives a mail from the admin confirming his registration, from when access is available.

### Registration Confirmation Email

Registration Confirmation



Inbox x

People (3)

 **Product Management Team** ajsalesforcedemo@gmail.com via hxxwvzqz77qxit.28-1tlbueaa.a

to me, rahulrao.cs


Nov 12 (8 days ago)

Dear Rao



Thank you for connecting with us. We will update you once your account is active.

Thanks,  
Admin

 Click here to [Reply](#), [Reply to all](#), or [Forward](#)

**Product Management Team**

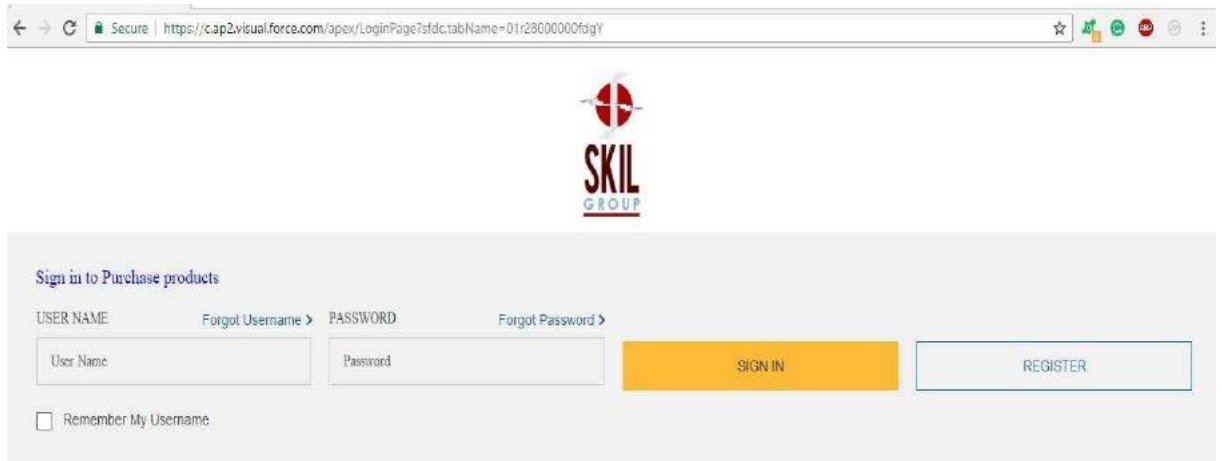
Add to circles

Show details

## Login Page

On confirmation of approval from the admin/store manager, the user can now login in into the portal.



Sign in to Purchase products

USER NAME [Forgot Username >](#) PASSWORD [Forgot Password >](#)

User Name Password SIGN IN REGISTER

☐ Remember My Username

## Forgot Username

‘Forgot Username’ icon is available for the user to retrieve username through registered email address if they do not remember the current user name.



**Forgot User Name**

Enter the email address you signed up with to have your user name emailed to you.

Email:

Submit

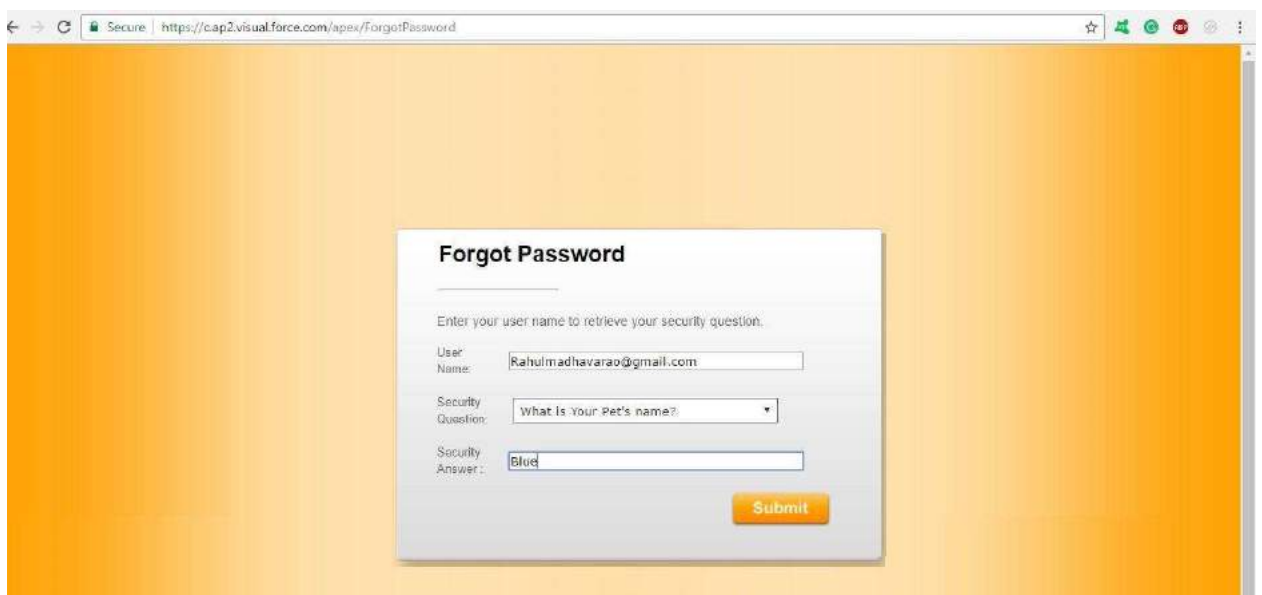
The request mail would only be sent if the email address provided by the user matches with the registered email address, available in the database.

