**6.0 Design Phase**

**6.1 Design the Application Architecture**

Because of two key reasons, the application architecture is straightforward. First, because the specifications do not state that external user access is required, Internet capability is not included. The use of intranet is the second argument for basic architecture. The system is intranet because we will greatly lower our exposure to cyber security risk by limiting the system's access scope to the bounds of the clinic's premises.

The end user (i.e., provider) will utilize the tablets to connect to the server via the router and access the program. The database will be housed on a separate server that will use SQL Server as the database management system. If more servers or storage are needed in the future, two servers will be deployed to boost performance and maximize capacity.

**A picture containing square

Description automatically generatedA picture containing square

Description automatically generatedA picture containing square

Description automatically generated**

Tablets

**A picture containing electronics, computer

Description automatically generated**A picture containing text, device

Description automatically generated

A picture containing text, device

Description automatically generatedA picture containing text

Description automatically generated

A picture containing text, device

Description automatically generated

Router

Server with Application

**A picture containing electronics, computer

Description automatically generated**

**Icon

Description automatically generated**

Server with Database

**6.1.1 Networks – Intranet**

* The user will use tablets to access the application via router.
* The servers will not be exposed to external connection beyond each other and tablets.
* The router will be used for communication between servers and tablets only.

**6.1.2 Database Distribution**

* One server will be used to house the database.
* The server will be located on site of the clinic.
* Servers will be accessed using the router.

**6.1.3 “Off the Shelf” Software**

SQL Server will be used as the database management system to administer the database.

**6.1.4 User Interface Technology – With Other Users**

No other users besides the provider will interface with the technology.

**6.1.5 System Interface Technology – With other Systems**

The system will not interface with any other system.

