Milestone 2 Report

CS5150: Software Engineering
Cornell University
Ithaca, NY

Under the guidance of

Prof. William Arms

Department of Computer Science

Clients

Rajiv Kumar, Product Owner

Abhijnan Saha, Architect

Ayush Soni (aas427), Harshit Manchanda (hm545),
Mukul Shukla (ms3528), Rohith MD (rm977), Shubham Agrawal (sa2279),
Shubham Khandelwal (sk3266), Vidush Vishwanath (vv259)

Table of Contents

Introduction

The Bills of Material (BOM) topic is one of the core topics of Product Lifecycle Management area at SAP. The BOM consists of a unique header defining the product and a list of items used to create that header. An example is as follows: {Coffee Cup: Plastic lid, green plastic straw, plastic transparent cup, etc.}.

Our algorithm will advance this framework by taking in the history of customer data and various other features (cost, texture, color, availability, region, etc.) and finally suggesting the best material that could be used to replace the given material by employing Machine Learning. If the preceding functionality is complete, we will add functionality to suggest next best materials on similar lines. There would also be a user interface, but this would be for demonstration purposes only.

An agile model of software development is being followed, and the whole project would be a **single sprint** with **three iterations**.

The project in all, aims to incorporate the principles of Software Engineering while using other tools and technologies involving machine learning, data processing and wrangling, principle component analysis and web technologies.

The existing solution :

a) Considers only 3 parameters: Stock, Availability and Location
b) Naive Implementation with basic selects and manual logic for best recommendation
c) No Machine Learning techniques incorporated for premium recommendation
d) Not scalable and on premise solutions per industry

Scope

- Build a recommendation system for the replace materials in BOMs using a machine learning approach
-
  Approach includes feature engineering and evaluating against test data

- Experiment with different machine learning algorithms such as Logistic Regression, Naive Bayes, Support Vector Machines, Bagged Decision Trees and Deep Learning.

The scope does not involve making of a User Interface or building a database but just requires that the new APIs should be compatible with the client's proprietary cloud and database.

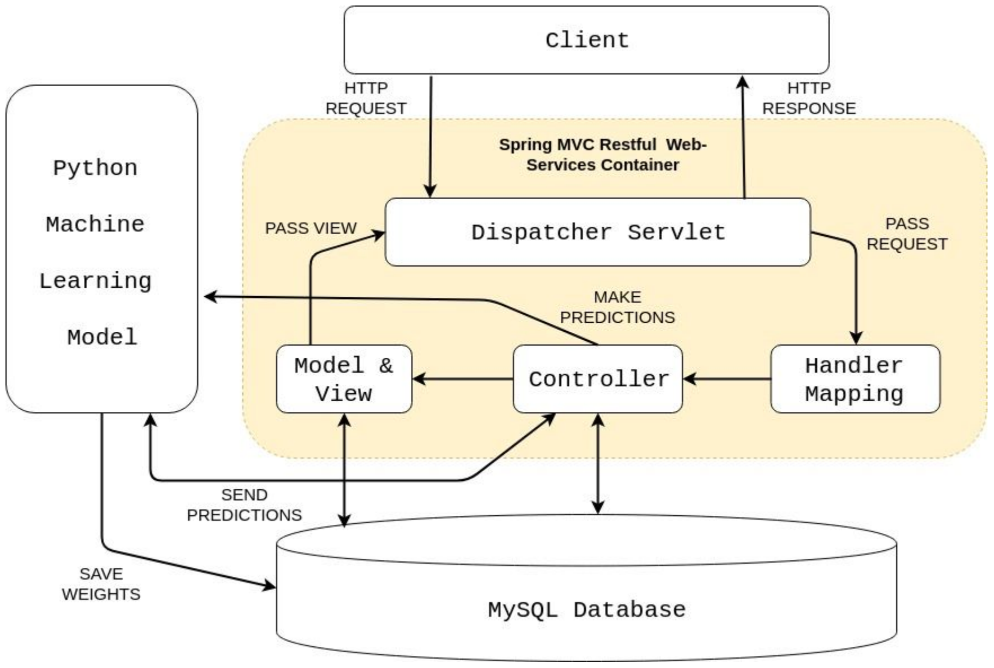
Requirement Analysis

Most of the requirement analysis was performed in the feasibility report and is provided there, including functional and nonfunctional reports. Apart from those requirements, the following were added upon discussion and clarification with the client.

