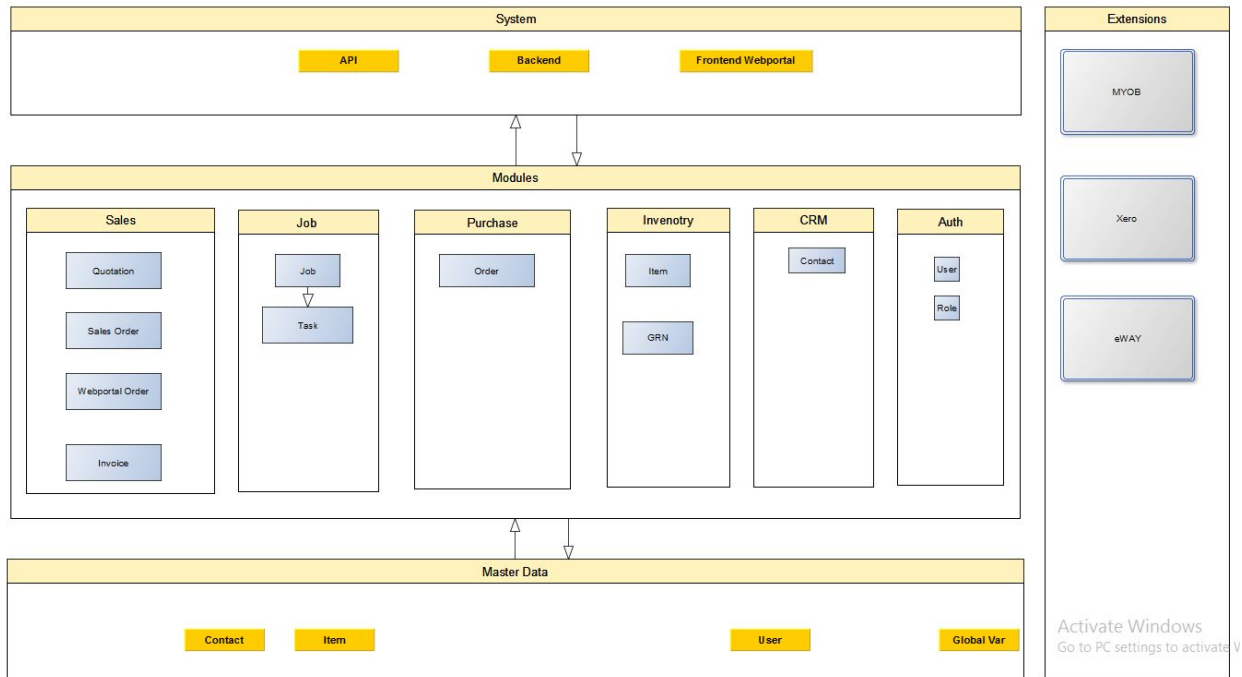


Overview



The system will be consisted by an API, backend which has full system access and frontend web portal which is much like a web store (this can be one or multiple)

Sections have been categorized into independent modules as sales, job, purchase inventory, CRM, Auth and Log (for admin and developer purpose).

By reason of quotation, sales order, web order, invoice, purchase order, good received note, job are following same format in term of costing they will be maintained as individual objects (loosely coupled) to avoid complexity and increase flexibility and reduce bugs. That means none of the aforementioned objects will hold any foreign keys of other objects. As an example a job will not keep a quotation ID in the job table as a foreign key when it is created from a quote or sales order.

To reference aforementioned objects each other, system will use a mechanism called “order reference”.

So all of the above objects (Orders) can be created and update separately by json string (json file) or using a form in the backend system.

Modules

Sales

- Quotation
- Sales Order
- Webportal Order
- Invoice

Purchase

- Order

Job

- Job
- Task

Inventory

- Item
- GRN

Contact

- Customer
- Supplier
- Employee

Auth

- User
- Role & Permissions
- Rules

Log

Components

Kanban Board -

The simplest kanban board consists of three columns: "to-do", "doing" and "done"

This component can be used to manage job or to manage customer relations etc.