## Forgot Username Email



## Forgot Password

The similar process of recovering username is to be followed to 'Forgot Password'. A link would be sent to a registered email address when the user has to input valid username and answer his security question.



## Forgot Password Confirmation

If the username and security question matches with the data in the database, then he receives a mail with his password



## Homepage

When user signs into the login Page, the login page redirects to the home page where he can search the products in the inventory and then he can purchase the products depending on the quantity



Options :
# Search Product
# Purchase Single Product
# Purchase Bulk Product

## Search Products

The user can search for products in the inventory based on brand, category and by product name. The available quantity and price of the product are displayed to the user. After selection

SEARCH PRODUCT [Home](#)

Enter Product Name

Enter Brand name

Enter Category Name

SEARCH

#	Product Name	Brand	Category	Unit Price	Available Quantity
1	Beats Head Set	Apple	Electronics	200.00	150

## Single purchase product

The user can continue with purchase options if he is looking for just one product. The quantity of the product can be chosen based on requirement and availability and the purchase can be made using a credit card. After payment, an invoice is generated.





Enter Product information

Home

Select Product

Available Quantity

Unit Price

Purchase Quantity

Date Of Purchase

Total Price

CANCEL

PURCHASE

### Bulk Purchase Product

Similarly, if multiple products are to be bought, it can be carried out from Bulk purchase page. The quantity of the product can be chosen based on requirement and availability and the purchase can be made using a credit card. After payment, an invoice is generated.

SEARCH PRODUCT

Home

Select Product

PURCHASE

Action	Product Name	Category	Brand	Unit Price	Available Quantity	Purchase Quantity	Total Price
Delete	Iphone	Apple	Mobiles	200.00	5	6 <small>Requested Quantity is not available for Iphone</small>	800.0
Delete	Beats Head Set	Apple	Electronics	200.00	125	25	5000.0

The user can select the product to buy along with the quantity as required. The unit price and the total price for the required quantity are displayed under this tab for convenience. Using credit card details, the items in the cart can be purchased.