- The system should be able to suggest recommendations that may be a suitable alternative.
- The system should help the user complement his domain expertise in choosing the best replacement.
- Further detail included in feasibility report

Provisional Design

Overall System Design - MVC



The above system shows the overall architecture of the system.
Each of the rectangular box in the figure above shows an independent stand alone component with the arrows showing the transfer of data between the different components to create the pipeline flow.
Let us take an example for a particular use case where the user needs to get similar recommendations for a material to be searched.

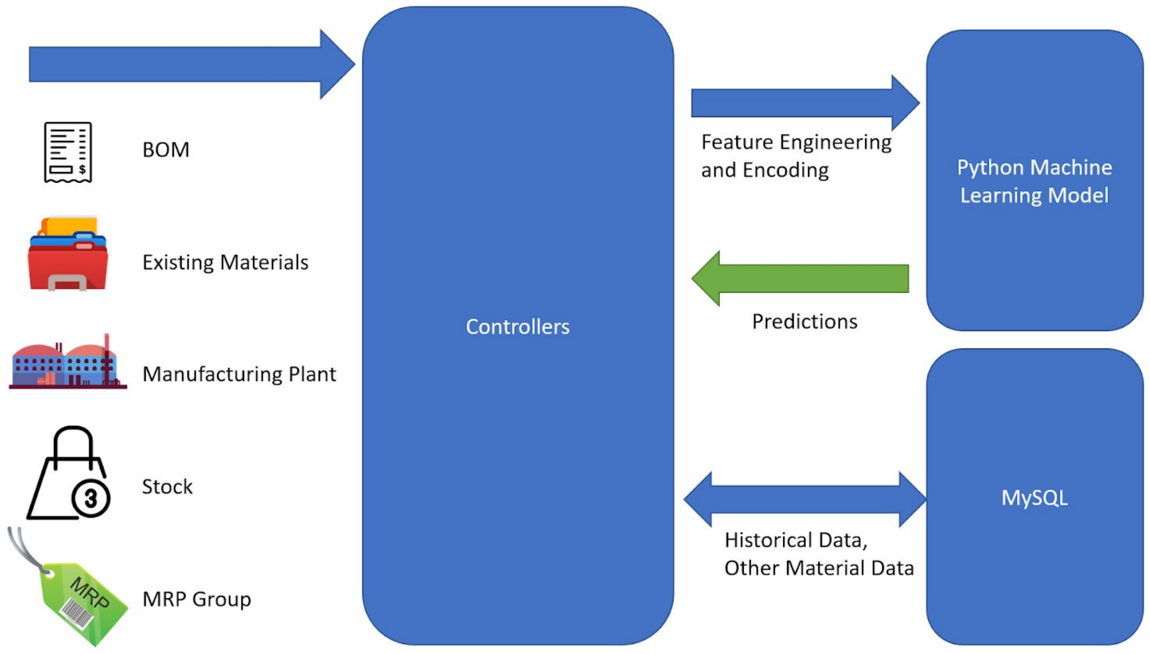
Overall Pipeline:

1) User Enters Text and clicks on the search button (extra filters for “Plant/Manufacturer”, “Max number of recommendations” can be added)
2) A request is made to Server and a corresponding DAO Call is made (Fetch availability, material, plant and properties and most importantly **replacements and frequencies**)
3) Apply the filters passed in the UI
4) Fetched Material properties will be mapped to feature weights (populated from the last batch run of ML models)
5) Dot product with feature weights of possible replacements and the current material searched will give a result
6) The descending order of ratings would be given as output on the display

Model

The model comprises of the data modelling and storage formats we’ll be using to develop the overall system. We are using Microsoft Azure’s Blob storage for storing the data in a MySQL database. The data has the same schema as the one used in the client’s proprietary database with an extra intermediate table for Feature weights to be maintained further. We have discussed and described the entire storage model and schema along with the Entity Relationship diagram in the following sections.