PDF template plugin

User can upload their own templates based on given data object details

As an example; if user need a customizable quote template rather than built in template, user can see what data available from quotation object and make his/her template file according to that and upload the file. Template file can be choosed before printing, downloading or sending it, also user can override template setting variable with new template name

Extension

- MYOB API
- XERO API
- eWAY payment gateway

Development Instructions

Hacks

- Always test if jquery been registered more than one time. Otherwise pjax and grid filtering may not work

Git repository

Coding standard

follow the [standard PHP style coding guideline](#)

For database

Enable pretty URL

Enable pretty URLs and take advantage of HTTP verbs.

Caching

to improve the performance of the Web application

Database migrations

For every **database** schemas, migrations should be generated.

Unit Testing

Unit testing will be done with Codeception Unit Tests

Authentication

Logging with different devices

Particular user may be logged to the system by different devices at a given time, or by different browsers in same device. So system will recognize this behavior and user might have to pass an authentication (like two-factor authentication layer) layer to be logged in.

Use user recognize mechanism object with user ID

Authorization

- All permissions should be created in developing mode.
- To create permissions, all permissions should be declared in a migration class.
- User with ID 1 will be treated as “Admin” user and allow to access whole system
- By default visitor “role” should be available which has access to just for backend index (dashboard)
- Don’t assign default role to users when created, because of some user might have access to only for some modules by passing backend - home page. As an example user whose role has been assigned as supplier, can only see supplier related data under “Supplier” module

Permission should be created in the following format and should be match to module id, controller id and action id

module.controller.action

Example ;

app-backend.site.index
auth.rbac.view-permission
auth.rbac.set-permission
sales.quotation.create
sales.quotation.print-pdf

- Add description when create permission to give more user friendly instruction about the permission
- Define user ids in config as “**noPermissionUserIds**” that need to bypass the AuthAccessControl filter
- AuthRule should be define in app/backend/config component as authRule and “**allow**” property should be true to work the AuthAccessControl filter properly
- All controller actions which need to be invoked without permissions (like ajax loadings) should be declared in controller under “rules” as “**silentAjaxActions**”

Components

MarginMarkupLookup

Function margin(rate)

return integer margin;

Function markup(margin,percentage = false)

if percentage == true return markup as percentage

else return decimal rate;

The purpose of this class is to find the appropriate margin out of the markup multiplier (or rate) and vise versa

For example the output of “36” margin percentage would be “1.56”

Calss DecimalRounding (abstract or interface)

Function roundValue(val) return decimal

Classes need to be derived from the superclass for each rounding methods

Rounding methods

Follow the link

<https://www.syspro.com/blog/applying-and-operating-erp/rounding-for-accounting-accuracy/>

1. Conventional Rounding
2. Argentina Rounding
3. Swiss Rounding

There might have more rounding methods as client needed

Detail Grouper

If there are multiple item matching for given details like quote details, sales order details ect. Then group them and merge them into single line item

Take arguments to group them like item code, price ect. If provided arguments match then do the grouping

This component can be used in several places including

1. On adding items to detail section
2. Export order
3. Print order
4. Send order

System settings and global vars

Following system settings will be available and can be set globally. Some global settings can be overridden or overwritten in module sections.

Layout

Colour theme

Font type

Operation menu position - right verticle/ right horizontal

Operation menu buttons - icons only/ icon with text/ text only

Menu items order - most visited first

Dashboard layout - select predefined layout

Security

1. **Enable Two-Factor Authentication** - enable or disable two factor authentication layer
2. **Two-Factor Authentication Method** - choose two factor authentication method. Default to QRCODE scan. Other available one is email auth code
3. **Allowed IPs**. if there are any IP address then system will only allow to access the system from that IP, multiple IP address can be added separated by comma. An IP address can contain the wildcard * at the end, so that it matches IP addresses with the same prefix. For example, '192.168.*' matches all IP addresses in the segment '192.168.'
4. **Enable Lock screen** - if user idle for certain time, then user will be redirect to lock screen. User need to provide the password to unlock the screen.
5. **Enable password reset** by users themself otherwise admin can reset

Format

1. Date
2. Rounding precision - number of precision
3. Currency name - AUD/USD
4. Currency symbol - show or not
5. Number format - 123,123.00 / 123.123,00