**6.2 Models for Hospital Bed Management System**

**6.2.1 Context Model for Hospital Bed Management System**

Diagram

Description automatically generated

**6.2.2 Data Flow Diagram**

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generatedDiagram

Description automatically generated

**6.2.3 Software Structure Chart for Module Hierarchy**

**6.2.4 Use Case**

Diagram

Description automatically generated

**6.2.5 Use Case Narratives for Hospital Bed Management System**

|  |  |  |  |
| --- | --- | --- | --- |
| **Hospital Bed Management System (HBMS)** | | | |
| Authors: Harsh Savaliya, John Dempsey and Heather Dempsey | | | **Date: 06/03/2023** |
|
| **Version: 1.0** |
| **Use-Case Name:** | System Access | | **Use-Case Type Business Requirements:** |
| **Use-Case ID:** | HBMS-1.00 | |
| **Priority:** | High | |
| **Source:** | Provider | |
| **Primary Business Actor:** | Provider | | |
| **Other Participating Actors:** | N/A | | |
| **Other Interested Stakeholders:** | Patient, Referrer | | |
| **Description:** | This use-case describes the necessary steps for the provider to logon and access the HBMS system. | | |
| **Precondition:** | N/A | | |
| **Trigger:** | This use case is initiated by the immediate need to access the system. | | |
| **Typical Course of Events:** | **Actor Action** | **System Response** | |
| **Step 1:** Provider accesses login screen. **Step 2:** Provider enters login credentials including username and password. | **Step 3:** System authenticates credentials through secure database. **Step 4:** Access granted taken to main system page. | |
| **Alternate Courses:** | **Alt-Step 2:** Provider does not remember credentials. Taken to reset.  **Alt-Step 4:** Access denied. Locked after six incorrect attempts. | | |
| **Conclusion:** | This use case concludes when access to HBMS system is granted to provider. | | |
| **Postcondition:** | Full system access granted. | | |
| **Business Rules:** | N/A | | |
| **Implementation Constraints and Specifications:** | Provider has full view privileges. | | |
| **Assumptions:** | Credentials are valid. | | |
| **Open Issues:** | N/A | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Hospital Bed Management System (HBMS)** | | | |
| Authors: Harsh Savaliya, John Dempsey and Heather Dempsey | | | **Date: 06/03/2023** |
|
| **Version: 1.0** |
| **Use-Case Name:** | Update Patient DB with Patient Personal Information | | **Use-Case Type Business Requirements:** |
| **Use-Case ID:** | HBMS-2.00 | |
| **Priority:** | High | |
| **Source:** | Provider | |
| **Primary Business Actor:** | Provider | | |
| **Other Participating Actors:** | Patient | | |
| **Other Interested Stakeholders:** | Provider, Patient | | |
| **Description:** | This use case describes the event of the provider updating patient personal information DB via the application. If the patient is new, the provider will create a new profile for the patient. Once the patient is in the system, the provider can modify any of the patient's personal information. | | |
| **Precondition:** | N/A | | |
| **Trigger:** | This use case is initiated when there is a new patient or an existing patient's personal information needs to be modified. | | |
| **Typical Course of Events:** | **Actor Action** | **System Response** | |
| **Step 1:** Patient provides his or her personal information.  **Step 2**: Provider determines if the patient is new or existing. **Step 3**: For new patient, the provider creates a new patient profile by clicking "New Patient". **Step 5:** Provider inputs patient's personal information and selects programs the patient is to be enrolled in. The provider then submits the information. **Step 6:** For existing patients, provider opens desired patient's profile, modifies information, and then submits the information. | **Step 4:** The system responds by blank new profile for data input and Patient Account DB is updated with new patient's information. **Step 7:** Existing patient information is displayed, and changes are updated in the Patient Account DB. | |
| **Alternate Courses:** | **Alt-Step 8:** If new profile is created, notify the provider of missing information in required input fields. **Alt-Step 9:** If existing patient profile is modified, notify the provider of missing information in required input fields. **Alt-Step 10:** Do not process any new patient form with missing information. If existing patient, restore the information to last valid updated information. | | |
