

**Programming Applications and Frameworks (IT3030)**

**3rd Year, 1st Semester**

**2020**

**Hospital Management System**

**Feedback Report**

Submitted by: **Group: S1142.1**

* IT18162974 - Lakshan P.A.D
* IT18112474 - Rodrigo M.N.D
* IT18185126 - Mallawarachchi S.N
* IT18136234 - Sathsarani B.G.K
* IT18167160 - Watthuhewa M.P

**Table of Contents**

Methodology 1

Version Control Mechanism---------------------------------------------------------------------1.1

Group & work distribution 2

Requirements 3

Stakeholder analysis------------------------------------------------------------------------------ 3.1

Requirement analysis------------------------------------------------------------------------------ 3.2

**Functional ----------------------------------------------------------------------------------------------3.2.1**

**Non-functional----------------------------------------------------------------------------------------3.2.2**

**Technical-------------------------------------------------------------------------------------------------3.2.3**

Requirement modeling-----------------------------------------------------------------------------3.3

System Design 4

Overall Architecture------------------------------------------------------------------------------4.1

Database design-------------------------------------------------------------------------------------4.2

Workflow design-----------------------------------------------------------------------------------4.3

Functions 5

Patient management ------------------------------------------------------------------------------5.1

Hospital management-----------------------------------------------------------------------------5.2

Doctor management ------------------------------------------------------------------------------5.3

Appointment management -----------------------------------------------------------------------5.4

Payment management -----------------------------------------------------------------------------5.5

Gantt chart 6

References 7

Appendix 8

1. Methodology

n E healthcare system will be implemented by the project. It is a hospital management system which allows three user levels in the system. Patient, doctor and hospital will be the three user levels above mentioned.

To use the currently implementing system, user must be registered to the system. Reports will be generated by the system, according the requirements of the user levels. Registered users can make appointments with the registered doctors who visit the registered hospitals. The users can even make the payments for the appointments online.

Version Control

2.Group and Work Distribution Details

|  |  |  |  |
| --- | --- | --- | --- |
| Name | IT Number | Function | Description |
| Lakshan P.A.D. | IT18162974 | Hospital | * Add schedule. * update schedule. * delete schedule. * Update profile. * Delete profile. * View the hospital details. |
| Rodrigo M.N.D | IT18112474 | Appointment | * Add the appointment details. * Edit the appointment. * Cancel appointments. * View the appointment details. |
| Watthuhewa M. P. | IT18167160 | Patient | * Add personal details * Update personal details * Delete the profile * Add appointment * View the patient details. |
| Sathsarani B.G.K | IT18136234 | Payment | * Add payment details. * Update the payment details. * Delete the payment records. * View the payment profile. |
| Mallawarachchi S.N | IT18185126 | Doctor | * Add doctor details. * Update doctor details. * Add patient   reports details**.**   * Update patient reports**.** * View the doctor details. |

3. Requirements

3.1 Stakeholder Analysis

**UI**

**DB**

**Testing**

* **Patient API**
* **Doctor API**
* **Hospital API**
* **Appointment API**
* **Payment API**
* **Patient Management**
* **Doctor**

**Management**

* **Hospital Management**
* **Appointment Management**
* **Payment Management**

4.2 Requirement Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality | Functional Requirements | Non-Functional  Requirements | Technical Requirements |
| Patient Management | * Should be able to register as a new user. * Can receive a schedule of doctors and hospitals. * Can register an appointment and make payments online. * Allow to edit their profile. * Should be able to provide a feedback about the system. * Should be able to have printouts according to the requirements. | * Security * Usability * Privacy * Reliability * Efficiency | * Patient registering to the system as a new user. * After validating the account of the user, update, delete and view patient details. * Connecting with appointment class to register an appointment. * Connecting with payment to make a payment. |
| Doctor Management | * Should be able to register as a new user. * Can receive a list of schedules for the required time period. * Can receive a list of patients. * Should be able to have printouts according to the requirements. | * Security * Usability * Privacy * Reliability * Efficiency | * Doctor registering to the system as a new user. * After validating the account of the user, update, delete and view doctor details. * Connecting with appointment class to receive a schedule. * Connecting with hospital class to create a list of hospital with doctor IDs. * Connecting with payment class to finalize the doctor charges. |
| Hospital Management | * Should be able to register as a new user. * Can receive a list of schedules for the required time period. * Can receive a list of patients and doctors. * Should be able to have printouts according to the requirements. | * Security * Usability * Privacy * Reliability * Efficiency | * Hospital registering to the system as a new user. * After validating the account of the user, update, delete and view doctor details. * Connecting with hospital class to create a list of hospital with doctor IDs. * Connecting with payment class to finalize the hospital charges. |
| Appointment Management | * Patient should be able to schedule an appointment after registering to the system with valid username and password. * Patient should be able to receive a schedule about appointments. | * Security * Usability * Privacy * Reliability * Efficiency | * Patient will register an appointment after signup. * Admin will manage appointments and connect with patient, doctor and payment to complete the task of registering an appointment. * After validating the account of the user, update, delete and view doctor details |
| Payment Management | * A registered patient should be able to make an online payment. * User should be able to receive a bill as a printout. | * Security * Usability * Privacy * Reliability * Efficiency | * Connect with patient, doctor, appointment, and hospital to complete calculating the total fee. * Provide payment methods to use. * After validating the admin account of the user, update, delete and view doctor details. |