Controller


BOM
Existing Materials
Manufacturing Plant
Stock
MRP Group
Controllers
Feature Engineering and Encoding
Python Machine Learning Model
Predictions
MySQL
Historical Data, Other Material Data

From the controller's perspective, we shall describe the working by looking at an example. Assume that the user wants to search for BOM replacement where he wants only those replacement options where the stock available is more than a 100 units:

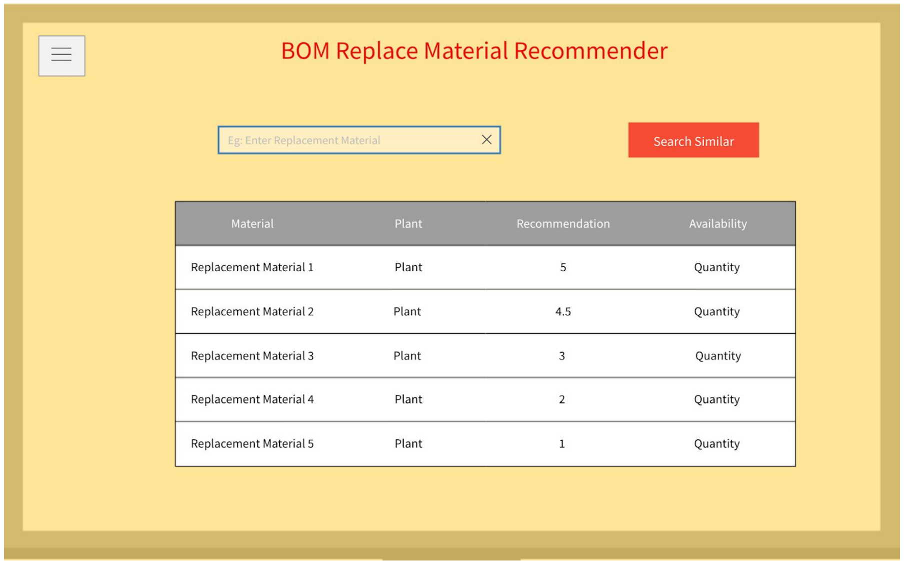
1. The user would enter the BOM material (mandatory) along with any filters he wants to set such as manufacturing plant, or stock available. In our case, he would enter the BOM material along with minimum stock amount he requires for any recommendation to be valid.
2. The request on reaching the controller, the controller asks for other material data along with any historical data, and then requests the ML model to provide recommendations for a given set of features.
3. The ML model would subsequently send the recommendations list back to the controller which is then passed on to the view.

This is the functioning described from the controller's perspective.

View

Upon discussion with the client, it was decided that our UI would need to provide several primary features. First, a search box where the user can enter a BOM name to search by. Next, a search button. And finally, the results which must indicate the given material, the rating for the material, availability and pricing, and manufacturer. A prototype for this view is provided below and was approved by the client during the presentation. As can be seen, the system will ultimately provide a list of recommendations to the client and the client may make the end decision about which replacement to use based on parameters such as the plant of manufacture and the availability of the given item.

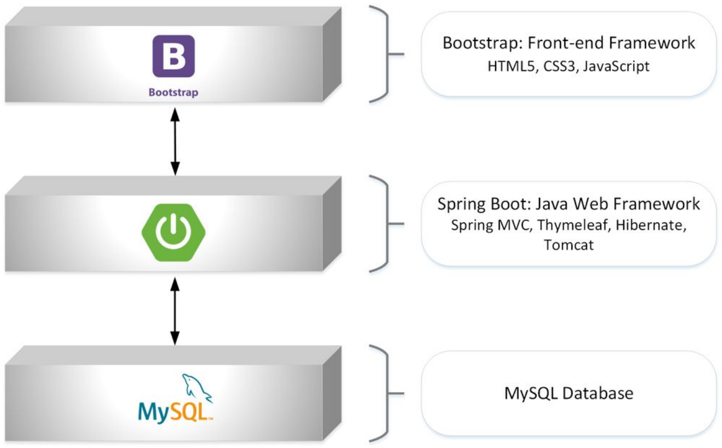
Prototype



BOM Replace Material Recommender

Material	Plant	Recommendation	Availability
Replacement Material 1	Plant	5	Quantity
Replacement Material 2	Plant	4.5	Quantity
Replacement Material 3	Plant	3	Quantity
Replacement Material 4	Plant	2	Quantity
Replacement Material 5	Plant	1	Quantity