| **Conclusion:** | This use case concludes when all information of a desired patient is filled out on the forms and provider submits the form. | | |
| **Postcondition:** | **1.** All information is updated in the Patient Account DB.  **2.** Forms with missing information are not processed and provider is notified of empty fields. | | |
| **Business Rules:** | N/A | | |
| **Implementation Constraints and Specifications:** | **1.** The patient personal information will be gathered using a form format GUI. **2.** The GUI must have a logical flow for simple navigation and ease of use, yet comprehensive enough to include all vital patient information. | | |
| **Assumptions:** | **1.** The provider will ensure that all information is legally valid. | | |
| **Open Issues:** | N/A | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Hospital Bed Management System (HBMS)** | | | |
| Authors: Harsh Savaliya, John Dempsey and Heather Dempsey | | | **Date: 06/03/2023** |
| **Version: 1.0** |
| **Use-Case Name:** | Update Patient Medical Account DB | | **Use-Case Type Business Requirements:** |
| **Use-Case ID:** | HBMS-3.00 | |
| **Priority:** | High | |
| **Source:** | Provider | |
| **Primary Business Actor:** | Provider | | |
| **Other Participating Actors:** | Patient | | |
| **Other Interested Stakeholders:** | N/A | | |
| **Description:** | This use case describes the event for updating the Patient Medical Account DB via the application at every interval. In addition, this use case will also describe how patient's medical reports will be run. These reports will be based on the medical information stored in the Patient Medical Account DB and will allow the provider to assess the health and progress of the patient through the medical treatment plans. | | |
| **Precondition:** | Patient's personal information must be filled out before medical information can be updated. | | |
| **Trigger:** | New patient arrives at the clinic or existing patient reports for a follow-up. | | |
| **Typical Course of Events:** | **Actor Action** | **System Response** | |
| **Step 1.** New patient arrives at the clinic and their new profile is created. In addition, personal information is updated. **Step 3.** Provider inputs medical history and submits the information. **Step 5.** Patient arrives at the clinic on a follow-up basis. For each patient, new medical metrics are reported. **Step 7**. Provider runs medical reports for patients and groups. | **Step 2.** The system prompts for patient medical history.  **Step 4.** Submitted information is send to the Patient Medical Account DB. **Step 6.** System provides forms for each treatment plan. Each form has empty input fields for medical metrics. **Step 8.** System provides pre-defined medical reports for patient. Each report is associated with a specific treatment plan. | |
| **Alternate Courses:** | **Alt-Step 9.** If patient does not show up for a follow-up, the provider must contact the patient and arrange for him/her to come in. **Alt-Step 10.** If medical metrics are incorrectly updated, provider must edit the metrics. | | |
| **Conclusion:** | This use case concludes when the provider either inputs medical history, medical metrics, or runs reports. | | |
| **Postcondition:** | N/A | | |
| **Business Rules:** | **1.** It is the responsibility of the provider to ensure all medical metrics are correctly reported. **2.** If medical metrics are incorrect, the provider must correct them immediately. **3.** The provider is responsible for correctly interpreting the medical metrics and health of the patient through the treatment plan. 4**.** The provider must communicate to the patient, the health and performance of the patient. | | |
| **Implementation Constraints and Specifications:** | **1.** GUI for taking patient medical history, medical metrics, and running reports must be easy to navigate. **2.** All forms for taking medical history and medical metrics must be predefined for each treatment plan. **3.** For each patient, forms should be displayed only according to the treatment plan the patient is enrolled in. **4.** All medical reports must be associated with a treatment plan and each report must be predefined. | | |
| **Assumptions:** | **1.** The provider will accurately interpret medical metrics to gauge the health and performance of the patient.  **2.** The provider will communicate the patient, their health and progress. | | |
| **Open Issues:** | N/A | | |

