System Description

For

Strathmore University Medical Centre Clinic Management System

Document Version: 1.0

Date: 18/02/2015

Contents

1 Document Management

1.1 Contributors

Please provide details of all contributors to this document

Role	Name
Production Management Support Coordinator	Joyce Munyao
Technical Architect	Alvaro Mahasi Masitsa
Project Manager	Martin Kiprop Tarus
Project Sponsor	Strathmore University
Business Analyst	Dr. Evans Almeida

1.2 Version Control

Please document all changes made to this document since initial distribution.

Date	Version	Author	Section	Amendment
18/02/15	1.0	Martin	All	Creation of an SDD document for the system at the clinic

2 OVERVIEW

SUMC is Hospital Management software specifically designed for Strathmore University. The software is integrated to provide Strathmore Medical Centre a range of modules for information management and reporting purposes.

2.1 Service Description

SUMC includes several modules that are based on hospital management functions. These modules are constantly updated and increased with the changing requirements of the clinic.
The modules implemented in this version (Version 1.0) include:

a) Reception Module
b) Nurse Module
c) Pharmacy Module
d) Doctor Module
e) Laboratory Module
f) Accounts Module
g) Administration Module
h) Dentist Module
i) Ultra sound Module
j) Physiotherapy Module

2.2 Data Model

This section should be used to provide an diagram or list of data dependencies and relationships. (e.g. Active Directory, SOA etc.)

2.3 Reporting

This section basically should have details about reports in the system.
The reports are basically all administrative and come from the information or data collected by the system.

The basic user population is defined by the number of users using the system to bill patients.

2.4 Technology

The system has been design and coded in PHP using the codeignighter framework. Mysql is the framework used as the database.

2.5 Development Tools

The software source control is performed using the GIT platform and basically the deployment of changes is tracked by the source files the git repository.

2.6 Interfaces and services

All the interfaces or modules have separate functionalities and they have all been designed to have their own folders under the modules folder found under: **application > modules**
Apart from the doctors module all the other modules functions all runs within itself, meaning that all the system modules have been designed to be able to run by itself therefore when a problem comes up you just have to access the actual module to get the details.

2.7 Access, Authentication and Authorisation

How to access the application:

On a browser the application can be accessed using the following URL:

System Access	**Url**
	192.168.170.91/hms

How authentication works:
The system has an authentication protocol which ensures that all users have to log in with a username and password to access the system.
The system has been designed to follow the following in order to manage users passwords:
The password should have the following features:

- The password must be at least 8 characters long

- You may not use any of your last 5 passwords
- It must have at least one uppercase letter and one number
- The password expires after 2 months

Basically how authorization works is that:

- A user logs in using the username and password, the password on default it 123456,
- On login the password strength is checked if it has met the set policy as shown above
- If it is correct the user is directed to the system
- If not the user is requested to change the password following the policy as above
- Then he or she can access the system

How authorisation works

All users are subject to rights given to them by the administrator of the system. The rights are based on modules that every user can access or make changes to, therefore upon creation of a user/personnel, the administrator should give the user rights to the specific modules that he or she can access.

2.8 Delivery

The delivery channel rather the system can be accessed from a browser setting therefore the site is delivered on a portal that has the system modules attached to it.

3 Support details

3.1 Third Party

The system is designed with two interfaces which basically forms part of the functionality required for the system to perform well. The two interfaces are: an interface to the students' Academic Management System and also an interface to the Human Resource system.

The location of these two interfaces are in **application > modules > reception > models > strathmore_population.php**

There are two functions namely **get_ams_student() and get_hr_staff();** snippets for the two functions are as follows:

1. **get_ams_student()**

```
public function get_ams_student($student_id = NULL){
    $conn = oci_connect('AMS_QUERIES','Oo_Hecha1_rohm3','//192.168.170.171:1522/
    ACADEMIC');
    //…….. Code continuation ……….
}
```

2. **get_hr_staff();**

```
public function get_hr_staff($staff_number = NULL)
{
        //connect to database
    $connect = mysql_connect("hr_url", "database_username", " database_password")
                    or die("Unable to connect to MySQL".mysql_error());
      //selecting a database
    mysql_select_db("hr_database_name ", $connect)
                    or die("Could not select database".mysql_error());
            //…….. Code continuation ……….
}
```

Note: For the interface to work you need the username and password to the database view or table given to the Strathmore Clinic, therefore the blanks should be filled in appropriately for a successful connection.

3.2 Documentation

The system documentation and manual is at the following locations in the systems' root folder.

System user manual	**Url**
	192.168.170.91/hms/assets/documentation/user manual.docx

3.3 Standard tasks

There are three standard tasks implemented in the system provided.

1. The system has been designed with a manual way to transfer and keep a copy of students and staff information from the students' ams view and hr view respectively into the Clinic students and staff tables.

   Note: This activity can only be done by personnel who have been given rights as an administrator in the system.

