

# Final Modules List

## Functional Modules to be implemented for Bahmni

### 1. Bahmni\_atom\_feed:

**Dependency :** base, auth\_crypt, bahmni\_product, bahmni\_sale

**Master Data required :**

**Village:** In localization we can create 1 new object for storing villages, as existing bahmni module has this field as char, we can replace it with many2one field.

- This module will be useful for all activities, which are needed to sync Bahmni data with openerp.
- This module will have dependency of product and sale module.
- This module also consists dhis report, which can be generated from company's form view.

Class Name	Implementation	Class exists In module	Development status
drug.service.create	<b>New class</b> has to be defined for creating products, which <b>will be replacement of 5 classes of existing bahmni_modules i.e. drug.service, lab.test.service, lab.panel.service, radiology.test.service.</b> This class would not create table in database, as storing api call information is not necessary.	New class	Done
product.uom.service	<b>New class</b> for syncing i.e. creating/updating product.uom and product.uom.category records , which we get from bahmni's api call to openerp. This class will not create a table in database.	New class	Done
atom.event.worker	<b>New class</b> to which api call is done from bahmni side. This class won't create a table. Will consist a method to process all the events.	New class	Done
event.records	A <b>new class</b> for publishing events, whenever there is any change done in specified category. e.g. If any change is done to product, a new product is created or product is deleted such events are stored in form of event records.	New class	Done
res.partner	<ul style="list-style-type: none"> <li>• Override name_get method to add ref field in string returned by it.</li> <li>• Add unique constraint on ref field of customer.</li> <li>• Add many2one field for village.</li> </ul>	base	Done
res.partner.address	<ul style="list-style-type: none"> <li>• This class stores the details for the address of partner and a one2many field has added in</li> </ul>	New class	Done

# Final Modules List

	<p>res.partner, which is associated with this class.</p> <ul style="list-style-type: none"> <li>• Since each time partner details are getting updated existing record from the class is getting unlinked, hence removing dependency of this class, as all the fields exists in partner itself.</li> </ul>		
res.partner.attributes	<ul style="list-style-type: none"> <li>• This class is having a single field partner_id, as a field one2many exists in class res_partner.</li> <li>• And this class is nowhere getting used, instead just used while updating partner details, stating no use of this class, hence eliminating dependency of this class</li> </ul>	New Class	Done
res.company	<ul style="list-style-type: none"> <li>• New field need to be added for dhis2_code</li> <li>• New action has to be added to res.company to export generated csv.</li> </ul>	base	Done
res.users	<ul style="list-style-type: none"> <li>• Override password field definition, and make it normal char field, hence it automatically removes effect of methods getting called for computing this field.</li> <li>• New method to generate salt(md5_encrypted_string).</li> <li>• Override check_credentials method to set this md5 encrypted string in password crypt.</li> <li>• Inherit write method, to set password_crypt as md5 encrypted string.</li> </ul>	auth_crypt	Not Done, ===== Didn't get why exactly it is done, and md5_crypt is deprecated, hence not disturbing anything in authorization
order.picking.type.mapping	<p>New Object for mapping OrderType passed through bahmni and Picking Type existing in Odoo. As for creating sale order, we need warehouse_id instead of shop_id in version 10.</p>		Done
syncable.units	<p>New object for storing units which are allowed sync as it is</p>		Done

# Final Modules List

## 2. bahmni\_product

**Dependency :** product, product\_expiry

- This Module is custom addon for bahmni, which consists of all the customization related to product.

product.category	<ul style="list-style-type: none"> <li>• new character field need to added <b>uuid</b>, and set it's value by overriding create method. since this id will act as unique id for that record, no need to update it on write.</li> <li>• this uuid field is synced with bahmni and emr , hence it's value is get from the api call, if no value got from call then it's generated.</li> </ul>	product	Done
product.uom	<ul style="list-style-type: none"> <li>• new character field need to added <b>uuid</b>, and set it's value by overriding create method. since this id will act as unique id for that record, no need to update it on write.</li> <li>• this uuid field is synced with bahmni and emr , hence it's value is get from the api call, if no value got from call then it's generated.</li> </ul>	product	Done
product.uom.category	<ul style="list-style-type: none"> <li>• new character field need to added <b>uuid</b>, and set it's value by overriding create method. since this id will act as unique id for that record, no need to update it on write.</li> <li>• this uuid field is synced with bahmni and emr , hence it's value is get from the api call, if no value got from call then it's generated.</li> </ul>	product	Done
res.partner	<p>New Fields:                      manufacturer: A boolean field to identify whether partner is manufacturer.</p>		Done
product.supplierinfo	<ul style="list-style-type: none"> <li>• New fields:                             <ul style="list-style-type: none"> <li>○ <b>unit_price</b>: functional field for getting price from pricelist.partnerinfo, as whenever value in that one2many field is filled, and that supplier is selected in purchase quotation, price is taken from record which has highest sequence i.e. lower value in sequence field.                                     <ul style="list-style-type: none"> <li>■ This field should be visible only when pricelist.partnerinfo's one2many field is visible. As this field is only computed when there is record in pricelist.partnerinfo.</li> </ul> </li> </ul> </li> </ul>	product	Done