|  |  |  |
| --- | --- | --- |
| **Hospital Bed Management System (HBMS)** | | |
| Authors: Harsh Savaliya, John Dempsey and Heather Dempsey | | **Date: 06/03/2023** |
|
| **Version: 1.0** |
| **Use-Case Name:** | Manage Follow-Up | **Use-Case Type Business Requirements:** |
| **Use-Case ID:** | HBMS-4.00 |
| **Priority:** | High |
| **Source:** | Provider |
| **Primary Business Actor:** | Provider | |
| **Other Participating Actors:** | Patient | |
| **Other Interested Stakeholders:** | Provider, Patient | |
| **Description:** | This use case describes the event of the provider managing scheduled follow-up appointments via the application, for the patient to return to the clinic at a future date. | |
| **Precondition:** | Patient has completed initial visit. | |
| **Trigger:** | This use case is initiated when a patient completes initial visit and must schedule a return date, or a pre-scheduled date is no longer feasible, and the patient needs to reschedule. | |
| **Typical Course of Events:** | **Actor Action** | **System Response** |
| **Step 1:** Provider selects follow-up tab. **Step 3:** Provider determines whether to create new follow-up or modify existing.  **Step 4:** For new patient follow-up, provider clicks "Add New Follow-Up". **Step 6:** Provider enters patient first and last name and desired return date. **Step 8:** For change to an existing date, provider selects "Edit" from list of active patients corresponding with the patient requesting a different date. **Step 10:** Provider verifies patient name displayed is correct and modifies date field to match desired patient return date.  **Step 11:** Provider clicks "Submit" button. | **Step 2:** Screen displays list of active patients as well as a button to "Add New Follow-Up".  **Step 5:** Form displays with fields to enter patient first and last name and date.  **Step 7:** Patient name and date saved to follow-up database. **Step 9:** Edit form opens displays selected patient first and last name and modifiable date field. **Step 12:** Patient name and modified date saved to follow-up database. |
| **Alternate Courses:** | **Alt-Step 2:** If patient is not listed, profile was not created must return to "Patient" tab.  **Alt-Step 10:** If patient displayed information is incorrect, provider clicks "Cancel" button.  **Alt-Step 12:** Return to list of active patients scheduled for follow-up as well as the "Add New Follow-Up" button. | |
| **Conclusion:** | This use case concludes when provider submits the patient’s desired follow-up date. | |
| **Postcondition:** | Patient's information and scheduled return date saved to follow-up database. Available for retrieval. | |
| **Business Rules:** | N/A | |
| **Implementation Constraints and Specifications:** | GUI to be used by provider displays saved active patients. Edit form and follow-up form will be auto populated from patient database. Date field is only modifiable field on edit form. Follow-up form will have a drop-down of existing patients. | |
| **Assumptions:** | Patient has completed initial visit and profile has been created and saved to patient database. | |
| **Open Issues:** | N/A | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Hospital Bed Management System (HBMS)** | | | | | | | |
| Authors: Harsh Savaliya, John Dempsey and Heather Dempsey | | | | | | **Date: 06/03/2023** | |
|
| **Version: 1.0** | |
| **Use-Case Name:** | Update Patient Treatment Database | | | | | **Use-Case Type Business Requirements:** | |
| **Use-Case ID:** | HBMS-5.00 | | | | |
| **Priority:** | High | | | | |
| **Source:** | Provider | | | | |
| **Primary Business Actor:** | Provider | | | | | | |
| **Other Participating Actors:** | Patient | | | | | | |
| **Other Interested Stakeholders:** | Provider, Patient | | | | | | |
| **Description:** | This use-case is necessitated by the event a healthcare provider wants to view or add to existing treatment database. | | | | | | |
| **Precondition:** | N/A | | | | | | |
| **Trigger:** | This use case is initiated when a healthcare provider has need to view and/or update existing treatment information with new treatment concerns or corollary information. | | | | | | |
| **Typical Course of Events:** | **Actor Action** | | | **System Response** | | | |
| **Step 1:** Provider selects "Treatment" tab.  **Step 3:** Provider determines which diagnosis is of concern. **Step 4:** Provider selects "View" link from desired diagnosis line. **Step 6:** Provider selects "Edit" to add or remove treatment information.  **Step 8:** Provider selects "Submit" after desired information has been modified. | | | **Step 2:** Screen displays list of predetermined diagnosis by Id and name. **Step 5:** Form open to selected diagnosis listing treatment information. **Step 7:** Form will display with modifiable input fields. **Step 9:** Inputted information is saved to patient treatment database. | | | |
| **Alternate Courses:** | **Alt-Step 3:** Provider determines new treatment is needed.  **Alt-Step 4:** Provider selects "Add New Treatment" link located after diagnosis id on appropriate line.  **Alt-Step 5:** System responds by opening form to annotatable field for treatment id, name and provider's notes. "Submit" button saves treatment to database. | | | | | | |
| **Conclusion:** | This use case concludes when desired changes have been entered and "Submit" button has been depressed and information has been written to patient treatment database. | | | | | | |
| **Postcondition:** | Treatment database has been updated and reflects most current information available. | | | | | | |
| **Business Rules:** | N/A | | | | | | |
| **Implementation Constraints and Specifications:** | Updates cannot be written concurrently. One provider must finish sequence of updating before another input can be appended. | | | | | | |
| **Assumptions:** | Healthcare provider is writing accurate, up-to-the minute treatment breakthroughs in the medical community as it pertains to each diagnosis. | | | | | | |