CARD INFORMATION

Card holder Name

Bill Gates

Card type

Visa

Card number

1234567890

csv number

123

Expire Date

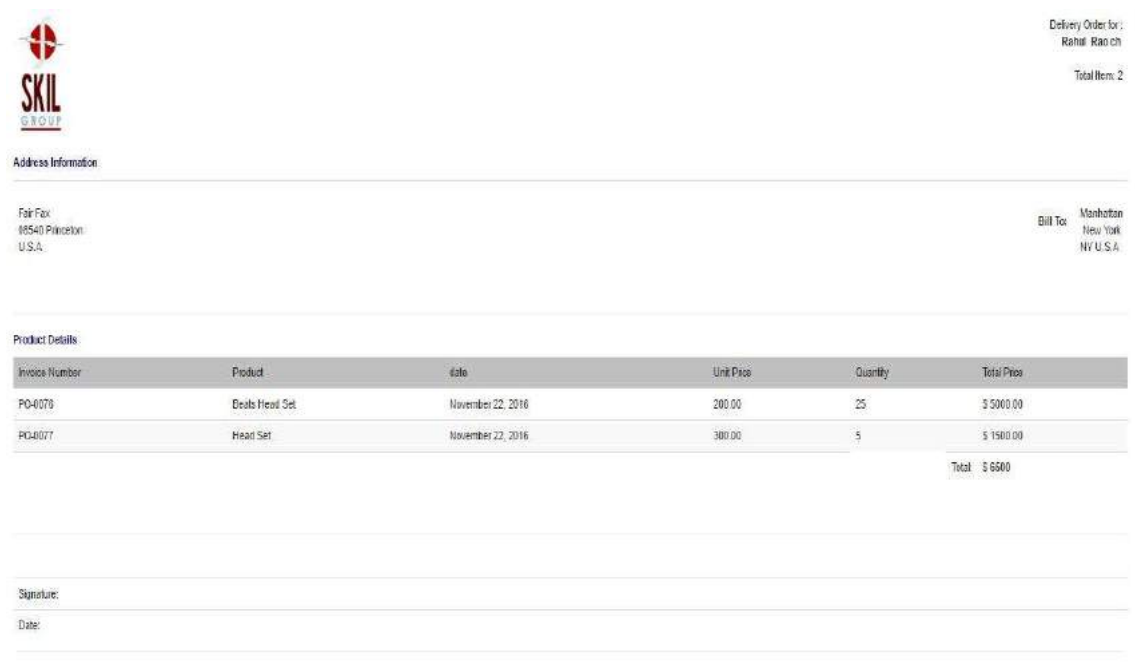
11

17

CANCEL

PAY

As payment is made, an invoice is generated. The invoice contains the company details like the registered name and address along with customer details such as his full name, his address, followed by the items he purchased, quantities of the items, the unit price of each item and the total price of purchase.



**SKIL GROUP**

Delivery Order for :  
Rahul Rao ch  
Total Item: 2

Address Information:

Fair Fax:  
16540 Princeton  
U.S.A.

Bill To: Manhattan  
New York  
NY U.S.A.

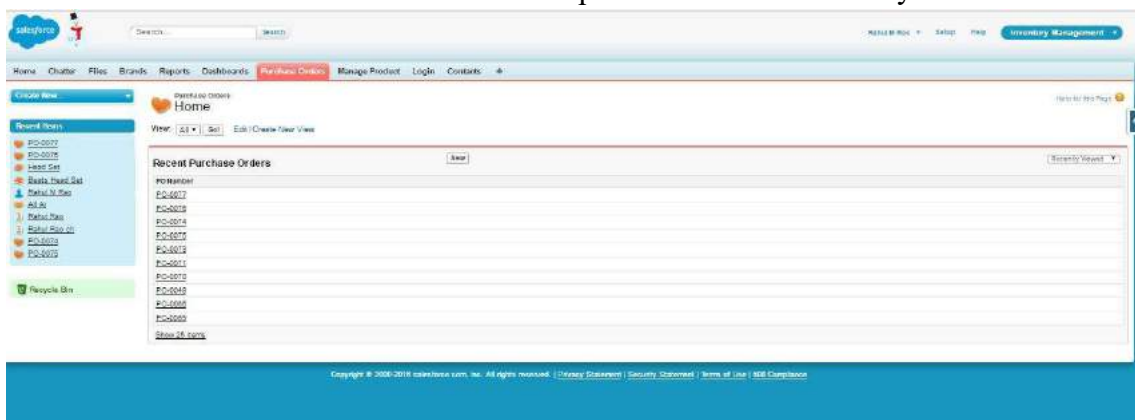
Product Details:

Invoice Number	Product	date	Unit Price	Quantity	Total Price
PO4076	Beats Head Set	November 22, 2016	200.00	25	\$ 5000.00
PO4077	Head Set	November 22, 2016	300.00	5	\$ 1500.00
<b>Total</b>					<b>\$ 6500</b>

Signature:

Date:

Once the purchase is completed, details are stored in purchase order and the inventory stock is altered and the same is updated in the inventory stock database.



VisualForce

Home Chatter Files Brands Reports Dashboards **VisualForce Objects** Manage Product Login Contacts