Accounting

1. Decimal Rounding method : Three rounding methods will be available.
2. Tax type - GST/VAT
3. Tax Rate - 1.1 -> 10%
4. Tax account - account name
5. Sales account - default account name
6. Buying account - default account name
7. Show cost - for these role names

Print

Header - on/off

Page number - on/off

Footer - on/off

PDF layout - import as a php file or select an available layout based on provided parameters

Export

Bulk export - enable/disable

File types - excel/ PDF/ csv / XML

Format - import format as PHP file based on provided parameters

Gridview

Advance filter - on/off

Advance filter roles - role names

Number of rows - 20

Default sort order - created date/ updated date

Row options label - icons/ text

Show delete option each row - on/off

Show update option each row - on/off

Email

1. Send order email with order confirmation link
2. Email layout : use default or upload custom layout as PHP file based on parameters
3. Enable multiple attachments
4. Enable cc and bcc
5. From email : admin email as default
6. Default cc, bcc - email address
7. Signature : image/ text / or template file

Order

1. Enable Detail Grouper which can group same kind of items into one line

Security

Following security principles will be applied to the system as needed.

Allowed IPs

Because of using this application is based on a particular organisation, user IPs can be recognized and filtered. In this way system security can be improved in a remarkable way. Admin will be able to manage IP addresses.

All inputs should be filtered and all outputs should be escaped including data which is sent and received via api extensions like MYOB, Xero, eWAY.

Encryption and Decryption

For example, we need to store some information in our database but we need to make sure only the user who has the secret key can view it (even if the application database is compromised):

Confirming Data Integrity

There are situations in which need to verify that the data hasn't been tampered with by a third party or even corrupted in some way. Use Yii provided data integrity mechanism to hash data with a secret key and validate it.

Avoid SQL injections

To avoid SQL injections always use bind parameters or PDO prepared statements.
In case it's not possible, use raw queries with special syntax that Yii2 provides to escape sql injections like in following query example **`$sql = "SELECT COUNT([[$column]]) FROM {{table}}";`**

Avoiding XSS

To avoid XSS or cross-site scripting two methods can be used

Escape plain text with `Html::encode`

Output HTML with `HtmlPurifier::process`

Avoiding CSRF

In order to avoid CSRF

Follow HTTP specification. i.e. GET should not change application state.

Keep Yii CSRF protection enabled.

In a case of disabling Yii CSRF protection, extra validation such as checking IP address or a secret token should be implemented.

Avoiding debug info and tools in production

Remove `php_info()` in production mode if possible

Disable `gii` module and debug tool in production mode.

If developer need to enable them in production mode restrict it to developer IP only

Avoiding Host-header attacks

fix your web server configuration to serve the site only for specified host names or explicitly set or filter the value by setting the `hostInfo` property of the request application component.

For more information refer to the documentation of the server

In a case that don't have a access to the server configuration, setup

`yii\filters\HostControl` filter at application level in order to protect against such kind of attack:

Authentication

The system will use currently available most secure password encryption method **bcrypt** to store the user password in database.

So users should have to remember their passwords otherwise they might have to reset them by themselves or by admin.

Two-step authentication

To make logging in to the system more secure and is now required by the most government authorizations including Australian tax office, user should go through two authentication steps to be logged in.

First step is normal username and password entering step and then system will send a randomly generated 6-digit authentication code to users' email which is required to pass the second authentication step.

This auth code will expire in (n) seconds (this can be set by the admin) and remember the device upto (n) days (this also can be set by the admin). That means during that period second auth step may not be needed.

Logging with different devices

Particular user may be logged to the system by different devices at a given time, or by different browsers in same device. So system will recognize this behavior and user might have to pass an authentication (like two-factor authentication layer) layer to be logged in.

Logging attempts

If user fails to enter their logins correctly in (n) times (admin can set this values) next time login screen appears with CAPTCHA. Still user fails to enter the correct logins with the CAPTCHA verify code in (n) times (admin can set this value) user will be blocked (admin to set the option) based on the user recognizing mechanism.

User recognizing mechanism

System will use following methods to identify and determine the probability of being a same user.