| **Open Issues:** | N/A | | | | | | |
| **Hospital Bed Management System (HBMS)** | | | | | | |
| Authors: Harsh Savaliya, John Dempsey and Heather Dempsey | | | | | **Date: 06/03/2023** | |
|
| **Version: 1.0** | |
| **Use-Case Name:** | | Update Referrer List | | | **Use-Case Type Business Requirements:** | |
| **Use-Case ID:** | | HBMS-6.00 | | |
| **Priority:** | | High | | |
| **Source:** | | Provider | | |
| **Primary Business Actor:** | | Provider | | | | |
| **Other Participating Actors:** | | Patient | | | | |
| **Other Interested Stakeholders:** | | Provider, Referrer | | | | |
| **Description:** | | This use-case is in response to the event the referring doctor needs to be added or updated in the referring doctor database. Either the referrer is new or his/her contact information has changed. | | | | |
| **Precondition:** | | N/A | | | | |
| **Trigger:** | | This use case is initiated when a patient is referred by a new doctor to HBMS system or the referring doctor’s information is no longer current and needs updated. | | | | |
| **Typical Course of Events:** | | **Actor Action** | **System Response** | | | |
| **Step 1:** Provider determines whether referring doctor is new or exists with outdated information. **Step 2:** Provider selects "Referring Doctor" tab. **Step 4:** Provider selects "Add New" referrer. **Step 6:** Provider enters new referrer’s information and clicks "Submit". | **Step 3:** Screen displays list of active referring doctors along with "Add New" button.  **Step 5:** Form opens up with pertinent contact, name and affiliation input fields.  **Step 8:** Inputted information is saved to referring doctor database. **Step 9:** Form closes. | | | |
| **Alternate Courses:** | | **Alt-Step 4:** Provider selects "Edit" link listed after corresponding referrer whose information needs to be updated. **Alt-Step 5:** Form displays with modifiable fields allowing changes to be made.  **Alt-Step 6:** Provider makes changes and clicks "Submit".  **Alt-Step 7:** Changes are written and saved to referring doctor database. Form closes. | | | | |
| **Conclusion:** | | This use case concludes when new or updated information has been submitted and written to referring doctor database. | | | | |
| **Postcondition:** | | Referring doctor database reflects most current and accurate information available. | | | | |
| **Business Rules:** | | N/A | | | | |
| **Implementation Constraints and Specifications:** | | Once referring doctor information has been written and submitted the pre-existing contact information is no longer available. | | | | |
| **Assumptions:** | | Only one provider may annotate doctor’s personal information at a time. The most recent addition will supersede the previous. | | | | |
| **Open Issues:** | | N/A | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Hospital Bed Management System (HBMS)** | | | |
| Authors: Harsh Savaliya, John Dempsey and Heather Dempsey | | | **Date: 06/03/2023** |
|
| **Version: 1.0** |
| **Use-Case Name:** | Update Referring Doctor Database | | **Use-Case Type Business Requirements:** |
| **Use-Case ID:** | HBMS-7.00 | |
| **Priority:** | High | |
| **Source:** | Provider | |
| **Primary Business Actor:** | Provider | | |
| **Other Participating Actors:** | Patient | | |
| **Other Interested Stakeholders:** | Referrer, Patient | | |
| **Description:** | This use case allows the provider to search existing referrers name, affiliation and recommended treatments for the patient or to add new physician profile to the referring doctor information database. | | |
| **Precondition:** | Referrer has recommended patient for evaluation. | | |
| **Trigger:** | This use case is initiated when referral is received by provider for new patient intake. | | |
| **Typical Course of Events:** | **Actor Action** | **System Response** | |
| **Step 1:** Provider selects system tab "Referring Doctors". **Step 3:** Provider determines if referrer is new or existing. **Step 4:** For new referrer information, provider clicks "Add New Doctor" button.  **Step 6:** Providers reviews information that was entered is correct. **Step 7:** Provider clicks "Submit" button. **Step 9:** For existing referrers, provider selects appropriate line displaying physician’s name.  **Step 11:** Desired information entered to reflect the most current available.  **Step 12:** Provider clicks "Submit". | **Step 2:** Screen displays list of participating doctors' first and last name as well as phone, email, address and hospital affiliations. There is "Add New Doctor" button available.  **Step 5:** Form displays with input fields for doctor contact and affiliation information.  **Step 8:** New doctor profile is saved to referring doctor information database.  **Step 10:** The saved treatment recommendations and doctor's profile are displayed with modifiable input fields.  **Step 13:** Updated information saved to referring doctor database. | |
|
| **Alternate Courses:** | **Alt-Step 6:** If information is incorrectly displayed provider clicks "Cancel" button.  **Alt-Step 9:** If existing referrer information is incorrect, provider clicks "Edit" after that referrer's name, brought to form to re-edit profile. | | |
| **Conclusion:** | This use case concludes when referring doctor information has been updated written to database and form closes. | | |
| **Postcondition:** | Referring doctors' most current information saved for retrieval. | | |
| **Business Rules:** | N/A | | |
| **Implementation Constraints and Specifications:** | GUI assists provider in adding, updating and retrieving doctors' contact and affiliation, who have or will refer patients to system. Referrers are limited to only to physicians licensed to practice medicine in the United States. | | |
| **Assumptions:** | Provider will be informed of referring doctor prior to patient’s intake into system. Referring physicians are licensed to practice medicine in the United States. | | |
| **Open Issues:** | N/A | | |