Home

Recent Purchase Orders

PO Number	Status
PO-0001	Completed
PO-0002	Completed
PO-0003	Completed
PO-0004	Completed
PO-0005	Completed
PO-0006	Completed
PO-0007	Completed
PO-0008	Completed
PO-0009	Completed
PO-0010	Completed
PO-0011	Completed
PO-0012	Completed
PO-0013	Completed
PO-0014	Completed
PO-0015	Completed
PO-0016	Completed
PO-0017	Completed
PO-0018	Completed
PO-0019	Completed
PO-0020	Completed
PO-0021	Completed
PO-0022	Completed
PO-0023	Completed
PO-0024	Completed
PO-0025	Completed
PO-0026	Completed
PO-0027	Completed
PO-0028	Completed
PO-0029	Completed
PO-0030	Completed
PO-0031	Completed
PO-0032	Completed
PO-0033	Completed
PO-0034	Completed
PO-0035	Completed
PO-0036	Completed
PO-0037	Completed
PO-0038	Completed
PO-0039	Completed
PO-0040	Completed
PO-0041	Completed
PO-0042	Completed
PO-0043	Completed
PO-0044	Completed
PO-0045	Completed
PO-0046	Completed
PO-0047	Completed
PO-0048	Completed
PO-0049	Completed
PO-0050	Completed
PO-0051	Completed
PO-0052	Completed
PO-0053	Completed
PO-0054	Completed
PO-0055	Completed
PO-0056	Completed
PO-0057	Completed
PO-0058	Completed
PO-0059	Completed
PO-0060	Completed
PO-0061	Completed
PO-0062	Completed
PO-0063	Completed
PO-0064	Completed
PO-0065	Completed
PO-0066	Completed
PO-0067	Completed
PO-0068	Completed
PO-0069	Completed
PO-0070	Completed
PO-0071	Completed
PO-0072	Completed
PO-0073	Completed
PO-0074	Completed
PO-0075	Completed
PO-0076	Completed
PO-0077	Completed
PO-0078	Completed
PO-0079	Completed
PO-0080	Completed
PO-0081	Completed
PO-0082	Completed
PO-0083	Completed
PO-0084	Completed
PO-0085	Completed
PO-0086	Completed
PO-0087	Completed
PO-0088	Completed
PO-0089	Completed
PO-0090	Completed
PO-0091	Completed
PO-0092	Completed
PO-0093	Completed
PO-0094	Completed
PO-0095	Completed
PO-0096	Completed
PO-0097	Completed
PO-0098	Completed
PO-0099	Completed
PO-0100	Completed

Copyright © 2006-2016 salesforce.com, Inc. All rights reserved. | Privacy Statement | Security Statement | Terms of Use | Sitemap

## Appendix

The code for Visual Force Page of Manage Product is below:

```

<apex:page standardstylesheets="false" standardController="Manage_Product__c" docType="html-5.0"
showheader="false" applyHtmlTag="true" extensions="ManageProductController" title="All Product"
tabStyle="Manage_Product__c">
  <style>
    .btnClass {      border: 0;
outline: none;      background: black
!important;        width: 100%;      border-
radius: 5px;        height: 40px;    color:
white;              margin: 20px 0 0 0;  font-
size: 16px;
    }
  </style>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
/>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  < script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>      <div
class="container">
  <apex:form styleclass="form-horizontal">
    <apex:outputPanel styleClass="myCustomMessage">
      <apex:pageMessages ></apex:pageMessages>
    </apex:outputPanel>
    <div class="form-group ">
      <label> &nbsp;  </label>
      <center>
        <apex:image value="{!URLFOR($Resource.ProdImage)}"></apex:image>
      </center>
    </div>
    <div class="form-group">
      <label class="col-sm-3 control-label" for="textinput">Product name</label>      <div
class="col-md-6">
        <apex:inputText id="name" value="{!productName}" styleClass="form-control input-md"
        />
      </div>
    </div>
    <div class="form-group">
      <label class="col-md-3 control-label" for="textinput">Unit Price $</label>      <div
class="col-md-6">
        <apex:inputText id="UPrice" value="{!unitPrice}" styleClass="form-control input-md"
        />
      </div>
    </div>
    <div class="form-group">
      <label class="col-md-3 control-label" for="textinput">Supplier's Price $</label>      <div
class="col-md-6">