# Final Modules List

	<ul style="list-style-type: none"> <li>○ <b>manufacturer:</b> many2one field with res.partner class, and domain applied for partner which are manufacturers.</li> </ul>		
product.template	<ul style="list-style-type: none"> <li>● Override def create and write to set mrp in related product_product record</li> <li>● New fields : <ul style="list-style-type: none"> <li>○ New float field for mrp.</li> <li>○ manufacturer : many2one field for showing partners in which manufacturer checkbox is checked.</li> <li>○ drug: character field to store generic name of product.</li> </ul> </li> </ul>		Done
product.product	<ul style="list-style-type: none"> <li>● New field for <b>dhis_code2</b> has to be added.</li> <li>● Override get_product_available method to list products which are not expired.</li> <li>● New fields: <ul style="list-style-type: none"> <li>○ uuid: character field to identify product while reverse-sync</li> <li>○ actual_stock : functional field , for computing actual stock of product in warehouse, if location provided then for that specific location.</li> <li>○ mrp : float field for storing mrp of product, when product variants are defined for a product mrp will get maintained at variant level, hence this separate field.</li> </ul> </li> <li>● name_get is overridden to add product category as suffix. - name_get is overridden for showing category as suffix, this can be seen on any many2one field with relation: product.product.</li> </ul>	product	Done

### 3. bahmni\_purchase :

**Dependency :** purchase, bahmni\_product

- This module is for customization in purchase module. hence, in this module there will be dependency of purchase module.

purchase.order.line	<ul style="list-style-type: none"> <li>● <b>New fields:</b> <ul style="list-style-type: none"> <li>○ <b>mrp:</b> Float field, which will get value of mrp from product master, if set else user can set.</li> <li>○ <b>manufacturer:</b> many2one field for setting</li> </ul> </li> </ul>	purchase	Done
---------------------	--	----------	------

# Final Modules List

	<ul style="list-style-type: none"> <li>value of manufacturer. <ul style="list-style-type: none"> <li>○ <b>product_category</b>: many2one field of product.category to store value of category of product set in the product master.</li> </ul> </li> <li>● override product_id onchange method to set all above values.</li> </ul>		
product.product	<ul style="list-style-type: none"> <li>● New methods defined to get mrp from supplierinfo, if available ; else set mrp : def <b>get_mrp</b> and def <b>set_mrp</b> from bahmni_pharmacy_product to be referred.</li> <li>● <b>low_stock</b>: boolean field, but a functional field, and is set when available quantities for a product are below minimum qty level mentioned in reordering level.</li> </ul>	product	Done

#### 4. bahmni\_sale:

**Dependency** : sale, sale\_stock, bahmni\_account

- **Master Data**: Sale price markup table configuration.
- New widget created for link-prescription, for adding a new button to redirect to bahmni app.  
**Try** : instead of widget, if normal button can be used in js functionality, Or is there any other way to redirect to url.
- Sale Order report is overridden to add provider name column in the report.
- New report has to be added for Average Turnaround time and sale turnaround time.

sale.order	<ul style="list-style-type: none"> <li>● <b>New fields</b> <ul style="list-style-type: none"> <li>○ <b>external_id</b>: this field stores the encounter ID got from api call.</li> <li>○ <b>dispensed</b>: A boolean field, to identify whether drug order is dispensed or not.</li> </ul> </li> <li>● Create Invoice button's type is changed from action to workflow.</li> <li>● directly workflow signal action_invoice_create is called on button.</li> </ul> <p><b>New fields:</b></p> <ul style="list-style-type: none"> <li>● <u><i>Datetime_order : Datetime type field.</i></u> <ul style="list-style-type: none"> <li>○ <u><i>Override create and write method to set value of date_order field.</i></u></li> <li>○ <u><i>date_order field to be made invisible.</i></u></li> </ul> </li> </ul> <p><b>Date_order field is already datetime type.</b></p> <ul style="list-style-type: none"> <li>● partner_village: many2one field with village class, which is set as master data in bahmni_atom_feed module.</li> <li>● partner_uuid: unique id required in case of</li> </ul>	sale, sale_stock	Done
------------	---	---------------------	------

