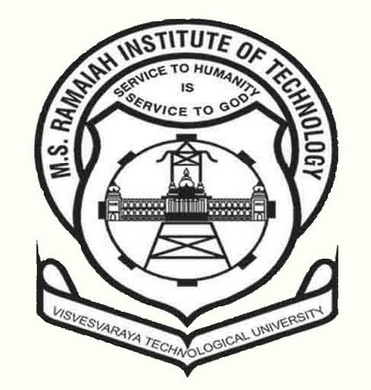
**PRODUCT MASTER UI**



Department of Computer Science and Engineering

M.S.Ramaiah Institute of Technology

(Autonomous Institute Affiliated to VTU)

Bangalore-560054

Under the Guidance of: Submitted by:

Ms. Meera Devi

Associate Professor, 1MS10CS002

Department of Computer Science Aanchal Jain

**LITERATURE SURVEY**

**SALESFORCE:**

* Salesforce is Customer Relationship Management Software that is maintained on the cloud. Its follows an RDBMS structure where every table is an object and every record is an instance of the object.
* Apex is an additional feature that developers can use to implement custom functionality for Salesforce to write web services.
* Salesforce gives us the ability to create multiple copies of our organization in separate environments for a variety of purposes, such as testing and training, without compromising the data and applications in our Salesforce production organization. These copies are called sandboxes.
* The sandbox types are:

Developer Sandbox

* + Developer sandboxes are special configuration sandboxes intended for coding and testing by a single developer. Multiple users can log into a single Developer sandbox, but their primary purpose is to provide an environment in which changes under active development can be isolated until they’re ready to be shared.

Developer Pro Sandbox

* + Developer Pro sandboxes copy all of our production organization's reports, dashboards, price books, products, apps, and customizations under Setup, but exclude all of our organization's standard and custom object records, documents, and attachments.

Partial Data Sandbox

* + Partial Data sandboxes include all of our organization’s metadata and add a selected amount of our production organization's data that we define using a sandbox template. A Partial Data sandbox is a Developer sandbox plus the data we define in a sandbox template. It includes the reports, dashboards, price books, products, apps, and customizations under Setup (including all of our metadata).

Full Sandbox

* + Full sandboxes copy our entire production organization and all its data, including standard and custom object records, documents, and attachments.

Sandbox templates allow us to pick specific objects and data to copy to our sandbox, so we can control the size and content of each sandbox. Sandbox templates are only available for Partial Data or Full sandboxes.

* Salesforce supports a variety of toolkits. Currently, AJAX toolkit has been used for this project. The AJAX Toolkit is a JavaScript wrapper around the API. The purpose of choosing AJAX toolkit is:

 The AJAX Toolkit is available for any organization that has API access.

 we can execute any call in the API, and access any API object that we normally have access to.

 we can issue asynchronous calls, and use callback functions to handle the results.

 we can use header options with a different syntax than in the API.

 we can handle errors with the AJAX Toolkit.

 The AJAX Toolkit supports relationship queries.

**APTTUS:**

Apttus was formed to deliver the promise of enterprise class application capability, but very rapidly and with high value. Apttus has combined business domain expertise with the Salesforce and provides modules with rich functionality. Apttus’ goal is to provide organizations with an alternative and significantly better way to automate business functions in a way that maps more closely to the needs of organizations.

Apttus delivers Software as a Service (SaaS) applications, allowing us to rapidly and simply deploy enterprise class applications. The functionality allows us to gain complete visibility and end to end management of our business function. The Suites are “out of the box” allowing for extremely rapid deployment. Additionally, configuration tools allow for fast and easy extensibility. We can begin to use the applications immediately, but can change them as we learn more about what we need.

Apttus utilizes Salesforce and couples this with deep domain expertise. The result – we gain applications that are very functionally deep, running off a platform used by over millions of users globally.

**ANGULAR JAVASCRIPT:**

AngularJS is an open-source JavaScript framework, maintained by Google, that assists with running single-page applications. Its goal is to augment web-based applications with model–view–controller (MVC) capability, in an effort to make both development and testing easier.

The library reads in HTML that contains additional custom tag attributes; it then obeys the directives in those custom attributes, and binds input or output parts of the page to a model represented by standard JavaScript variables. The values of those JavaScript variables can be manually set, or retrieved from static or dynamic JSON (JavaScript Object Notation) resources.