```



```

public class ManageProductController {

//variables   public String categories {get; set;}
public String brands {get;set;}   private final Integer
COMPARE_INT = 0;   public Manage_Product__c
manageProdVar {get;set;}   public String
productName {get;set;}   public Integer unitPrice
{get;set;}   public Integer supplierPrice {get;set;}
public String suppliers {get;set;}   public Integer
quantity {get;set;}   private final String NONE_VALUE
= '--None--';

//Constructor   public ManageProductController(ApexPages.StandardController
stdController){

}

//Method to get the list of all Categories   public
List<SelectOption> getCategory () {
List<SelectOption> options = new List<SelectOption>();
List<Manage_Category__c> categoryList = [SELECT Id,Name
FROM Manage_Category__c];
options.add(new SelectOption(NONE_VALUE, NONE_VALUE));
if(categoryList.size() > COMPARE_INT){           for
(Manage_Category__c cat : categoryList) {
options.add(new SelectOption(cat.Id, cat.Name));
}
}
return options;
}

//Method to get All brands
public List<SelectOption> getBrand () {
List<SelectOption> options = new List<SelectOption>();
List<Brand__c> brandList = [SELECT Id,Name FROM brand__c];
options.add(new SelectOption(NONE_VALUE, NONE_VALUE));
if(brandList.size() > COMPARE_INT){           for (Brand__c cat :
brandList) {           options.add(new SelectOption(cat.Id,
cat.Name));
}
}
return options;
}

//Method to get All suppliers
public List<SelectOption> getSupplier () {
List<SelectOption> options = new List<SelectOption>();
List<Manage_Supplier__c> supplierList = [SELECT Id,Name FROM Manage_Supplier__c];
options.add(new SelectOption(NONE_VALUE, NONE_VALUE));           if(supplierList.size() >
COMPARE_INT){

```

```

        for (Manage_Supplier__c cat : supplierList) {
            options.add(new SelectOption(cat.Id, cat.Name));
        }
    }
    return options;
}

//Method to Save Product    public
PageReference saveProduct () {

manageProdVar = new Manage_Product__c();    if
    (suppliers != NONE_VALUE) {
        manageProdVar.Supplier__c = suppliers ;
    }

    if (categories != NONE_VALUE) {
        manageProdVar.Category__c = categories ;
    }

    if (brands != NONE_VALUE) {
        manageProdVar.Brand__c = brands ;
    }

manageProdVar.Unit_Price__c = unitPrice;    manageProdVar.Name
    = productName;    manageProdVar.Supplier_s_Price__c =
supplierPrice;    manageProdVar.Total_quantity__c = quantity;
try {
        insert manageProdVar;
    } catch (Exception e) {
        ApexPages.Message myMsg = new ApexPages.Message(ApexPages.Severity.ERROR, 'error');
    }

PageReference pageRef = new PageReference('/'+ manageProdVar.id);    pageRef.setRedirect(True);

    if(manageProdVar.id == Null){
        return null;
    }else {
createOrUpdateStockForProduct (manageProdVar.id);    return
        pageRef;
    }
}

//Method to Add and update stock Records    public void
createOrUpdateStockForProduct (Id ProductId) {

    List<Manage_Stock__c> listManageStock = [SELECT Id,NAME,Product_Brand__c,Product_Name__c,
        Avaible_Quantity__c FROM Manage_Stock__c
        ];

    List<Manage_Product__c> productVarList = [SELECT Id,Name,Brand__c,Total_quantity__c,Brand__r.name , Category__c
        FROM Manage_Product__c
        ];

```

```
Manage_Product__c productVar = [SELECT Id,Name,Brand__c,Total_quantity__c,Brand__r.name , Category__c  
FROM Manage_Product__c
```



```

WHERE Id =: ProductId];

Integer incrementProd = 0;

if (productVarList.size() > COMPARE_INT) {
    for (Manage_Product__c varProd : productVarList) {
        if(varProd.Name == productVar.name && productVar.Brand__r.name == productVar.Brand__r.name){
            incrementProd += Integer.valueOf(varProd.Total_quantity__c);
        }
    }
}

Manage_Stock__c stockObj;
Integer incrementQuantity = 0;
List<Manage_Stock__c> listTouupdate = new List<Manage_Stock__c> ();

if (listManageStock.size() > COMPARE_INT) {
    for (Manage_Stock__c stockVar : listManageStock) {
        if(stockVar.Product_Name__c == productVar.name && stockVar.Product_Brand__c == productVar.Brand__r.name){
            incrementQuantity ++;
            stockVar.Avaible_Quantity__c = incrementProd*incrementQuantity;
            listTouupdate.add(stockVar);
        }
    }
    update listTouupdate;

    if (incrementQuantity == COMPARE_INT) {
        stockObj =
        new Manage_Stock__c();
        stockObj.Product_Name__c =
        productVar.name;
        stockObj.Avaible_Quantity__c =
        productVar.Total_quantity__c;
        stockObj.Product_Brand__c =
        productVar.Brand__r.name;
        try{
            insert stockObj;
        }
        Catch(Exception e){
            System.debug('Inside maanage Product Controller : Line 143 : error' + e.getMessage());
        }
    }
}

}

}

}

public PageReference CancelProd () {
    PageReference pageRef = new
    PageReference('/apex/ProductTablePage');
    pageRef.setRedirect(True);
    return pageRef;
}
}

