

# Bayanno Hospital Management System

Advanced management software for hospital, clinic

## Developer Manual

copyright 2015 Creativeitem. All rights reserved.

<http://www.creativeitem.com>

# Index of Contents

- Installation
- Source code structure
- Source code description

# Installation Manual

- Unzip the downloaded folder
- Copy the files inside folder : "Bayanno-Hospital-Management-System" into your server
- The auto web installer will start
- Make sure ***application/config/database.php*** to be writable
- Make sure ***application/config/routes.php*** to be writable
- Make sure CURL is enabled in your server
- Enter codecanyon username and purchase code.
- Here is how to find your purchase code :  
<https://help.market.envato.com/hc/en-us/articles/202822600-Where-can-I-find-my-Purchase-Code->
- Enter your database settings
- Enter your admin settings
- Press the "Run installer" button.
- That's it!
- For installation support, contact to our support center.

# Source Code Structure

The application is developed on Codeigniter framework and completely follows MVC. The details of this framework can be found in <http://www.codeigniter.com/>

The root directory contains

- Application  
contains the core files of the application
- Assets  
contains all css styles and javascript files
- system  
contains all the configurations and library files of the framework
- uploads
  - admin\_image  
contains the profile image uploaded by admin himself
  - doctor\_image  
contains the uploaded images of doctors
  - patient\_image  
contains the uploaded images of patients
  - nurse\_image  
contains the uploaded images of nurses
  - pharmacist\_image  
contains the uploaded images of pharmacists
  - laboratorist\_image  
contains the uploaded images of laboratorists
  - accountant\_image  
contains the uploaded images of accountants
  - receptionist\_image  
contains the uploaded images of receptionists
  - diagnosis\_report  
contains the uploaded documents of diagnosis reports
  - index.php  
this file on loading calls all the codeigniter functions

## application

- config
  - contains the configuration files of the application
  
- controllers
  - admin.php
  - doctor.php
  - patient.php
  - nurse.php
  - pharmacist.php
  - laboratorist.php
  - accountant.php
  - receptionist.php
  - error.php
  - login.php
  - modal.php
  - payment.php
  
- helpers
  
- libraries
  
- models
  - crud\_model.php
  - email\_model.php
  - sms\_model.php
  
- views
  - backend
    - admin
      - contains all the view files of admin panel
    - doctor
      - contains all the view files of doctor panel
    - patient
      - contains all the view files of patient panel
    - nurse
      - contains all the view files of nurse panel
    - pharmacist
      - contains all the view files of pharmacist panel
    - laboratorist

- contains all the view files of laboratorist panel
- accountant
  - contains all the view files of accountant panel
- receptionist
  - contains all the view files of receptionist panel
- footer.php
- header.php
- includes\_bottom.php
  - contains links to javascript files
- includes\_top.php
  - contains links to css files
- index.html
- index.php
- login.php
  - the login page
- modal.php

# Source code description

## Doctor Controller ( /application/controllers/doctor.php )

### Manage Appointments

appointment function contains the logical expressions of showing all appointments created by logged in doctor, creating a new appointment, editing the informations of an existing appointment and deleting appointment informations from the database. The parameters define the form action in which the data is to be created, edited or deleted. An sms is sent to the patient when a new appointment is created or edited by calling send\_sms function with message and receiver phone number as parameters which is defined in /application/models/sms\_model.php

```
function save_appointment_info()
{
    $data['timestamp'] = strtotime($this->input->post('date_timestamp').' '.$this->input->post('time_timestamp') );
    $data['status'] = 'approved';
    $data['patient_id'] = $this->input->post('patient_id');

    if($this->session->userdata('login_type') == 'doctor')
        $data['doctor_id'] = $this->session->userdata('login_user_id');
    else
        $data['doctor_id'] = $this->input->post('doctor_id');

    $this->db->insert('appointment',$data);

    // Notify patient with sms.
    $notify = $this->input->post('notify');
    if($notify != '') {
        $patient_name = $this->db->get_where('patient',
            array('patient_id' => $data['patient_id']))->row()->name;
        $doctor_name = $this->db->get_where('doctor',
            array('doctor_id' => $data['doctor_id']))->row()->name;
        $date = date('l, d F Y', $data['timestamp']);
        $time = date('g:i a', $data['timestamp']);
        $message = $patient_name . ', you have an appointment with doctor ' . $doctor_name . ' on ' . $date . ' at ' . $time . '.';
        $receiver_phone = $this->db->get_where('patient',
            array('patient_id' => $data['patient_id']))->row()->phone;

        $this->sms_model->send_sms($message, $receiver_phone);
    }
}
```