IP Address

- Real IP Address
- Proxy IP Address (users often use the same proxy repeatedly)

Cookies

- HTTP Cookies
- Session Cookies

- 3rd Party Cookies
- Flash Cookies

Browsers

- Click Tracking (many users visit the same series of pages on each visit)
- Browsers Finger Print - Installed Plugins (people often have varied, somewhat unique sets of plugins)
- Cached Images (people sometimes delete their cookies but leave cached images)
- Using Blobs
- URL(s) (browser history or cookies may contain unique user id's in URLs,
- System Fonts Detection (this is a little-known but often unique key signature)

HTML5 & Javascript

- [HTML5 LocalStorage](#)
- HTML5 Geolocation API and Reverse Geocoding
- Architecture, OS Language, System Time, Screen Resolution, etc.
- Network Information API
- Battery Status API

For each piece of information which system can gather on a given request will be stored with user profile and calculate the probability of being same user . if user probability is not in a satisfied level then the user will be logged out by the system.

User Activity Tracking

Each and every user activity will be tracked

System will track user data coming with request on before every action.

Code for the tracking user data will be placed in config file under "on beforeAction"

Reset password

Users can reset their passwords by themselves, an email verification link will be sent to a given user email (only if particular user email has already been registered in the system, otherwise silently send email to the given email address with warning). When a user click on verification link in the email he will be redirected to the password resetting page.

User creation / updation

Admin will be allowed to create logins for users with access roles by providing an email address. When create a user by the admin, a user account activation link will be sent to the given email address and user should activate their accounts by clicking on that activation link or copy and paste it in a web browser address bar. However the password will not be sent through the email. So the admin might be needed to provide the password to the user using another method.

Also admin can update the active state of the system users (so both activations by user and admin are needed users to be logged in)

Note : if some user lost both his or her password and email logins, the admin can update the email with password and sent an activation link to the new email.

Email addresses must be unique.

Authorization

System will use Yii2 Role-Based Access Control (RBAC) to verify that a user has enough permission to do something.

Roles and permissions

A role represents a collection of permissions (e.g. creating quote, updating quote) and can be assigned to one or multiple users.

By default index (dashboard or home) and logout permissions will be assigned to each user

Rule

Associated with each role or permission, there may be a rule. A rule represents a piece of code that will be executed during access check to determine if the corresponding role or permission applies to the current user.

For example, the “delete quote” permission may have a rule that checks if the current user is the quote creator. During access checking, if the user is NOT the quote creator, he/she will be considered not having the “delete quote” permission. With this “rule” system can handle more complex verifying processes.

System will provide an interface to manage roles and permissions and to assign them to user or revoke them from user.

Rules will be managed by the developers only..

Error Handling

Following errors can be occurred when run the application due to code bugs and other technical related issues like server settings or user requests issues

Fatal error - causes script termination

Fatal errors - that occur during PHP's initial startup

Run-time warning - that does not cause script termination

Compile time parse error

Run time notice caused - due to error in code

Warnings that occur during PHP's initial startup

Fatal compile-time errors indication problem with script.

User-generated error message.

Most possible cases of E_USER_ERROR is

- Call to Undefined Function
- Endless Loop
- Endless Dynamic Concatenation of String (or) Endless Dynamic Manipulation
- Creation of Undefined Class Object
- Redclaration of an existing Class
- Redclaration of an existing Function

User-generated warning message

User-generated notice message

Run-time notices

Catchable fatal error indicating a dangerous error

All error and warnings will be traced and sent to the admin or developer email immediately by the system so they can get an action to fix the issue as soon as they can.

No errors will be seen by end user instead system shows user friendly error pages.

HTTP errors

- 400 Bad Request. ...
- 401 Unauthorized. ...
- 403 Forbidden. ...
- 404 Not Found. ...
- 500 Internal Server Error. ...
- 502 Bad Gateway. ...
- 503 Service Unavailable. ...
- 504 Gateway Timeout.

These errors will be redirected to a specific template pages, and also other errors according to the client request as well.

Beside these errors there will be exceptions which are thrown by the developers.

Debug mode should be turned off when the application go live. Otherwise fatal errors are not redirected to the error pages instead system shows errors to user which is considered as a bad user experience and may lead to security issues as well.

All errors will be subjected to monitor via log module which has ability to fully manage the log messages. For more details see the log module section.