# Final Modules List

	<p>reverse sync.</p> <ul style="list-style-type: none"> <li>● Both partner_village and partner_uuid values will be populated on change of partner_id.</li> <li>● care_setting: selection field for giving user option to select between IPD/OPD.</li> <li>● provider_name: many2one field with res.partner.</li> <li>● discount_percentage : float field for entering discount percentage depending on type of discount.</li> <li>● Chargeable_amount : if user is allowed to do so, this field will be editable to user for setting overall chargeable amount. <ul style="list-style-type: none"> <li>○ This field is also visible on setting a sale configuration.</li> </ul> </li> <li>● Default_quantity : field visible to user, who is allowed to set default quantity to -1. <ul style="list-style-type: none"> <li>○ This field is used when sale order is getting created from Bahmni prescription.</li> <li>○ When in sale configuration, configuration for allowing default quantity as -1 is set, in order lines -1 quantity is set.</li> </ul> </li> <li>● discount_amount: if type of discount is fixed, then entering fixed discount amount in this field.</li> <li>● round_off: round_off amount calculated based on round off by value set in the configuration.</li> <li>● discount_acc_id: Discount Account head.</li> <li>● prev_outstanding_balance, total_outstanding : functional fields to get previous balances for partner.</li> <li>● def_prepare_order_line_move has to be overridden to add prodlot_id to move_line values.</li> </ul>		
sale.order.line	<ul style="list-style-type: none"> <li>● <b>external_id</b> : this field stores the encounter ID got from api call.</li> <li>● <b>external_order_id</b>: this field stores the order ID got from api call. This ID will help us to identify, that this line belongs to which sale order.</li> <li>● <b>order_uuid</b>: new char field for just generating a random unique id, using uuid. <ul style="list-style-type: none"> <li>○ <b>Override create and write</b> method to set value of order_uuid, if passed through api, else generate a random unique sequence using uuid package of python.</li> <li>○ Track the dispensed status of order while processing order line too.</li> </ul> </li> </ul>	sale	Done
stock.move.split.line	<ul style="list-style-type: none"> <li>● Default method is set for sale_price field.</li> </ul>		Done

# Final Modules List

s	<ul style="list-style-type: none"> <li>● Onchange of cost_price new method is defined to set sale_price.</li> <li>● <b>For calculating sale_price markup_percentage table is hard coded in this method, which can be made configurable.</b></li> <li>● <b>Since, this method has to be applied only on incoming shipments, moving this code to purchase module only.</b></li> </ul>		
sale.config.settings	<ul style="list-style-type: none"> <li>● New fields has to be added : <ul style="list-style-type: none"> <li>○ convert_dispensed: which allows directly confirm sale order, if drug is dispensed from local shop.</li> <li>○ group_final_so_charge : one new field added to sale order, which is changeable for user, this amount will be considered as final chargeable amount.</li> <li>○ group_default_quantity : added this field to allow setting product qty to -1. Field is defined in this module, but used in bahmni_atom_feed module. And only allowed to add -1 qty when sale order created through medication.</li> <li>○ round_off_by : this is the field used for rounding off amount to specified value. This method has to be generic as, this method is used.</li> </ul> </li> <li>● New method has to be added: def set_convert_dispensed</li> </ul>	sale	Done
sale.order.line.make.invoice	<ul style="list-style-type: none"> <li>● def make_invoices overridden to set the values for newly added fields in invoice</li> <li>● There are 2 methods involved for creating invoice from sale_order, so one is for down_payment and other is for invoiceable amount for delivered product depending on invoice_policy.</li> </ul>	sale	Done

## 5. bahmni\_account:

**Dependency :** account, account\_voucher

- New widget print-bill is introduced in this module, to directly give print command to printer. This widget is used on button which is shown on customer payment form, when user validate payment.
- One controller is written to fetch the bill details to qweb template, called in widget.
- There is method written for converting number to marathi, but not used in report.

account.invoice	override def invoice_pay_customer, to return view_id of	account	Done
-----------------	---	---------	------