**6.2.6 Sequence Diagrams**

**HBMS Access**

Diagram

Description automatically generated

**Sequence Diagram for Patient Enrollment**

Diagram

Description automatically generated

**Sequence Diagram for Updating Patient Personal Information**

Diagram

Description automatically generated

**Sequence Diagram for Updating Patient Medical Information**

Diagram

Description automatically generated

**Sequence Diagram for Bed Allocation**

Diagram

Description automatically generated

**Sequence Diagram for Managing Follow-Ups**

**Diagram

Description automatically generated**

**Sequence Diagram for Treatments**

Diagram

Description automatically generated

**Sequence Diagram for Referrals**

Diagram

Description automatically generated

**Sequence Diagram for Updating Referring Doctor Information Database**

**Diagram

Description automatically generated**

**6.2.7 Activity Diagrams for HBMS**

**Activity Diagram for HBMS Access**

**Diagram

Description automatically generated**

**Activity Diagram for Patient Enrollment**

**Diagram

Description automatically generated**

**Activity Diagram for Updating Patient Personal Information**

**Diagram

Description automatically generated**

**Activity Diagram for Updating Bed Allocation Information**

Diagram

Description automatically generated

**Activity Diagram for Patient Medical Report**

**Diagram

Description automatically generated**

**Activity Diagram for Treatments**

Diagram

Description automatically generated

**Activity Diagram for Referral List**

Diagram

Description automatically generated

**Activity Diagram for Updating the Referring Doctor Database**

**Diagram

Description automatically generated**

**6.3 System Database Design for HBMS**

**6.3.1 Entity Relationship Diagram (ERD) for HBMS**

Diagram

Description automatically generated

**6.3.2 Normalization (3NF) Listing of All Dependencies**



**6.3.3 Data Dictionary for HBMS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table** | **Attributes** | **Data Type** | **Primary Key** | **Allow Null** | **Description** |
| **Patient** | Id | Int | Yes | No | Patient's Identification Number |
| FirstName | varchar(50) | No | No | Patient's First Name |
| LastName | varchar(50) | No | No | Patient's Last Name |
| Address1 | varchar(50) | No | No | Patient's Primary Address |
| Address2 | varchar(50) | No | Yes | Patient's Secondary Address |
| City | varchar(50) | No | No | Patient's City |
| State | varchar(50) | No | No | Patient's State |
| Zip | varchar(50) | No | No | Patient's Zip Code |
| MaleFemale | varchar(10) | No | No | Patient's Gender |
| BirthDate | Date | No | No | Patient's Date of Birth |
| HomePhone | varchar(20) | No | No | Patient's Home Phone Number |
| WorkPhone | varchar(20) | No | Yes | Patient's Work Phone Number |
| SSN | Int | No | No | Patient's Social Security Number |
| ReferringDoctor | varchar(50) | No | No | Patient's Referring Doctor |
| HeightFeet | varchar(50) | No | No | Patient's Height in Feet |
| HeightInches | varchar(50) | No | No | Patient's Height in Inches |
| Weight | varchar(50) | No | No | Patient's Weight |
| DoctorsNotes | varchar(180) | No | Yes | Notes on Patient Written by Doctor |
|  |  |  |  |  |  |
| **Doctor** | Id | Int | Yes | No | Doctor's Identification Number |
| FirstName | varchar(50) | No | No | Doctor's First Name |
| LastName | varchar(50) | No | No | Doctor's Last Name |
| Address1 | varchar(50) | No | No | Doctor's Primary Address |
| Address2 | varchar(50) | No | Yes | Doctor's Secondary Address |
| City | varchar(50) | No | No | Doctor's City |
| State | varchar(50) | No | No | Doctor's State |
| Zip | varchar(10) | No | No | Doctor's Zip |
| Phone | varchar(20) | No | No | Doctor's Phone Number |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |  | |  | |
| **Referral** | Id | | Int | | Yes | | No | | Referral Identification Number | |
| FirstName | | varchar(50) | | No | | No | | Referring Doctor's First Name | |
| LastName | | varchar(50) | | No | | No | | Referring Doctor's Last Name | |
| Phone | | varchar(20) | | No | | No | | Referring Doctor's Phone Number | |
| Email | | varchar(50) | | No | | No | | Referring Doctor's Email Address | |
| DoctorId | | Int | | No | | No | | Referring Doctor's Identification Number | |
| DateCreated | | Date | | No | | No | | Date of Referral | |
|  |  | |  | |  | |  | |  | |
| **PatientBed** | bedId | | Int | | Yes | | No | | bed's Identification Number | |
| wardNumber | | Int | | No | | No | | Ward Number | |
| patientId | | Int | | No | | No | | Patient Identification Number | |
| bedCost | | Int | | No | | No | | The cost of the Bed. | |
|  |  | |  | |  | |  | |  | |
| **PatientProgram** | PatientId | | Int | | Yes | | No | | Patient's Identification Number | |
| ProgramId | | Int | | No | | No | | Patient's Program Identification Number | |
| Completed | | Date | | No | | No | | Date Program been Completed | |
| IntervalId | | Int | | No | | No | | Interval Identification Number | |
| IsActive | | Boolean | | No | | No | | Is Patient in Program or not | |
| DoctorNote | | varchar(180) | | No | | Yes | | Notes Written by Doctor | |
| IsSuccessfull | | Boolean | | No | | No | | Did Patient Complete Program Successfully | |
|  |  | |  | |  | |  | |  | |
| **Interval** | Id | | Int | | Yes | | No | | Interval Identification Number | |
| Name | | varchar(50) | | No | | No | | 1 week, 1 month, 3 month, 6 month | |
|  |  | |  | |  | |  | |  | |
| **MedicalCondition** | Id | | Int | | Yes | | No | | Identification Number given to Medical Condition | |
| Name | | varchar(50) | | No | | No | | Name of Medical Condition | |
| TypeId | | Int | | No | | No | | Medical Condition Type Identification Number | |
|  |  | |  | |  | |  | |  | |
| **FollowUp** | PatientId | | Int | | Yes | | No | | Patient's Identification Number | |
| FollowUpDate | | Date | | No | | No | | Patient's Follow Up Date | |
| Id | | Int | | No | | No | | FollowUp Identification Number | |
|  | |  | |  | |  | |  | |  | |
| **Treatment** | | Id | | Int | | Yes | | No | | Treatment Identification Number | |
| Name | | varchar(50) | | No | | No | | Name of Treatment | |
| DoctorNote | | varchar(180) | | No | | Yes | | Notes Written by Doctor | |
| ProgramId | | Int | | No | | No | | Treatment Program Identification Number | |
| SideEffects | | varchar(180) | | No | | No | | List of Side Effects for Treatment | |
|  | |  | |  | |  | |  | |  | |

**6.4 System Interface Design**

|  |  |
| --- | --- |
| **Model Name** | **HBMS System Login** |
| **Parameters Passed & Meaning** | Email: The email is essentially the username which is user specific. Password: Password is the password created by user and associated with username. |
| **Description of Module Function** | The module is used to gain access to the HBMS system. User will input their username and password. |
| **Input** | Click: Login |
| **Output** | Access to PEST system if credentials are valid. Deny access if invalid credentials. |
| **Called Modules** | Dashboard. |
| **Story** | Once the user submits credentials, they will either be granted or denied to the HBMS system based on the credentials provided. |
| **Error Message** | If invalid credentials, prompt user of invalid credentials and denied access. |