Technology Stack



Bootstrap
Bootstrap: Front-end Framework
HTML5, CSS3, JavaScript
Spring Boot: Java Web Framework
Spring MVC, Thymeleaf, Hibernate, Tomcat
MySQL
MySQL Database

The stack we will be using is bootstrap on the UI, followed by spring-hibernate-tomat backend and a MySQL database.

We use bootstrap because its unique and compelling features empower the developers to build well-functioning, fully responsive websites without any hassle.

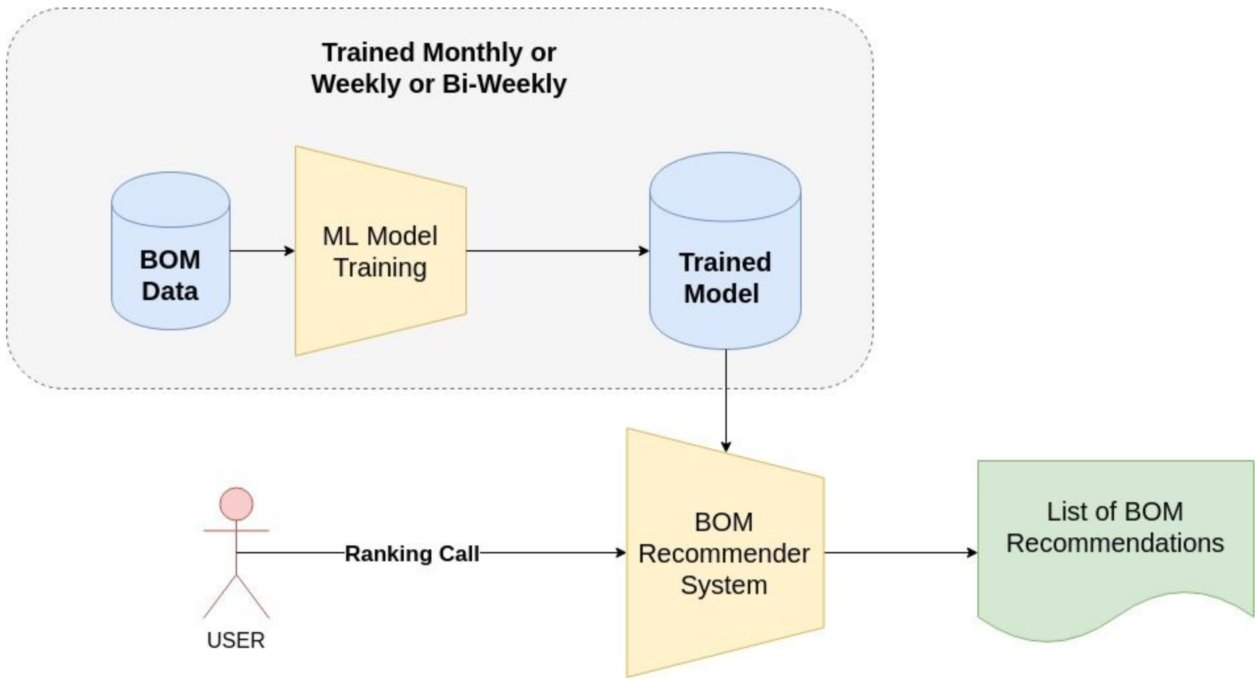
Websites built with this technique not only automatically adapt to whatever screen resolution users browse on, but they also require the least resizing, panning, and scrolling – delivering users an optimal viewing and interaction experience across different devices and screen sizes.

Spring is a framework that helps us "wire" different components together. It is most useful in cases where we have a lot of components and we might decide to combine them in different ways, or we wish to make it easy to swap out one component for another depending on different settings or environments.

Hibernate is ORM framework in Java. ORM means Object Relational Mapping, it’s used to map the Java object (its data) with the database tables’ row data and vice versa.
MySQL is an open source relational database system. It helps us quickly realize the schema given to us by SAP and is compatible with hibernate (the ORM we decided on).