Logging

Following logging methods will be used to record various types of messages

- [`Yii::debug\(\)`](#): record a message to trace how a piece of code runs. This is mainly for development use.
- [`Yii::info\(\)`](#): record a message that conveys some useful information.
- [`Yii::warning\(\)`](#): record a warning message that indicates something unexpected has happened.
- [`Yii::error\(\)`](#): record a fatal error that should be investigated as soon as possible.

Master data

The data tables and data which should be there to start the application and work auth process and other modules are considered as master data.

For example there should be a user table and other auth related data tables to work auth process so also customer table should be there to work sales module and purchase modules etc.

Contacts

System will keep customer, supplier and sales persons data in a table called contact and can be categorized into the groups like customer, supplier and sales persons.

In this way system may have another group of contact like employee for payroll module , so the system will be able to handle those kind of requirements with minimum changes.

Global vars (settings)

System will maintain variables that can be used in global level by the application components. As example date format, currency symbol, 1000 separator will be kept in global vars. System admin will be allowed to edit those variables and more variables could be used in the system as needed.

Overwrite global vars

Global variables can be overwritten in local components such as quote, invoice etc. as an example globally defined date format “m/d/Y” can be overwritten as “Y-m-d” in quote, so also locally overwritten global vars can be overwritten again when printing, sending email etc. on the fly.

Items

System will maintain all kind of items in one table. Instead using separate tables for countable items and another tables for non countable items like decorations and other costs etc. however grouping option is available to group items into various categories like products, decorations or other costs. Category levels will also be available to give more advanced filtering facility to items.

Typical item properties and usage

| Item |
|----------------|
| UUID |
| code |
| name |
| description |
| category_level |
| account_id |

```
final class Category{

    const CATEGORY_LEVEL_ONE = 1;
    const CATEGORY_LEVEL_TWO = 2;
    const CATEGORY_LEVEL_THREE = 3;

}
```

```
class Item {
    private category_level;
    private category_list = array(1,2,3);//white list

    __construct(int category_level){
        //category level must be defined in object creation
        if(in_array(category_level, category_list)){
            throw new CategoryLevelNotFoundException();
        }

        $this->category_level = category_level;
    }

    public function getCategoryLevel(){
        return category_level;
    }
}
```

Only items which has CATEGORY_LEVEL_ONE will be allowed to be inventoried that means countable products. All other level items will be considered as wrapping items.

Orders Objects

quotes , sales orders, web orders, invoices, purchase orders, good received note and job can be considered as orders in term of costing format. As an example all those components have a header and a detail section which are common among all orders. Header will be consisted of number, name, due date etc. while detail section will be consisted of products (items) and each product may have decorations and each decoration may have some kind of costs.

The following format will be used as the order costing format, which helps to covers most of the costing and customising needs

Order header basic

| | | | | |
|--------------------|------------|----------------------------|--------------|-----------------|
| To _____ | Date _____ | Due Date _____ | Number _____ | Reference _____ |
| Sales Person _____ | | Comment _____ | | |
| Currency: AUD | | Amounts Are: Tax Exlucisve | | |

Order detail basic format

| | | |
|-----------------|--|--|
| Category 1 Item | | Item-----Description-----Account-----Qty-----Cost-----MarkUp(%)-----Price-----CostAmount-----PriceAmount |
| | | IT01 test desc 200-1 100.00 12.50 1.05 17.25 1250.00 1725.00 |
| | | Tab-----Decoratoin-----Size/Qty/Colour-----WYSWING----- |
| Category 2 Item | | Item-----Description-----Cost-----MarkUp(%)-----Price-----CostAmount-----PriceAmount |
| | | Deco001 deco test 1 25.50 1.5 32.52 2550.00 4878.00 |
| Category 3 Item | | Item-----Description-----Cost-----MarkUp(%)-----Price-----CostAmount-----PriceAmount |
| | | DC1 test deco cost 1.25 1.05 3.2 1250.00 3200.00 |
| | | DC2 test deco cost 1.25 1.05 3.2 1250.00 3200.00 |
| | | Deco001 deco test 1 25.50 32.52 1.5 2550.00 4878.00 |
| | | Item-----Description-----Cost-----MarkUp(%)-----Price-----CostAmount-----PriceAmount |
| | | DC1 test deco cost 1.25 1.05 3.2 1250.00 3200.00 |
| | | DC2 test deco cost 1.25 1.05 3.2 1250.00 3200.00 |
| | | OtherCost deco test 1 25.50 32.52 1.5 2550.00 4878.00 |
| | | Item-----Description-----Cost-----MarkUp(%)-----Price-----CostAmount-----PriceAmount |
| | | DC1 Freight 1.25 1.05 3.2 1250.00 3200.00 |
| | | DC2 Service Charge 1.25 1.05 3.2 1250.00 3200.00 |