4.3 Requirement Modeling

* Use Case Diagram

A picture containing text, map

Description automatically generated

5.System Design

5.1 Overall Architecture

A screenshot of a cell phone

Description automatically generated

5.2 Database Design

A close up of a map

Description automatically generated

5.3 Workflow Diagram

6.Function Description

6.1 Hospital Management: IT18162974 (Lakshan P.A.D)

Service design

In this Healthcare system there are five major parts. Hospital service is also included into that part. Through this API system can add, remove, update and view the hospitals in the system. In this API there are two parts they are called model and service. Model represent how those insert update delete work and the service class include how to call those methods.

* In order to work word from these word from these be used HTTPs request as put post get delete

PUT-To create resources

POST-update resources

GET- get resources or list of resources

DELETE- to delete resources

* And, there are some attributes to represent the hospital. They are,

Hospital ID as HosId (int-primary key),

Hospital name as HosName (String),

Hospital City HosCity (String),

Number of Rooms as Rooms (int),

Hospital address as Address (String) and Hospital contact number as ContactNum (int)

Service development and testing

Dependency management tools -Maven

IDE-Eclipse

Database-MySql

Back End- Java - JAX-RS (Jersy)

Testing Tools-PostmanCode quality checking tools-Sonarcube

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description/ Test Steps** | **Test**  **Input(s)** | **Expected Output(s)** | **Actual**  **Output(s)** | **Result**  **(Pass/Fail)** |
| 01 | Insert Hospital details. | * HosId-1 * HosName -Asiri * HosCity-Narahenpita * Rooms-500 * Address-Kirimandala Mw,Narahenpita * ContactNum-011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |
| 02 | Update Hospital Details for a selected user. | Click Update in HosID=1’s profile  Updating Number of rooms in HosID 1 INTO 600 | Updated Messageand change 500 to 600 | Updated Messageand change 500 to 600 | pass |
| 03 | Delete a Hospital | Click Delete in HosID=1’s profile | Deleted Profile HosID=1 | Deleted Profile HosID=1 | pass |
| 04 | View Hospital Details | Click View for HosID=1 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |

6.2 Patient Management: IT18167160 ( Watthuhewa M.P)

Service design

Service development and testing

Dependency management tools -Maven

IDE-Eclipse

Database-MySql

Back End- Java - JAX-RS (Jersy)

Testing Tools-Postman

Code quality checking tools-Sonarcube

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description/ Test Steps** | **Test**  **Input(s)** | **Expected Output(s)** | **Actual**  **Output(s)** | **Result**  **(Pass/Fail)** |
| 01 | Insert Hospital details. | * HosId-1 * HosName -Asiri * HosCity-Narahenpita * Rooms-500 * Address-Kirimandala Mw,Narahenpita * ContactNum-011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |
| 02 | Update Hospital Details for a selected user. | Click Update in HosID=1’s profile  Updating Number of rooms in HosID 1 INTO 600 | Updated Messageand change 500 to 600 | Updated Messageand change 500 to 600 | pass |
| 03 | Delete a Hospital | Click Delete in HosID=1’s profile | Deleted Profile HosID=1 | Deleted Profile HosID=1 | pass |
| 04 | View Hospital Details | Click View for HosID=1 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |

6.2 Doctor Management: IT18185126 (Mallawarachchi S.N)

Service design

In health people system user has to first login into the system. If she/he hasn’t registered yet first they have to register to the system giving the details (Name, specialized, NIC, registered number, etc.….). After the user login. User can see the appointment and check the patient details. After checking the patient doctor can enter the prescription to the system when the patient visit on other time doctor can update the report of the patient after checking the patient. Then doctor get the promotions and specialized from another side doctor can update the she/he is profile. When the doctor resign admin can delete, she/he is profile.