```

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Abstract

Although relational behavior plays a vital role in improving project performance in megaprojects, the intricate relational behavior among stakeholders has not been fully explored. Based on social exchange theory, this study presents a novel approach extended from the stakeholder value network (SVN) to identify the critical relational behavior and unveil the reciprocal multi-type relational behavior exchanges among the main stakeholders in megaprojects. Built upon six relational behavior types and 110 relational behavior flows, a qualitative relational behavior SVN (RB-SVN) model was first constructed. Questionnaire surveys from 224 megaproject managers were then used to quantify the RB-SVN model. By analyzing critical relational behavior types, flows, and cycles, we found that effective communication is the most significant relational behavior type for stakeholders. Owner–designer cooperation could create the largest relationship value through reciprocal relational behavior. The top 10 restricted (dyadic) and general exchanges were also identified to help promote stakeholders to engage in critical relational behavior cycles.

In this paper, an empirical analysis was conducted to test and measure the moderating effects of institutional pressures on the performance impacts of trust and relational norms. The results show that institutional pressures positively moderate the impacts of trust but negatively moderate the impacts of relational norms, indicating that trust and relational norms function and benefit projects differently and should be employed differently contingent on the institutional environments.

This research contributes to the broad literature on relationship management, and more specifically, furthers the understanding of relational behavior networks that strengthen the relationship value of multiple stakeholders in megaprojects.

## 2.2 Introduction

While emerging economies are receiving more attention than ever from multinational corporations (MNCs), the institutional environments in emerging countries are also posing more challenges than ever [1]. Particularly, emerging economies are often characterized by the weaker legal and political institutions and the stronger cultural and normative institutions that often dominate the legal systems and exhibit very distinctive characteristics that are unfamiliar to the modern world (Peng and Heath, 1996). Institutional pressures present when organizations need to either adapt to different norms, values, and belief systems (Suchman, 1995) or respond strategically to specific institutions (Oliver, 1991). Since transactional relationships are embedded in the context of the institutional environments, the governance of transactions is much affected by institutional pressures. Given such, a key issue is “*how do international firms use appropriate governing strategies when conducting projects or business under different levels of institutional pressures?*” To answer this question, while the default contractual governance

emphasizing the strict design and application of contract terms (Ning, 2018), Transaction Cost Economics (TCE) (as represented by Williamson (1985)) argues and shows that “market” or “contractual governance,” meaning transacting in markets through contracts, could have serious hold-up problems and opportunism (i.e., transaction cost) under uncertainty, information asymmetry, and transaction-specific investments. Internalizing transactions into corporate “hierarchy” is an effective means to solve the opportunism in “market” but often too expensive for most transactions. In this regard, many studies began to focus on “relational governance” as another type of governance to mitigate the contractual hazards or opportunism (Macneil, 1985; Noordewier, John, Nevin, 1990; Uzzi, 1997).

## 2.3 Related Work

Megaprojects include multiple temporary teams of actors adapting to diverse demands and working independently, tending to fail when they are not well collaborated. Effective relational behavior can promote the success of the projects by establishing collaborations and enhancing mutual understanding; whereas the lack of relational behavior hinders realization of desired project outcomes, especially in the challenging and complex megaproject settings (Dekker, Donada, Mothe & Nogatchewsky, 2019; Nasir & Hadikusumo, 2019). Although the

extensive use and practice of relational behavior in megaprojects is crucial, it is unavoidable that participating organizations mainly view issues from their own perspectives, leading to differences in their judgments, ranking, and commitment to given stakes (Pesämaa, Larsson & Eriksson, 2018). For instance, owners often pursue time, cost, and quality goals, whereas contractors focus on immediate profits and the potential to renew contracts in future projects. Such differences make individual organizations have different adoption intentions and implementation priorities for various relational behavior types. Therefore, understanding relational behavior from a multi-stakeholder perspective and revealing the exchange process between the stakeholders based on their needs would help drive the adoption of relational behavior and achieve more effective megaproject management practices.