As shown in above draft format, an item has been categorized up to THREE categories. Items in category one can be wrapped by items which are in category two and items in category two can be wrapped by items which are in category three respectively. For now system allow only upto three category levels to be wrapped in.

Beside **wrapping** objects, an item has two other objects called **SizeQtyColourMetric** and **WYSIWYG**.

To collect data into the items by categories for order creation and updating process, class implementation might be as follows;

```

ItemCategoryOne{

    private ItemCategoryTwo[] category_two_item_list;

    public function cost(){
        double cost = 0;
        foreach($this->category_two_item_list as $item){
            cost += $item->cost();
        }

        return cost;
    }

}

```

```

ItemCategoryTwo extends ItemCategoryOne {

    ItemCategoryOne item_category_one;
    private ItemCategoryThree[] category_three_item_list;

    public function cost(){
        double cost = 0;
        foreach($this->category_three_item_list as $item){
            cost += $item->cost();
        }

        return cost + item_category_one->cost();
    }

}

```

```

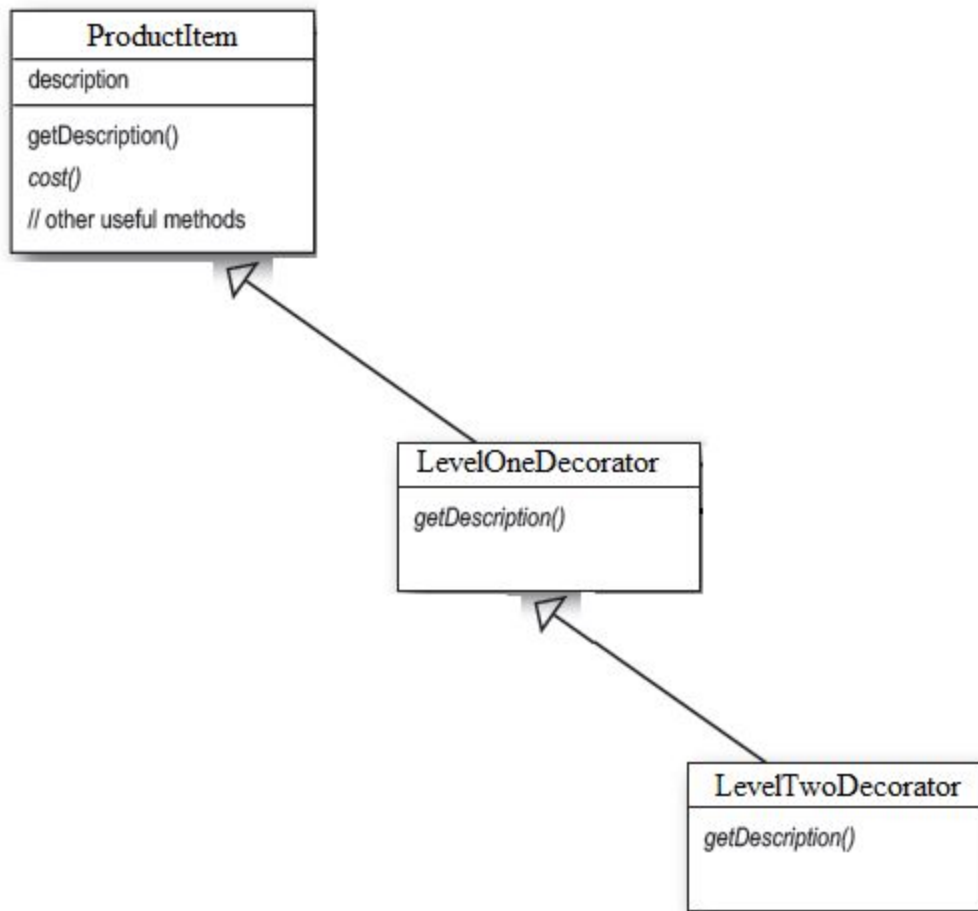
ItemCategoryThree extends ItemCategoryTwo {

    public function cost(){
        return 100.00;
    }

}

```

This class implementation uses decoration pattern.