Machine Learning Component:

Machine Learning System Design

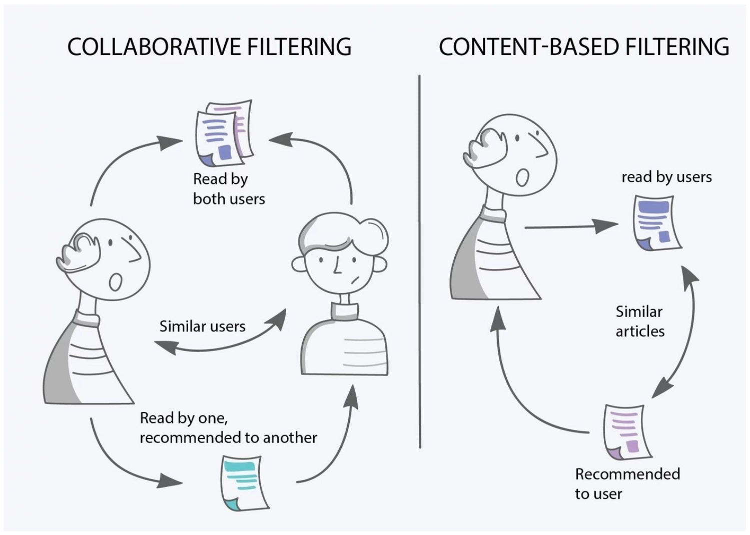


The machine learning system is designed to have two major components:

- The first part of the system has the trained model weights stored in a period of frequent intervals. BOM data stored is pulled by the ML model and then the model is trained on it, storing the results in a data store
- The second part of the system comes into action when a ranking call is made by the user. The user does this by pressing the ‘Search Similar’ button on the user interface. This triggers the recommender system to pull the trained model and find the recommendations, ultimately giving out a list

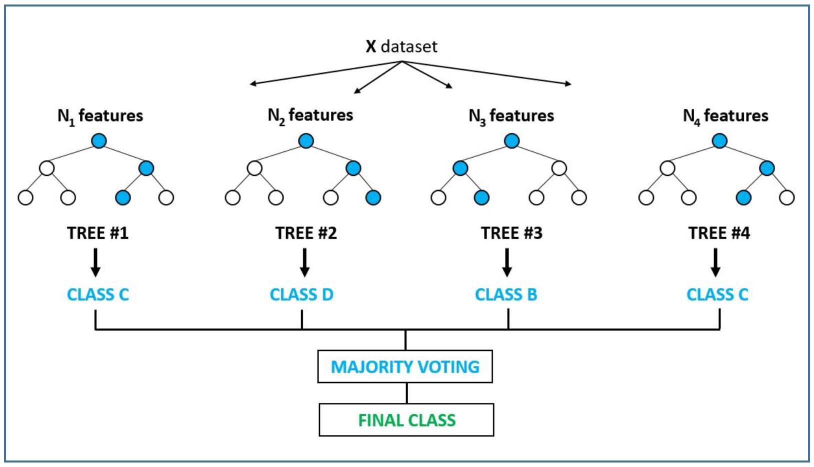
Algorithms:

- Collaborative Filtering (Baseline)



1. Collaborative filtering is of two types : user-user recommendation and user-item(or item-item) recommendation.
2. In BOM, user-user recommendation would stand for finding most similar merchants/industries while replacing materials and suggesting new replacements based on history of the most similar merchant.
3. In BOM, user-item recommendation would stand for making most similar predictions based on most similar items to the ones which were replaced in the past.
4. Similarity is calculated using cosine similarity and items/users are ranked based on the calculated similarity score.

- Random Forest



Random Forest algorithm involves three major steps:

1. Bootstrap Sampling: Sampling a dataset multiple times with replacement to form N different datasets. This helps in fighting bias in models. Once we complete this step, we move on to training of the model, i.e. making of decision trees.
2. Training Models: The decision trees form by using the metric of information gain - how much a feature reduces the entropy in the subtrees. There is one tree for each different data set, and each data set could lead to a different splitting than the other trees.
3. Bootstrap Aggregating: Finally the aggregating step occurs where a sample input is run through all the trees and a particular class is obtained for each. Once we do this, we take the majority voting to find the final class of the particular element.

