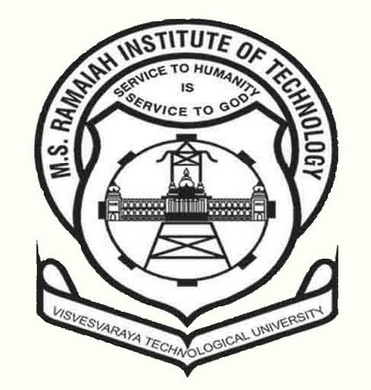
**PROJECT SYNOPSIS**



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

**PRODUCT MASTER**

**INTRODUCTION**

Akamai Technologies is an Internet content delivery network. It provides services to companies that have content on the Internet (Akamai's customers), to deliver this content to users browsing the Web and downloading content. It has developed a variety of products for its customer companies.

Initially these product details were maintained in an excel document and the customer contracts were manually handwritten. For obvious reasons this wasn’t a pragmatic approach.

**EXISTING SYSTEM**

A new system was built where in all the product details were maintained in something called the product catalogue and an ordering system called the momentum was built to convert quotes to orders. Seibel, a Customer Relationship Management tool was used to generate the final contract delivered to the customer. It kept an on-premise log of the customer and contract details. The product catalogue stores the product data in this format:

Solution -> Package -> Product -> Module -> Sub-Module -> Service-Feature

Where the relationship between each level is that of aggregation. i.e. a solution is a group of packages, a package is a group of products, a product is a collection of modules and so on..

The project catalogue, momentum and Seibel posed the following issues:

* Platform issues: Catalog sits on the portal database on premise
* UI issues: Launch item management UI is technical and not friendly for Product Managers for launching new products
* Package Submission errors: hard to debug (an error in 1 product in a package, in turn holds up the submission of the other products in that package)
* Data model Issues: All catalog concepts are modeled using database associations, requiring extensive data duplication and maintenance work
* Highly Complex Structure.

**PROPOSED SOLUTION**

To over come these issues, it was planned to replace ‘Product Catalogue’ by ‘Product Master’, ‘Momentum’ by ‘Q2O’ (Quote to Order) and to use ‘Salesforce’ instead of Seibel. All of these are currently under development process.

The new rationalized structure of the product master is:

Product -> Option Groups -> Options -> Features

* Product: Product is a primary unit of sale, which by itself is of no value unless certain mandatory and optional options are attached to it.
* Option groups: An option group is an abstraction of options. Products has options groups through which options are attached to Product
* Options: Options have features and Pricing used to customize products. They can be auto included and excluded using rules. They are re-usable across various products
* Features: The set of features under option are not context dependent (they don’t change based on the product under which option is sold) Features can have multiple engineering products

Engineering products are context dependent. This Context is based on 2 things:

* The Option to which the Feature is attached
* The Product under which the Option is being sold

**IMPLEMENTATION METHODOLOGY**

A two-layer Model-View-Controller structure is used in client server architecture to implement this. One layer of M-V-C is implemented on the client side and the next layer is implemented across the client-server. The client here refers to the User Interface and the server refers to the Salesforce cloud.

**SOFTWARE REQUIREMENTS**

The technologies used on the client side are:

* Hyper Text Markup Language (HTML)- for the UI
* Bootstrap (BS)- for styling. (It’s used instead of CSS to overcome the non-responsive nature of CSS)
* JavaScript (JS)- used for dynamic web pages
* Angular JavaScript (AJS)- for the controllers. It’s an M-V-V-M framework, i.e. updating the model automatically updates the view and updating the view automatically updates the model.
* JQuery- to avoid the cross browser discrepancies

The technologies used on the server side (Salesforce) are:

* Apex- (a derivative of Java) to code the modeling of data
* VisualForce- to view the data
* Salesforce Object Query Language (SOQL)- for data querying

**HARDWARE REQUIREMENTS**

The hardware requirements of the project are:

* Windows or Mac OS
* Memory on Salesforce cloud

**MY CONTRIBUTION**

My team is currently working on the project- ‘Product Master’. My contribution to the project will be a part in the UI where I’ll be working in Angular JavaScript, Bootstrap and JavaScript, in coordination with Salesforce to access the data