   How to perform this;

   - After logging in as an administrator
   - Click the icon and link named **administration** on the dashboard
   - On the side bar links Click the main menu **Administration** and sub menus should show.
   - Click on the **import student/staff** link
   - For staff import click on the staff button and for students import click on the students button.

   What this function does:

   This function basically helps in equipping the clinic database with the updated information of all students and staff as is in their respective systems in Strathmore University. Therefore with this information at hand, the probabilities of missing a student or staff record during registration of their records will be reduced.

2. The second standard task is a data clean-up task that will ensure the database has cleaned tables in terms of data replications. This activity involves a user who has access to the database as it involves running SQL's to check for any patients duplicates.

- **Staff duplicate query:**

```
SELECT s.Staff_Number, oc.dupeCount, s.staff_id, s.Surname, s.Other_names
from staff s
inner join (
   SELECT strath_no, COUNT(*) AS dupeCount
   FROM patients
            WHERE patients.dependant_id = 0
   GROUP BY strath_no
   HAVING COUNT(*) > 1
) oc on s.Staff_Number = oc.strath_no
```

- **Students duplicate query:**

```
select s.student_Number, oc.dupeCount, s.student_id
from student s
inner join (
   SELECT strath_no, COUNT(*) AS dupeCount
   FROM patients
            WHERE patients.dependant_id = 0
   GROUP BY strath_no
   HAVING COUNT(*) > 1
) oc on s.student_Number = oc.strath_no
```

NOTE: The two SQLs should help you determine the duplicate rows in the patients table which in turn will help you in updating the visit table in cases there are some visits opened

with the duplicate patient_id's. Ideally there should only be one record of a patient in the patients table corresponding to the same patient id in the visit table
Use the following SQL to update the duplicates on the visit table

UPDATE visit SET patient_id = '' WHERE patient_id = '' OR patient_id = '' ;

Ensure that you have deleted the patient id in duplicates to resolve the duplicate issues.

3.4 Troubleshooting Guide

The section should provide advice on how to troubleshoot issues that might come up from the system. The following are the items key when issues come up:

- **Known accessibility issues:**
  The system has been tested and works well on **Mozilla Firefox** V.19 upwards and **Google Chrome**. However the Internet Explorer should work though the latest version.

- **Where to find error messages:**
  The Codeignighter framework runs on an MVC framework which basically would assist in identifying where the errors could have come from; the basic errors that may come up are:

  a. URL or page error
  This is an error caused when the base or config url has not been defined correctly. Basically if this is not configured correctly the system will not be accessed or even opened in the first place.

  NOTE: Therefore this should basically contain the actual url that directs any request to server of the system.

  The file to look up for is under: **application > config > config.php**

  <?php

  code…….

  $config['base_url'] = 'http://server address here / folder name here';

  ……code

?>

b. Database connection error:

The file to check out on this should be found under: **application > config > database.php**

NOTE: Find the lines below and ensure that the username, password and the database name are correctly entered.

```
<?php
        code .....
        $db['default']['hostname'] = 'localhost';
        $db['default']['username'] = '';
        $db['default']['password'] = '';
        $db['default']['database'] = '';
        ……code
?>
```

c. Page not found error:

This error could have two implications either:

- The page might not be existing
- The URL being searched for cannot be found on the routes of the system
  - **Page not existing**

    There is not solution to this error because the page might not have been created on development therefore the user is required to enter a correct URL or select another link.
  - **URL not being found on the routes file in the system**

The system has a file that contains all the URL routes for the system. The main use of this routes file is actually reducing the vulnerability of hackers knowing the names of the file containing the codes. This file is found under: **application > config > routes.php**

<?php

code ……..

$route['login'] = 'auth/login/login_user';

$route['logout'] = 'auth/login/logout_user';

$route['control-panel/(:num)'] = 'auth/control_panel/$1';

……code

?>

Search for the route the url is being directed to and ensure that have it applied in the file correctly.

NOTE: After the = sign is the actual route where the request should be routing to.

- **Special debugging instructions**

  The system could probably have issues of access at times however much they were previously accessed. The procedure of debugging should be as follows:

  - Check whether the server can be accessed, this can be done by pinging the url of the server from the machine that was not able to access the system.
    - This can be done by getting the Command prompt running then enter the following :
      - *ping server url*
  - If the URL pings well then check whether the browser proxy settings has been set. This can be done on Mozilla by going to the **main menu > tools > network > settings**

    The proxy setting by default should not have been configured; rather the state of this should be no proxy.
  - If the URL does not ping ideally the computer at hand is not on the network therefore the network connection should be checked.
  - If it's an access issue the user could have forgotten the password or his or her account has been deactivated by the administrator. Therefore the administrator can be able to solve this issue by either resetting the password or activating the user.

4 Document Sign Off

Strathmore Clinic Representative	*Name:*	*Date*
	Signature:	
Strathmore ITD Representative	*Name*	*Date*
	Signature	
MAJ group Representative	*Name*	*Date*
	Signature	