Data Modelling:

Bills of Material:

A **bill of materials** or **product structure** (sometimes **bill of material**, **BOM** or **associated list**) is a list of the raw materials, sub-assemblies, intermediate assemblies, sub-components, parts, and the quantities of each needed to manufacture an end product.

In the process of creating a BOM, fIrst the materials or items for the BOM are created and added to their tables then the BOM is created.Therefore, BOM represents a structure of a Product and its constituents.

BOM for a Car:

HEADER: Car
ITEMS:

- Chassis
- Engine
- Battery
- Wheels

As we cannot have the proprietary database from SAP, in consultation with the client we have built up the schema based upon the original data schema and used the field that are present in the actual database to make the integration into the final system easy.

We have 6 major tables (from proprietary database):

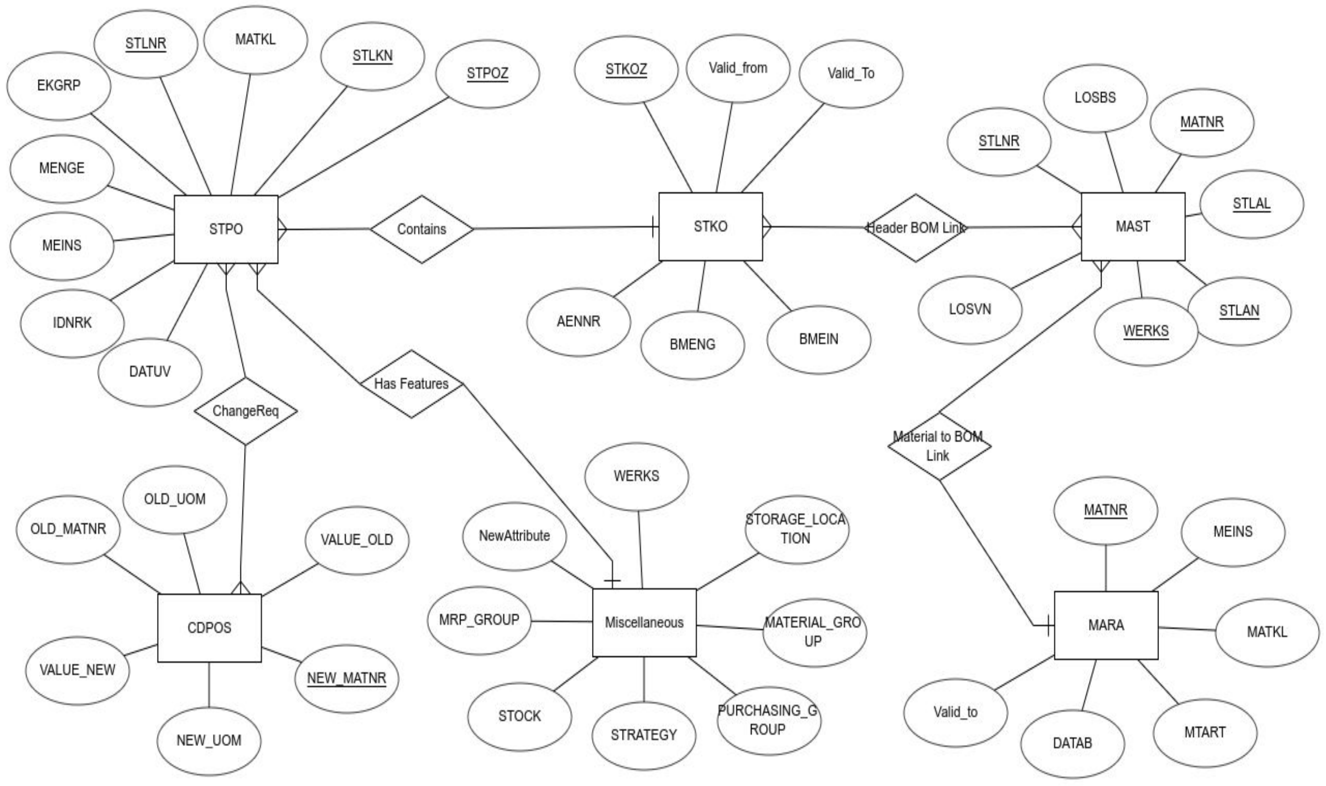
- **MARA**: This represents the material table and its properties. Every material like Car, Chassis, Engine, Battery are included in this table along with their specific attributes.