### Send SMS

Clickatell user, password and api\_id is fetched from the database and an authentication call is made to check if the user is valid. If valid then an sms is sent to the receiver phone

number with the message that was passed on to the function as parameters.

```
function send_sms($message = '' , $reciever_phone = '') {

    $clickatell_user      = $this->db->get_where('settings', array('type' => 'clickatell_user'))->row()->description;
    $clickatell_password  = $this->db->get_where('settings', array('type' => 'clickatell_password'))->row()->description;
    $clickatell_api_id    = $this->db->get_where('settings', array('type' => 'clickatell_api_id'))->row()->description;
    $clickatell_baseurl   = "http://api.clickatell.com";

    $text = urlencode($message);
    $to   = $reciever_phone;

    // auth call
    $url = "$clickatell_baseurl/http/auth?user=$clickatell_user&password=$clickatell_password&api_id=$clickatell_api_id";

    // do auth call
    $ret = file($url);

    // explode our response. return string is on first line of the data returned
    $sess = explode(":", $ret[0]);
    print_r($sess);echo '<br>';
    if ($sess[0] == "OK") {

        $sess_id = trim($sess[1]); // remove any whitespace
        $url = "$clickatell_baseurl/http/sendmsg?session_id=$sess_id&to=$to&text=$text";

        // do sendmsg call
        $ret = file($url);
        $send = explode(":", $ret[0]);
        print_r($send);echo '<br>';
        if ($send[0] == "ID") {
            echo "successmessage ID: ". $send[1];
        } else {
            echo "send message failed";
        }
    } else {
        echo "Authentication failure: ". $ret[0];
    }
}
```

## Private Messaging



```

/* private messaging */

function message($param1 = 'message_home', $param2 = '', $param3 = '')
{
    if ($this->session->userdata('doctor_login') != 1)
        redirect(base_url(), 'refresh');

    if ($param1 == 'send_new') {
        $message_thread_code = $this->crud_model->send_new_private_message();
        $this->session->set_flashdata('message', get_phrase('message_sent!'));
        redirect(base_url() . 'index.php?doctor/message/message_read/' . $message_thread_code, 'refresh');
    }

    if ($param1 == 'send_reply') {
        $this->crud_model->send_reply_message($param2); // $param2 = message_thread_code
        $this->session->set_flashdata('message', get_phrase('message_sent!'));
        redirect(base_url() . 'index.php?doctor/message/message_read/' . $param2, 'refresh');
    }

    if ($param1 == 'message_read') {
        $page_data['current_message_thread_code'] = $param2; // $param2 = message_thread_code
        $this->crud_model->mark_thread_messages_read($param2);
    }

    $page_data['message_inner_page_name'] = $param1;
    $page_data['page_name'] = 'message';
    $page_data['page_title'] = get_phrase('private_messaging');
    $this->load->view('backend/index', $page_data);
}