# Final Modules List

	<p>view_vendor_receipt_form, instead of view_vendor_receipt_dialog_form.</p> <p>New fields added :</p> <ul style="list-style-type: none"> <li>● discount : float field for fixed percentage.</li> <li>● discount_per : float field to define discount percentage.</li> <li>● discount_method: field to select type of discount, fixed/percentage</li> <li>● def compute_invoice_totals overridden to define logic for invoices of type out_refund.</li> <li>● def action_move_create is overridden to debit invoice discount against respective account head. And for that new method is defined i.e. def _update_discount_head.</li> <li>● def _prepare_invoice method is overridden to add values for new fields added in account.invoice</li> </ul>		
account.move	<p>Override def validate method.</p> <p><b>if abs(amount) &lt; 10 ** -4 is changed to if abs(amount) &lt; 0.</b></p>	account	
<p>Account.voucher Customization from bahmni module is applied to account.payment as new class defined for payment.</p>	<ul style="list-style-type: none"> <li>● New fields added ; <ul style="list-style-type: none"> <li>○ <b>Balance_before_pay</b>: this field is added to account_voucher to know the due amount for invoice or customer/supplier</li> <li>○ <b>Total_balance</b>: amount that would reflect the amount which would be pending after paying mentioned amount.</li> <li>○ <b>Invoice_id</b>: searched invoice from credit lines of voucher.</li> <li>○ <b>Bill_amount</b>: field to get the invoice total amount.</li> <li>○ But in invoice_id, only single invoice is getting considered and bill_amount also reflect the amount of single invoice.</li> </ul> </li> </ul> <p><b>Above fields instead of adding to account_voucher, added to account_payment, as payment is done through that object only.</b></p> <ul style="list-style-type: none"> <li>● <u>get_journal</u> method is overridden to overwrite a condition. <ul style="list-style-type: none"> <li>○ There is a condition for checking ttype value, and if that value matches to payment or receipt then journal type should be bank ; but removed this condition and made ttype to cash.</li> </ul> </li> </ul>	account_voucher	



# Final Modules List

	<p><b>This is achieved in Odoo 10 in account_payment class by applying (bank, cash) domain in field definition itself.</b></p> <p><b>Hence overriding field definition to set domain of cash only.</b></p> <ul style="list-style-type: none"> <li>• Inherit default report of account.voucher to add new fields in that report. &amp; replace this inherited report with original one.</li> <li>• <b>onchange_partner_id</b> method has to be overridden to populate amount to be paid in amount field.</li> <li>• New button to print the receipt, which will give print command to machine, instead of downloading report. <ul style="list-style-type: none"> <li>○ These reports are generated using controller.</li> <li>○ In this controller, error handling has to be done, when account_voucher does not have sale_order id stored in it.</li> </ul> </li> <li>• New button to print summarised bill in English.</li> <li>• New button to print summarised bill in Marathi.</li> </ul>		
res.users	<ul style="list-style-type: none"> <li>• One new field initials of character type added to partner.</li> </ul>	base	
res.company	Override get_header method, to change header for reports.	base	
account.count.report	<ul style="list-style-type: none"> <li>• Count of account heads in sale orders over a period</li> <li>• Sql report for getting number of invoices generated in the system for an account, date-wise.</li> <li>• Filter has to be added required accounts only. Account type has been hard-coded.</li> </ul> <p><b>NEED TO TRY: If this can be configurable. If configurable filter is not possible then, this filter will be treated as part of search_setup_data.</b></p>	New class	Done
account.report	Sql report for getting account wise expense and collection.	New class	Done

## 6. bahmni\_web\_extension :

**Dependency :** base, web

- accesskeyHighlight.js : Defining Accesskey in buttons. These Access keys are underlined and capitalized for buttons in which accesskey is defined.

# Final Modules List

- addContextOnExport.js : On click of export data, added domain, context and ids in returned action.
- addingAccessKeyToOne2ManyList.js: replaced string 'add an item' to Add an Item.
- addSerialNumberToListView.js: Added Sr.No. to each item in tree view. for each page, this sr.no. is generated.
- fixingErrorInCancelEdition.js: Error handling done in method cancel\_edition. If attrs is not null, then only attrs.id condition has been checked.
- init.js : initialisation of all the above files..
- overriding index method has solved the issue of exporting csv, which was getting faced in default web module.