Every order objects should maintain separate instances for their properties. As example after assign a customer to a quote, beside customer original Id, details of that assigned customer, like address, telephone can be changed without affecting to original customer details, and when create another order object wrapping this quote order, it should have post modified customer data and it should flows to next wrapping object if done.

Template and Theme

Feeling about the system layout and theme is the most important part in the client point of view. Before dig into the features and functions available in the system client catches sight of the layout and theme. So a bad layout keep them away from a system, though the client doesn't have an ability to tell what is the wrong with our system.

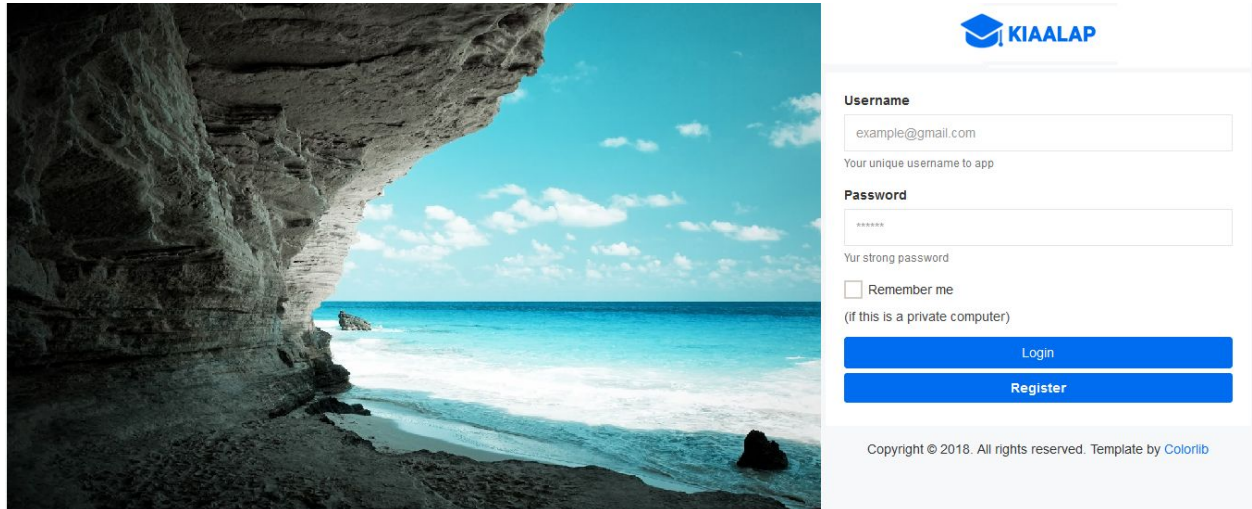
A bootstrap framework based template called "kiaalap" will be used as the backend main template. It has lot of attractive built in widgets and supports for responsive layouting.


So it will give a best user friendly environment for users who will be logged by other devices such as smartphones and tabs

Note : please zoom in if you cannot see the screen shot clearly

Note: for wallpapers use non-white background images(bottom part), otherwise "copyright" text can be disappeared as it's coloured in white.

Backend default login screen





Username

Your unique username to app

Password

Your strong password

☐ Remember me
(if this is a private computer)

[Login](#)

[Register](#)