```

Function message holds the logical expressions for sending new messages, replying to any existing message thread between any patient who has appointments with the logged in doctor. Messages are tracked by message thread code which is defined in the send\_new\_private\_message function located in /application/models/crud\_model.php.

```

function send_new_private_message() {
    $message = $this->input->post('message');
    $timestamp = strtotime(date("Y-m-d H:i:s"));

    $reciever = $this->input->post('reciever');
    $sender = $this->session->userdata('login_type') . '-' . $this->session->userdata('login_user_id');

    //check if the thread between those 2 users exists, if not create new thread
    $num1 = $this->db->get_where('message_thread', array('sender' => $sender, 'reciever' => $reciever))->num_rows();
    $num2 = $this->db->get_where('message_thread', array('sender' => $reciever, 'reciever' => $sender))->num_rows();

    if ($num1 == 0 && $num2 == 0) {
        $message_thread_code = substr(md5(rand(100000000, 2000000000)), 0, 15);
        $data_message_thread['message_thread_code'] = $message_thread_code;
        $data_message_thread['sender'] = $sender;
        $data_message_thread['reciever'] = $reciever;
        $this->db->insert('message_thread', $data_message_thread);
    }

    if ($num1 > 0)
        $message_thread_code = $this->db->get_where('message_thread', array('sender' => $sender, 'reciever' => $reciever))->row()->message_thread_code;
    if ($num2 > 0)
        $message_thread_code = $this->db->get_where('message_thread', array('sender' => $reciever, 'reciever' => $sender))->row()->message_thread_code;

    $data_message['message_thread_code'] = $message_thread_code;
    $data_message['message'] = $message;
    $data_message['sender'] = $sender;
    $data_message['timestamp'] = $timestamp;
    $this->db->insert('message', $data_message);

    return $message_thread_code;
}

```

## Accountant Controller ( /application/controllers/accountant.php )

### Add Invoice

invoice\_add function contains the logical expressions for creating a new invoice. This is done by calling create\_invoice function which is defined in /application/models/crud\_model.php. A random invoice number is generated for each invoice. Multiple invoice entries can be created for a single invoice.

```
// Create a new invoice.
function create_invoice()
{
    $data['title'] = $this->input->post('title');
    $data['invoice_number'] = $this->input->post('invoice_number');
    $data['patient_id'] = $this->input->post('patient_id');
    $data['creation_timestamp'] = $this->input->post('creation_timestamp');
    $data['due_timestamp'] = $this->input->post('due_timestamp');
    $data['vat_percentage'] = $this->input->post('vat_percentage');
    $data['discount_amount'] = $this->input->post('discount_amount');
    $data['status'] = $this->input->post('status');

    $invoice_entries = array();
    $descriptions = $this->input->post('entry_description');
    $amounts = $this->input->post('entry_amount');
    $number_of_entries = sizeof($descriptions);

    for ($i = 0; $i < $number_of_entries; $i++)
    {
        if ($descriptions[$i] != "" && $amounts[$i] != "")
        {
            $new_entry = array('description' => $descriptions[$i], 'amount' => $amounts[$i]);
            array_push($invoice_entries, $new_entry);
        }
    }
    $data['invoice_entries'] = json_encode($invoice_entries);

    $this->db->insert('invoice', $data);
}
```

## Receptionist Controller ( /application/controllers/receptionist.php )

### Manage Appointments

```

function appointment($task = "", $doctor_id = 'all', $start_timestamp = "", $end_timestamp = "")
{
    if ($this->session->userdata('receptionist_login') != 1)
    {
        $this->session->set_userdata('last_page' , current_url());
        redirect(base_url(), 'refresh');
    }

    if ($task == 'filter')
    {
        $doctor_id      = $this->input->post('doctor_id');
        $start_timestamp = strtotime($this->input->post('start_timestamp'));
        $end_timestamp   = strtotime($this->input->post('end_timestamp'));
        redirect('index.php?receptionist/appointment/search/' . $doctor_id . '/' . $start_timestamp . '/' . $end_timestamp);
    }

    if ($task == "create")
    {
        $this->crud_model->save_appointment_info();
        $this->session->set_flashdata('message' , get_phrase('appointment_info_saved_successfully'));
        redirect('index.php?receptionist/appointment');
    }

    $data['doctor_id'] = $doctor_id;
    if($start_timestamp == '')
        $data['start_timestamp'] = strtotime('today - 30 days');
    else
        $data['start_timestamp'] = $start_timestamp;
    if($end_timestamp == '')
        $data['end_timestamp'] = strtotime('today');
    else
        $data['end_timestamp'] = $end_timestamp;

    $data['appointment_info'] = $this->crud_model->select_appointment_info($doctor_id, $data['start_timestamp'], $data['end_timestamp']);
    $data['page_name']        = 'show_appointment';
    $data['page_title']      = get_phrase('appointment');
    $this->load->view('backend/index', $data);
}