|  |  |
| --- | --- |
| **Model Name** | **Home – Dashboard** |
| **Parameters Passed & Meaning** | No parameters passed. |
| **Description of Module Function** | The Dashboard module is the home page of the application. From here patients can access the different functions of the system using the tabs listed on the left. This page also serves as a quick reference point for upcoming follow ups and a list of the most recent new referrals. |
| **Input** | Click: “Admin”, “Patient Center”, or “Reports” to explore those functionalities. |
| **Output** | According to what is “clicked”, the tab will expand to display sub tabs. |
| **Called Modules** | Admin, Patient Center, Reports. |
| **Story** | Once access to system is granted, patient can click on dashboard to see the dashboard information to quickly see the upcoming follow ups and latest new referrals. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Patient** |
| **Parameters Passed & Meaning** | No parameters are passed. |
| **Description of Module Function** | Once patient center is clicked, all the options in Patient Center are provided. The function of Patients is to provide a list of existing patients, edit existing patient personal information, and create new patient. |
| **Input** | Click: “Edit” or “Create New”. |
| **Output** | If Edit is clicked, patient information screen for editing personal information will appear. If Create New is click, screen for creating new patient will appear. |
| **Called Modules** | Patient Center – Patients – Edit, Patient Center – Patients – Create New |
| **Story** | List of all patients is displayed. User can click either “Edit” or “Create New”. If user clicks Edit, the Edit screen of the patient will appear. If user clicks Create New, a blank form will appear for enrolling new patients. |
|  |  |
| **Model Name** | **Patient Center – Patient - Edit** |
| **Parameters Passed & Meaning** | FirstName, LastName, Address1, Address2, City, State, Zip, MaleFemale, BirthDate, HomePhone, WorkPhone, SSN, ReferringDoctor: All self-explanatory. |
| **Description of Module Function** | This module allows user to edit all the personal information of the patient. |
| **Input** | Click: “Submit” or “Cancel”. |
| **Output** | Return screen to Patient Center – Patient. |
| **Called Modules** | Patient Center – Patient. |
| **Story** | A patient informs the provider that they need their personal information updated. The provider will choose the patient’s name from module Patient Center – Patient and click “Edit”. This screen will appear. Once provider updates the information on this screen, the provider will click “Save” the information, which will then be updated on the Patient Account Database. |
| **Error Msg.** | If fields are empty, prompt user of missing field before saving form. |
|  |  |
| **Model Name** | **Patient Center – Patient – Create Patient** |
| **Parameters Passed & Meaning** | FirstName, LastName, Address1, Address2, City, State, Zip, MaleFemale, BirthDate, HomePhone, WorkPhone, SSN, ReferringDoctor: All self-explanatory. |
| **Description of Module Function** | Module allows new patients to be enrolled into system. More specifically, this module allows for recording patient personal information. |
| **Input** | Click: “Save” or “Cancel” |
| **Output** | If Save is “clicked”: Display the patient medical history form. If Cancel is “clicked”: Return screen to Patient Center – Patient. |
| **Called Modules** | Patient Center – Patient – Create New – Medical History, Patient Center – Patient |
| **Story** | When a new patient arrives at the clinic, the provider creates a new patient profile using “Create New” button. A blank form for personal information is displayed. The provider ascertains patient personal information from the patient and inputs data into empty fields. Once complete, the provider clicks Save. After clicking save, the provider is given another blank form for patient medical history. |
| **Model Name** | **Patient Center – Patient – Create New – Medical History** |
| **Parameters Passed & Meaning** | HeightFeed, HeightInches, Weight, DoctorsNotes, MedicalCondition: All this information is considered relevant patient medical information. |
| **Description of Module Function** | Module allows provider to record patient medical history once personal history has been recorded. |
| **Input** | Click: “Save” or “Cancel” |
| **Output** | Return screen to Patient Center – Patient |
| **Called Modules** | Patient Center – Patient |
| **Story** | Once provider has entered patient personal information, the provider is prompted with this screen to record patient medical history. Provider can inquire the patient and refer to any patient medical document to ascertain medical information. When provider clicks “Save”, a new patient profile is created and added to the system. In addition, patient is listed on Programs. |
| **Error Msg.** | If required fields on form are incomplete, notify user of incompleteness before allowing form submission. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Patient – Programs** |
| **Parameters Passed & Meaning** | PatientId, ProgramId, Completed, IntervalId, IsActive, DoctoreNote, IsSuccessful: Patient ID is used to get patient information, program ID refers to which of the program is selected, Completed simply means is the patient complete with the treatment program, InvervalId is the time interval, DoctorNote is used if the doctor has any comments, lastly the IsSuccessful is used to determine if the patient was successful in the treatment program. |
| **Description of Module Function** | This module is used to enroll patients into treatment programs, determine if the program is successful, and take in doctor’s comments via Notes textbox. |
| **Input** | Click: “Save” or “Cancel”. |
| **Output** | Display Patient Center – Patient screen. |
| **Called Modules** | Patient Center – Patient |
| **Story** | Once patient is enrolled into the system, they need to be enrolled into programs. By default they are enrolled into the smoking program by default. In addition to smoking, they can be enrolled into Diabetes, Obesity, or Cholesterol programs by checking the box. Lastly, at the conclusion of the program, the Provider can make conclusion on the patient’s overall health and progress. |
| **Model Name** | **Patient Center – Programs** |
| **Parameters Passed & Meaning** | No parameters are passed. |
| **Description of Module Function** | Module allows user to view all patients currently active in programs. In addition, this module allows the user to open up “Edit Patient Progress” which is then used to report the medical status of a patient. |
| **Input** | Click: “Edit Patient Progress”. |