Copyright © 2018. All rights reserved. Template by Colorlib

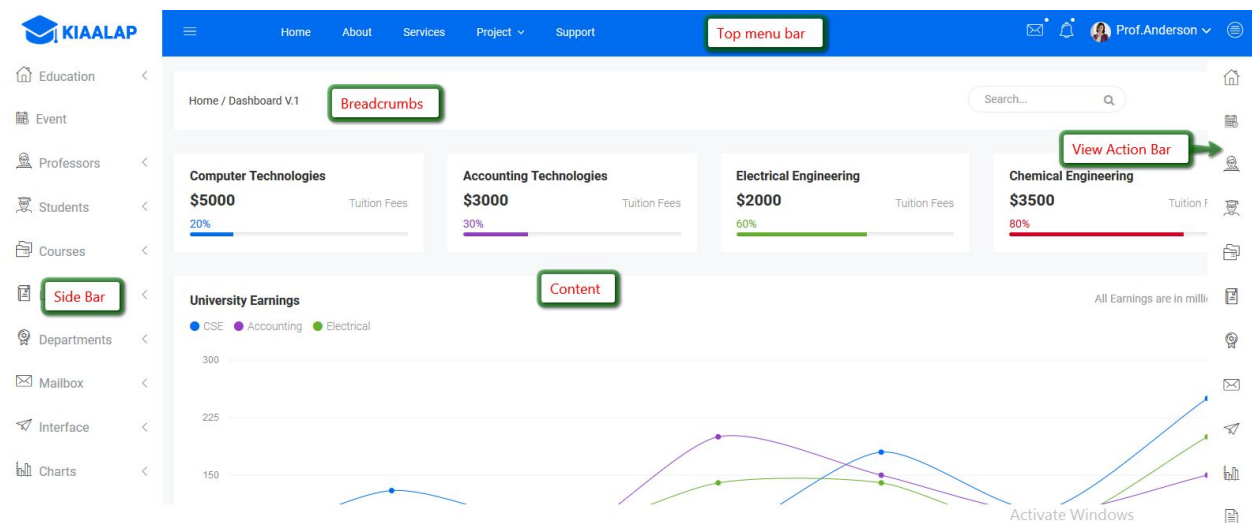
Login form shown in right side with company logo.

Company logo image should be put inside backend/images and should be named as logo.png. The image should be created with transparent background and 200 x 60 in pixels.

Left side wallpaper will be randomly changed.

Admin can add wallpapers.

Wallpapers should be in JPG format and minimum 1200 × 630 in pixels



Top menu bar

Company Logo

Logo will be placed on the top left side corner. Same logo image used here as login form logo.

Logo image should be put inside backend/images and should be named as logo.png. Image should be created with transparent background and 200 x 60 in pixels
Admin can change the logo image.

Main navigations

Links to access home page , quote, job, invoice and global setting pages will be placed in main navigation bar

Left sidebar

Module menus and sub menus will be placed on left side bar. Also other necessary links by categorising into a menu like Reports, Mails, extensions etc.

Content Area

Main contents will be rendered here

View action bar

This bar consists with icons which are related to currently loaded page. As example if currently loaded page is invoice the action bar would have update, delete, print buttons etc.

Global Search

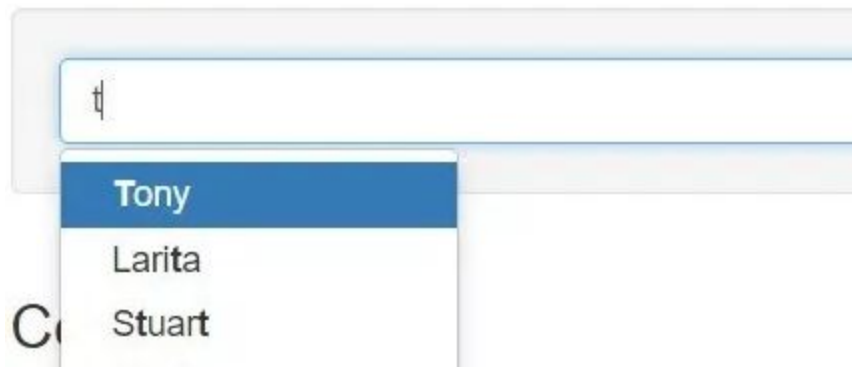
Search box will be placed In top left corner of the content area.

Search box will be support for typehead option.

Search result will redirect user to another page and give categorized search results under following categories

Modules, reports, quotes, sales orders, jobs, invoices, purchase orders etc.

A Typehead example



Breadcrumbs

Breadcrumbs will be placed on the top right corner of the content section, . It indicates current page's location to the user and it will also be used as navigation scheme.

PJAX extension