- **MAST**: Relation between material and Bills of Material (BOM). Only headers are included in this table. This table is mainly used for our purpose for getting the corresponding identifiers for different BOMS (and their alternatives too).
- **STKO**:This table describes details only for the headers, the details include their identifiers, properties etc.
- **STPO**: Items (engine, chassis, etc.) & parameters for items
- **CDPOS**: Table is used for getting information about the BOM and its replacements. Basically a link between BOM and its replacements
- **Miscellaneous:** This table stores the MRP related attributes that contribute

We also have an intermediate table that id periodically updated by the ML model assigning weights to individual features.

- **Intermediate Table**:Feature weights table to be created and updated every time feature engineering is performed. (To be run periodically as part of the batch jobs).

ER Diagram:



A Snapshot of the tables is given below:
To have a clear idea of the table contents and the data in the field we have created a snapshot with exact data using the CAR Bills of Material as the example.

Table 1: MARA

MATNR	Material Name	Car
MEINS	Unit of measure	EA (each)
MATKL	Material group	000000001 (9 digit #)
MTART	Material Type	FERT (finished product)
DATAB	Valid from Date	02-02-2019
Valid_To	Valid to date	20-02-2019

Table 2: MAST

MATNR	Material Name	Car
STLNR	Bill of Material	000000005 (same as BOM)
STLAL	Alternative BOM	01
STLAN	BOM Usage	01
WERKS	Plant	0001
LOSVN	From lot size	0
LOSBS	To lot size	100

Table 3: STKO

STLNR	Bill of Material	000000005 (same as BOM)
STKOZ	Unique identifier for table (primary keys)	00000001
DATUV	Date from which item is valid	02-02-2019
Valid_To	Date until item is valid	05-02-2019
BMEIN	Unit of measure (EG kg or lb)	KG
BMENG	Quantity in unit of measure given by BMein	10
AENNR	Change #, for replacement	CHANGE_10

Table 4: STPO

STLNR	Bill of Materials	00000005
STLKN	BOM Item Node Number	00000034
STPOZ	Internal Counter	00000007
DATUV	Valid from date	05-02-2019
IDNRK	Name of Item	ENGINE
MEINS	Units of Measure	EA
MENGE	Quantity	10

EKGRP	Purchasing Group	00000001
MATKL	Material Group	00000001

Table 5: CDPOS

OLD_MATNR	Old BOM Material Number	VISIBLE_V8_ENGINE
OLD_UOM	Unit of measure	KG
VALUE_OLD	Quantity of old material	20
NEW_MATNR	BOM Material Number	HAYNES_V8_ENGINE
NEW_UOM	Unit of measure	KG
VALUE_NEW	Quantity of new material	20
AENNR	Change #, for replacement	CHANGE_10

Table 6: Miscellaneous

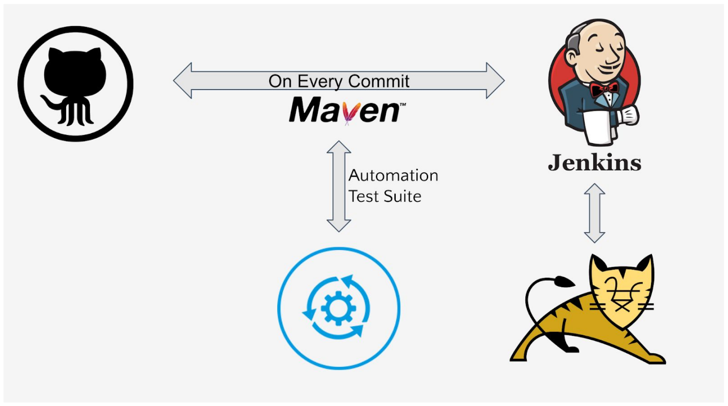
WERKS	Plant	0001
STORAGE_LOCATION	Place where item is stored	1010(Unique Location)
MATERIAL_GROUP	A particular group to which it belongs	00000001
PURCHASING_GROUP	A particular group to which it belongs while purchasing	0001
STRATEGY	Decision of Make-to-Stock/ Make-to-Order	10/20