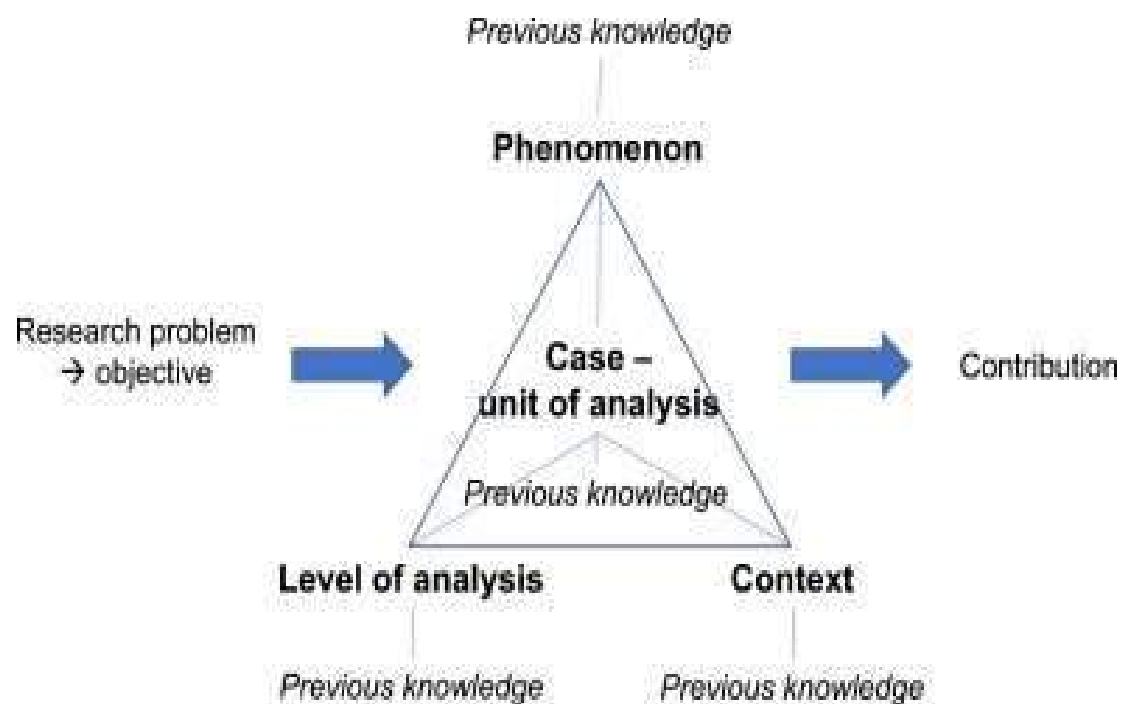
## **2.4 Design(GUI, analysis, implementation, evaluation)**

One obvious feature of a case study is that it investigates a phenomenon in its natural context, in a real-life setting. It typically deals with a contemporary event where the researchers are not attempting to manipulate the behavior being investigated (Yin, 2009). Yin (2009) also highlights that the boundaries between the phenomenon and its context are not necessarily evident, and researchers purposely want to investigate both of them because the context may be highly relevant to the phenomenon being studied. This attempt to study a phenomenon in its natural setting is both attractive and challenging

(Vossetal., 2002), and it can be considered quite suitable and beneficial for research in project studies. As project management is strongly influenced by social and behavioral issues and by the specific context of the project, case studies represent an extremely potent research design option in such a field. In project studies (as in any other research field), there are usually two distinct moments in which the researchers choose the research design. The first and more dramatically influential moment is when the researchers are preparing for a new research project and decide to engage in a certain type of research. At that moment, the researchers usually identify a need or a research problem. They then specify the key goals and research questions for the study and determine how they will carry out the study. The researchers may choose to work with certain types of organizations or networks and collect data on certain projects, programs, or processes from individuals, documents, or databases.

Case study researchers must of course know what a case is. The choice of the case(s) is significant not only for the empirical study but also for the literature review, so that the latter would be able to justify and frame the empirical study. Many paper submissions erroneously regard the organization as the case (and

refer to it as the case company) but end up reporting findings about a process, a project, a certain type of manager, a practice, or something else as the case (here, the organization can be referred to as the focal or target company instead of the case company).