## 7. bahmni\_stock

**Dependency** : stock, bahmni\_product, bahmni\_account

product.template	<b>procure_method</b> : set default value to make_to_stock	procurement	Done
product.product	<ul style="list-style-type: none"> <li>● new method added to class, def get_stock_for_location to get the available quantities of product at selected location. <ul style="list-style-type: none"> <li>○ This method is used when available quantities for specific location is to be computed.</li> <li>○ This method is dependant on batch.stock.future.forecast</li> </ul> </li> </ul>		
stock.picking	<ul style="list-style-type: none"> <li>● New boolean field has to added: warned.</li> <li>● New onchange method has been added on location fields. So that the locations which are selected in the parent form, will get reflected in the child lines. - <b>In child lines values for location_id and location_destination_id are passed through context</b></li> </ul>	stock	
stock.partial.picking	<ul style="list-style-type: none"> <li>● Def do_partial is inherited to add validation when picking_type is not internal on these two conditions :</li> <li>● When duplicate serial number is entered or no serial number is entered - <b>In the window where serial numbers has to be added, there is validation for duplicate serial number by default. In bahmni new field is added for serial number in stock_move, which is not required.</b></li> </ul>	stock	

# Final Modules List

	<ul style="list-style-type: none"> <li>• Override def do_partial. To raise warning when warned flag is set in picking.</li> </ul>		
stock.move	<ul style="list-style-type: none"> <li>• Method onchange_lot_id is overridden to add warning message when selected serial number/lot is already expired. <b>If lot is expired, and it cannot be processed, instead of raising warning of expired lot, will filter out records in lot_id field to show only those records which are not expired.</b></li> <li>• New method defined _get_stock_for_location, for getting available product quantities at selection location. <ul style="list-style-type: none"> <li>○ For this new field for available qty is added to stock_pack_operation, which will reflect actual qty in stock for product.</li> <li>○ New field in stock_pack_operation_lot to reflect available qty for specific lot.</li> </ul> </li> <li>• New field added : stock_available <ul style="list-style-type: none"> <li>○ Onchange_product_id method is overridden to set qty , which is obtained from _get_stock_for_location method call. <b>Detailed description is in previous pt.</b></li> </ul> </li> <li>• New field added : stock_picking_time <ul style="list-style-type: none"> <li>○ This is a functional field, whose value is set from picking's date value.</li> </ul> </li> </ul>	stock	
stock.move.split	<p>New field added : stock_move Whose value is set as active_id, through overriding default_get method. Rest other values were set in default stock module only.</p>	stock	
account.invoice.line	<ul style="list-style-type: none"> <li>• New fields <b>batch_name</b> and <b>expiry_date</b> to be added.</li> </ul>	account	
batch.stock.future.forecast	<ul style="list-style-type: none"> <li>• This a new object introduced in this module, but this a sql record, and hence records of this object are auto-created through sql view query.</li> <li>• Introduced to get list of serial number according to their future availability.</li> <li>• This class is used to get available qty for a</li> </ul>	New class	Done

# Final Modules List

	batch at specific location.		
prod_last_moved.report	This is a new sql report developed to get the view for getting list of products with their respective last moved date. Last moved date is date of movement of product from one location to another.	New class	Done
prodlots.report	New sql report of products to track the serial number by location. This report shows the serial number and its current location and qty at that location. <b>This can be seen from stock quanrs view, hence eliminated new report</b>	New class	

# Final Modules List

## Setup data modules for already implemented projects.

### 1. **jss\_setup\_data :**

- Refer data files from **lab\_seed\_setup** module.
- these files contains data for creating Accounts, Product Category, Suppliers, Unit of measure, stock location.
- Change of date format
- deletion of service product, which by default created on openerp installation.

### 2. **jss\_logger:**

- Refer **bahmni\_logger** module.
- SMTP logger specifically designed for jss.
- As on bahmni wiki, it's been instructed to uninstall this module; since this module is set to install automatically.

### 3. **search\_setup\_data :**

- Refer to data files in **search\_seed\_setup** module.
- These data files consists of data for creating Accounts, Product Categories, Products, Unit Of measures, Stock Location, Warehouse and shop.
- Service product which gets created on default openerp installation , has to be deleted.
- In settings.xml, main company data is changed, and it's partner data is also changed, partner's name, city, image, phone, email is changed.
- Hard-coded filter added to account.count.report is also part of search\_setup\_data, if configurable filter is not possible.

## Migration of Existing Data

- For migrating existing data of bahmni getting used, we will need to write a separate migration script for each project.
- This script will depend on modules getting used; if any additional custom module is used, then will need to do some changes in migration script.
- It can't be generic for all the bahmni projects which are already implemented.