Service development and testing

Dependency management tools -Maven

IDE-Eclipse

Database-MySql

Back End- Java - JAX-RS (Jersy)

Testing Tools-Postman

Code quality checking tools-Sonarcube

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description/ Test Steps** | **Test**  **Input(s)** | **Expected Output(s)** | **Actual**  **Output(s)** | **Result**  **(Pass/Fail)** |
| 01 | Insert Hospital details. | * HosId-1 * HosName -Asiri * HosCity-Narahenpita * Rooms-500 * Address-Kirimandala Mw,Narahenpita * ContactNum-011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |
| 02 | Update Hospital Details for a selected user. | Click Update in HosID=1’s profile  Updating Number of rooms in HosID 1 INTO 600 | Updated Messageand change 500 to 600 | Updated Messageand change 500 to 600 | pass |
| 03 | Delete a Hospital | Click Delete in HosID=1’s profile | Deleted Profile HosID=1 | Deleted Profile HosID=1 | pass |
| 04 | View Hospital Details | Click View for HosID=1 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |

6.4 Payment Management

In this system user first login to the system. Without login, user can't make an appointment. After successful login, user must make an appointment according to availability of doctors. After that system will display payment page. Then user have to fill visa or master card details (card type, card no, card expiry date and card cvc number) according to the payment process. Finally, user can make payment successfully after providing valid information to the system.

Service development and testing

Dependency management tools -Maven

IDE-Eclipse

Database-MySql

Back End- Java - JAX-RS (Jersy)

Testing Tools-Postman

Code quality checking tools-Sonarcube

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description/ Test Steps** | **Test**  **Input(s)** | **Expected Output(s)** | **Actual**  **Output(s)** | **Result**  **(Pass/Fail)** |
| 01 | Insert Hospital details. | * HosId-1 * HosName -Asiri * HosCity-Narahenpita * Rooms-500 * Address-Kirimandala Mw,Narahenpita * ContactNum-011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |
| 02 | Update Hospital Details for a selected user. | Click Update in HosID=1’s profile  Updating Number of rooms in HosID 1 INTO 600 | Updated Messageand change 500 to 600 | Updated Messageand change 500 to 600 | pass |
| 03 | Delete a Hospital | Click Delete in HosID=1’s profile | Deleted Profile HosID=1 | Deleted Profile HosID=1 | pass |
| 04 | View Hospital Details | Click View for HosID=1 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |

6.5 Appointment Management

Appointment function will be used by the user level patient and will send data to the user level doctor. The user should have a valid account to use this function. Once the user entered relevant details to the appointment interface, system will send the data to the central database. Then the system will display a list of scheduled appointments for the required time period. Patient can choose a hospital and a doctor and submit the details. System will generate an invoice with relevant charges. Patient has to pay and receive a notice to confirm the appointment. Confirmation notice will be provided as a report that is generated from the system.

Service development and testing

Dependency management tools -Maven

IDE-Eclipse

Database-MySql

Back End- Java - JAX-RS (Jersy)

Testing Tools-Postman

Code quality checking tools-Sonarcube

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Description/ Test Steps** | **Test**  **Input(s)** | **Expected Output(s)** | **Actual**  **Output(s)** | **Result**  **(Pass/Fail)** |
| 01 | Insert Hospital details. | * HosId-1 * HosName -Asiri * HosCity-Narahenpita * Rooms-500 * Address-Kirimandala Mw,Narahenpita * ContactNum-011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |
| 02 | Update Hospital Details for a selected user. | Click Update in HosID=1’s profile  Updating Number of rooms in HosID 1 INTO 600 | Updated Messageand change 500 to 600 | Updated Messageand change 500 to 600 | pass |
| 03 | Delete a Hospital | Click Delete in HosID=1’s profile | Deleted Profile HosID=1 | Deleted Profile HosID=1 | pass |
| 04 | View Hospital Details | Click View for HosID=1 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | * 1 * Asiri * Narahenpita * 500 * Kirimandala Mw,Narahenpita * 011234234 | pass |

6. Gantt Chart

7. References

8. Appendix

* As an integration tool, we used git hub.

A screenshot of a cell phone

Description automatically generated

* Git hub repository link:

[https://github.com/PAF-00/S1142.1](https://github.com/NipunikaRodrigo/PAF-00/S1142.1)

A close up of text on a white background

Description automatically generated

A close up of a map

Description automatically generated

![A picture containing drawing

Description automatically generated]()A close up of a logo

Description automatically generated

A close up of a map

Description automatically generated