```

appointment function contains the logical expressions of showing all appointments, creating a new appointment and filtering appointment informations based on doctor and time periods. By default appointments of the last 30 days for all doctors are shown. The parameters define the form action in which the data is to be created or filtered. An sms is sent to the patient when a new appointment is created by calling send\_sms function with message and receiver phone number as parameters which is defined in /application/models/sms\_model.php.

## Approve Appointments

```

function approve_appointment_info($appointment_id)
{
    $data['timestamp'] = strtotime($this->input->post('date_timestamp').' '.$this->input->post('time_timestamp') );
    $data['status'] = 'approved';

    if($this->session->userdata('login_type') == 'receptionist')
        $data['doctor_id'] = $this->input->post('doctor_id');

    $this->db->where('appointment_id',$appointment_id);
    $this->db->update('appointment',$data);

    // Notify patient with sms.
    $notify = $this->input->post('notify');
    if($notify != '') {
        $doctor_id = $this->db->get_where('appointment',
            array('appointment_id' => $appointment_id)->row()->doctor_id;
        $patient_id = $this->db->get_where('appointment',
            array('appointment_id' => $appointment_id)->row()->patient_id;
        $patient_name = $this->db->get_where('patient',
            array('patient_id' => $patient_id)->row()->name;
        $doctor_name = $this->db->get_where('doctor',
            array('doctor_id' => $doctor_id)->row()->name;
        $date = date('l, d F Y', $data['timestamp']);
        $time = date('g:i a', $data['timestamp']);
        $message = $patient_name . ', your requested appointment with doctor ' . $doctor_name . ' on ' . $date . ' at ' . $time . ' has been approved.';
        $receiver_phone = $this->db->get_where('patient',
            array('patient_id' => $patient_id)->row()->phone;

        $this->sms_model->send_sms($message, $receiver_phone);
    }
}

```

appointment\_requested function contains the logical expressions of showing all requested appointments and approving an appointment. The parameters define the form action in which the data is to be approved. An sms is sent to the patient when a requested appointment is approved by calling send\_sms function with message and receiver phone number as parameters which is defined in /application/models/ sms\_model.php

## Login Controller ( /application/controllers/login.php )

function ajax\_login holds the logical expressions for

- matching email and password for different users
- setting the login type for different users
- setting session data which are used for other views

function reset\_password

- matches email address to saved email address
- generates a new password
- sends email to that address with the newly generated password

function logout

- unsets current session and destroys all session data saved for that session

### **The Model Files ( /application/models )**

- crud\_model.php  
contains the basic functions for creating, retrieving, updating and deleting which works in association with controllers and views.
- email\_model.php  
contains the functions that send email on different events.
- sms\_model.php  
contains the function that sends sms on different events.

### **The View Files ( /application/views )**

- backend
  - admin  
contains all the view files of admin panel
  - doctor  
contains all the view files of doctor panel
  - patient  
contains all the view files of patient panel
  - nurse  
contains all the view files of nurse panel
  - pharmacist  
contains all the view files of pharmacist panel
  - laboratorist  
contains all the view files of laboratorist panel
  - accountant  
contains all the view files of accountant panel
  - receptionist  
contains all the view files of receptionist panel
  - footer.php

- contains the footer informations of the application
- header.php
  - contains the header view of the application
- includes\_bottom.php
  - contains links to javascript files
- includes\_top.php
  - contains links to css files
- index.html
- index.php
  - includes the footer, header, javascript, css, modals at the time of page loading
- login.php
  - the login page
- modal.php
  - contains the modal views which are used in the view of different panels