STOCK	Available Quantity per plant	50
MRP_GROUP	A particular group of MRP to which it belongs	0001

Continuous Integration Testing and Version Control:

Commits to the project will be made using github wherein each team member has a branch by their individual name. This will allow team members to experiment with the code and get a good low level understanding. When team members feel they have a new feature to add or a bug to fix, they will submit a pull request, requesting that their commit(s) be merged in with master branch. Each pull request will have to be approved by at least one member of the team who did not initiate it.

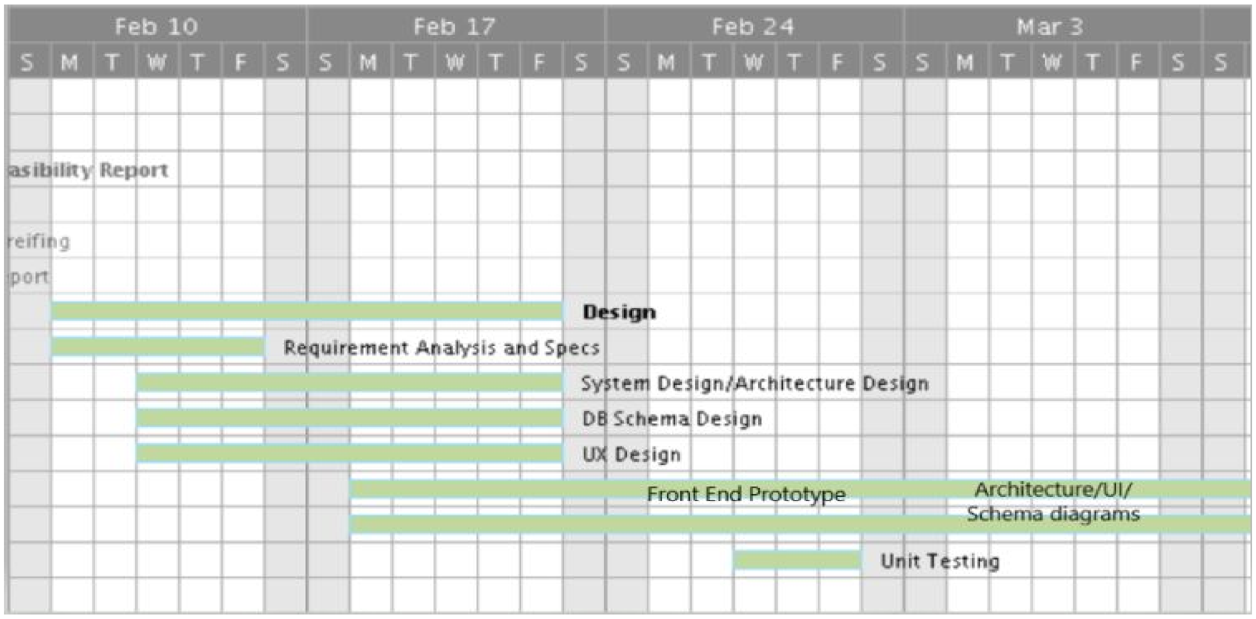
Every time a commit is made into the master branch, we will launch a series of tests to ensure that the overall architecture still works as expected.



The tools we will use for CI testing are Maven and Jenkins. On each commit, compilation of all existing code will be made through Maven, and Jenkins will be used to perform unit testing on functions as we see fit and deploy the model onto an Apache Tomcat server. Tomcat will ensure that we can see the development of our product as code additions are made.

Timeline:

The timeline between the submission of the feasibility report and milestone two is shown below.



As can be seen, all targets are being met. By the time of the presentation on March 7th, all items mentioned in the above timeline were completed and a few days of slack was there for the Architecture and schema diagrams tasks.