Angular is what HTML would have been had it been designed for applications. HTML is a great declarative language for static documents. It does not contain much in the way of creating applications

Angular attempts to minimize the impedance mismatch between documents centric HTML and what an application needs by creating new HTML constructs. Angular teaches the browser new syntax through a construct called directives.

AngularJS follows an M-V-V-M architectural pattern. MVVM facilitates a clear separation of the development of the graphical user interface (as markup language) from the development of the business logic or back end logic known as the model (also known as the data model to distinguish it from the view model). Any change in the model is reflected in the view and any change in the view is reflected in the model.

Angular frees us from the following pains:

* Registering callbacks: Registering callbacks clutters code, reducing readability. Removing common code such as callbacks is a good thing. It vastly reduces the amount of JavaScript coding *we* have to do, and it makes it easier to see what our application does.
* Manipulating HTML DOM programmatically: Manipulating HTML DOM is a cornerstone of AJAX applications, but it's cumbersome and error-prone. By declaratively describing how the UI should change as our application state changes, we are free from low-level DOM manipulation tasks. Most applications written with Angular never have to programmatically manipulate the DOM, although we can if we want to.
* Marshaling data to and from the UI: CRUD operations make up the majority of AJAX applications' tasks. The flow of marshaling data from the server to an internal object to an HTML form, allowing users to modify the form, validating the form, displaying validation errors, returning to an internal model, and then back to the server, creates a lot of redundant and unwanted code. Angular eliminates almost all of this, leaving code that describes the overall flow of the application rather than all of the implementation details.

Few appealing features about AngularJS are:

* Decoupling DOM manipulation from app logic: This improves the testability of the code
* Supports Bootstrap: Bootstrap is a highly responsive styling language. It allows us to improve the appearance of the HTML pages.
* Decoupling client and server sides of an app: This allows development work to progress in parallel, and allows for reuse of both sides.

**BOOTSTRAP:**

Bootstrap is a sleek, intuitive, and powerful front-end framework for faster and easier web development. It is currently the most used pre-fabricated CSS style sheet framework in existence today. Even though it is heavy, it provides a lot of advantages. It has been chosen to serve the following purposes:

* Save Time: A web developer doesn’t have to spend time laboriously working out and writing code, but simply find the right piece and fit it into the structure they’re working on. In addition to this, a lot of the styling and design aspects are already taken care of.
* Customizable: A great aspect of Bootstrap is that we can make it our own. We can keep what we need and leave what we don’t. Bootstrap lets us factor in our needs efficiently
* Great grid system: Bootstrap is built on responsive 12-column grids, layouts and components. Whether we need a fixed grid or a responsive, its only matter of a few changes. Offsetting & Nesting of columns is also possible
* Consistency: Bootstrap’s results are uniform across platforms. It shows the same thing on Internet Explorer, Chrome and Firefox.
* Integration: Integration with existing code is simple, fast and easy to accomplish and once we’re done we can play with our design to our heart’s content.
* Responsiveness: Bootstrap is responsive across devices (whether it's a laptop or mac or ipad). Bootstrap adapts to the change in platforms with super speed and efficiency.
* Future Compatibility: Bootstrap comes equipped with many elements that are being considered the future of design itself such as HTML5 and CSS3.

**JSTREE:**

JsTree is jQuery plugin that provides interactive trees. It is absolutely free and supports all modern browsers. jsTree can display trees by parsing HTML or JSON and supports AJAX, it is themeable and easy to configure and customize. Events are fired when the user interacts with the tree. Other notable features are inline editing, drag'n'drop support, fuzzy searching (with optional server side calls), tri-state checkbox support, configurable node types, AMD compatibility, easily extendable via plugins

JsTree functions properly in either box-model (content-box or border-box), can be loaded as an AMD module, and has a built in mobile theme for responsive design that can easily be customized. It uses jQuery's event system, so binding callbacks on various events in the tree is familiar and easy.

Few of the features worth noting are:

* drag & drop support
* Optional keyboard navigation
* inline edit, create and delete
* tri-state checkboxes
* fuzzy searching
* customizable node types
* Various data sources - HTML, JSON, XML
* Supports AJAX loading
* Highly configurable
* Theme support + included themes
* Numerous callbacks to attach to
* Maintain the same tree in many languages
* Open/close optional animation
* Configurable multitree drag & drop
* Optional multiple select
* Supports plugins & datastores
* Optional state saving using cookies