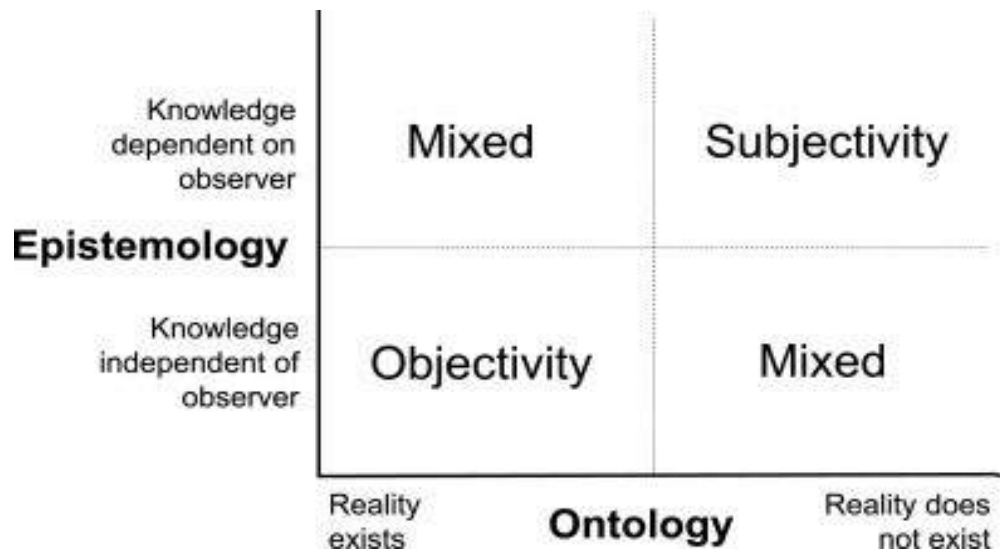


Research philosophy is considered a combination of two dimensions: ontology or the “nature of reality” and epistemology or the “nature of knowledge”

( [Mayloretal., 2017](#) ). While ontology spans a continuum of beliefs from “social

Reality exists independently” to “social reality does not exist at all,”

epistemology spans a continuum of beliefs regarding the researcher's role in the knowledge creation process, ranging from being only an independent observer.



## 2.5 Results & Discussion

The sample mean and standard deviation (SD) of each variable are checked for their reasonableness and possible serious issues. We find that the variables' statistics should be within reasonable ranges based on our knowledge in the international construction sector. It is also worth our attention that the sample mean of Performance is 3.73, that of Trust 3.32, and that of Relationship 4.08, which indicates that relational governance is commonly used in our sample projects.

Variable	Mean	SD	1	2	3	4	5	6	7	8	9	10
1. Performance	3.73	0.7	1									
2. Building	0.56	0.5	0.002	1								
3. Korean	0.48	0.5	-0.36		1							
4. BudLarge	0.26	0.4	-0.14			1						

5. BudSmall		0.47		0.5	.250	-.121				
6. Trust		3.32		0.7	.332	.201	.036			
7.Relationship		4.08		0.6	.510	.043	-.37	-.04		
.289	.315									
8.InstPressure	2.78	1.0	-.34	.149	.579	.200	-.35	.10	-.22	1
9.InstPressure*Trust		-.10	.219	.466	.133	-.211	.56	-.02	.85	
1										
10.InstPressure*Relation		-.110	.175	.397	.176	-.22	.23	.255	.87	
.83	1									

Sample size n=165.

To ensure the stability of the regression analysis, variance inflation factors (VIFs) were calculated to identify the potential collinearity. The maximum VIF obtained in our models was substantially below 3, above which multicollinearity would be considered high (Hair, Page, Brunsveld, 2020) A potential issue of the OLS regression is due to the homoscedasticity assumption. In the case of heteroskedasticity, the statistical significance of independent variables may be overestimated. In this study, the White test was performed to examine whether the sample's homoscedasticity assumption was rejected. The p-value of the White test was 0.0001, indicating that there was a heteroskedasticity problem.

## 2.6 Conclusion

Because of the institutional pressures in emerging countries and the incompleteness of contractual governance, it is of great interest and importance for international firms to know “*how do international firms use trust and relational norms, functioning as different types of relational governance, when conducting projects under different levels of institutional pressures?*” This objective is achieved by measuring how institutional pressures moderate the impacts of trust and that of relational norms. An empirical assessment was conducted to test the proposed hypotheses and evaluate the magnitudes of the hypothesized impacts. Our empirical results show that, while trust and reciprocal relationship have positive impacts on project performance, institutional pressures positively moderate the performance impacts of Trust but *negatively* moderate the performance impacts of a reciprocal relationship.



## 2.7 References

1. Peng and Heath,(1996) .
2. Suchman,(1995).
3. Oliver,(1991).
4. Ning, (2018) ,
5. Williamson (1985).
6. Macneil, (1985) .
7. Noordewier, John, Nevin, (1990).
8. Uzzi, (1997) .
9. Dekker, Donada, Mothe & Nogatchewsky,(2019)
10. Nasir & Hadikusumo, (2019) .
11. Pesämaa, Larsson & Eriksson, (2018).
12. Yin, (2009) .
13. Vossetal., (2002).
14. Mayloretal., (2017) .
15. Hair, Page, Brunsveld, (2020).