| **Output** | Display screen to edit patient progress. |
| **Called Modules** | Patient Center – Patient – Create New – Medical History, Patient Center – Patient |
| **Story** | When a new patient arrives at the clinic, the provider creates a new patient profile using “Create New” button. A blank form for personal information is displayed. The provider ascertains patient personal information from the patient and inputs data into empty fields. Once complete, the provider clicks Save. After clicking save, the provider is given another blank form for patient medical history. |
| **Model Name** | **Patient Center – Programs – Edit Patient Progress** |
|  |  |
| **Parameters Passed & Meaning** | No parameters are passed. |
| **Description of Module Function** | This module allows the user to update patient progress at the time intervals: Initial, 1 Week, 1 Month, 3 Month, and 6 Month. |
| **Input** | Click: “Initial”, “1 Week”, “1 Month”, “3 Month”, “6 Month” |
| **Output** | Display the time interval screen that is clicked. |
| **Called Modules** | Patient Center – Programs – Program Progress |
| **Story** | Once the patient is enrolled and their treatment program begins. At each of the stated time intervals, the patient will visit the clinic. The provider will click the appropriate time interval, and the module Patient Center – Programs –Chart Medical Metrics will appear. The provider will move onto that screen and input the required fields. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Programs – Chart Medical Metrics** |
| **Parameters Passed & Meaning** | PatientId, IntervalId, ProgramMetricId, MetricValue: Each medical metric is associated with a patient, time interval, and program. There are four programs, each with multiple medical metrics. Any patient can come in at the stated interval, and their medical metrics are recorded for that time interval and what programs they are enrolled in. |
| **Description of Module Function** | This module’s primary function is to record medical metrics of a patient at each time interval. The look of the module does not change from one time interval to another. However, the PatientId, IntervalId, and MetricValues will change. The ProgramMetricId will remain same but associated MetricValue will change from interval to interval. |
| **Input** | Click: “Save” or “Cancel” |
| **Output** | Display the Patient Center – Programs screen. |
| **Called Modules** | Patient Center – Programs |
| **Story** | The treatment programs have time intervals. At each time interval, the patient comes to the clinic, at which point their medical metrics are charted recorded. This is essentially the way to check the progress and health of the patient. Note: The only programs and medical metrics that are displayed, are the ones the patient is enrolled in. In this case, the patient is enrolled in all four programs, thus all four programs and their medical metrics are displayed. However, if the patient was only enrolled in Obesity program, then only the smoking and obesity programs would be displayed. Remember all patients are enrolled in the smoking program by default, hence two programs would be displayed. |
| **Error Msg.** | If the form is incomplete, notify the user of incompleteness and system does not process form. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Follow-Ups** |
| **Parameters Passed & Meaning** | No parameters are passed. |
| **Description of Module Function** | Module function is to list all follow-ups and display screen for creating new follow-up. |
| **Input** | Click: “Edit” or “Create New” |
| **Output** | According to what is clicked, display the appropriate screen. |
| **Called Modules** | Patient Center – Follow Ups – Edit Follow-Ups, Patient Center – Follow Ups – New Follow-Up. |
| **Story** | This screen shows the full list of follow-ups. Whenever a patient is due for a follow-up, the doctor creates a new follow-up using “Create New” button. This will bring the provider to the Create New Follow-Up screen. If a patient cannot make a follow-up, then the provider can edit the follow-up to assign a new date. The Provider can also view details in of any follow-up. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Follow-Ups – Edit Follow-Up** |
| **Parameters Passed & Meaning** | PatientId, FollowUpDate, FollowUpID: Each follow has the patient ID so the system can identify to whom the follow-up is pertaining to. In addition, the follow-up date is used to indicate the date the patient will come in. |
| **Description of Module Function** | Module function is to edit existing follow-ups that are scheduled. |
| **Input** | Click: “Save” or “Cancel” |
| **Output** | Display Patient Center – Follow-Ups screen. |
| **Called Modules** | Patient Center – Follow Ups |
| **Story** | Whenever a patient is due for a follow-up, the doctor creates a new follow-up using “Create New” button. This will bring the doctor to the Create New Follow-Up screen. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Follow-Ups – Create Follow -Ups** |
| **Parameters Passed & Meaning** | PatientId, FollowUpDate, FollowUpID: Each follow has the patient ID so the system can identify to whom the follow-up is pertaining to. In addition, the follow-up date is used to indicate the date the patient will come in. |
| **Description of Module Function** | The function of this module is to create new follow-ups. |
| **Input** | Click: “Save” or “Cancel” |
| **Output** | Display Patient Center – Follow-Ups screen. |
| **Called Modules** | Patient Center – Follow Ups |
| **Story** | The provider uses this function every time a patient is due for a follow-up. The patient provides the date available and the provider uses that date as the new follow-up date. |
| **Error Msg.** | If a date for follow-up is not selected, prompt user of incompleteness and system must not process follow-up. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Referrals** |
| **Parameters Passed & Meaning** | No parameters are passed. |
| **Description of Module Function** | The function of this module is to track referrals and display screens to edit, create, and view all referrals. |
| **Input** | Click: “Edit” or “Create New” |
| **Output** | Display screen according to what option was clicked. |
| **Called Modules** | Patient Center – Referrals – Edit, Patient Center – Referrals – Edit. |
| **Story** | A referring doctor refers a new patient to the provider via phone, email, or fax. The provider takes down the patient information and uses this function to create a new referral. Once a referral is created, Provider can edit and view the referrals. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Referrals – Edit Referral** |
| **Parameters Passed & Meaning** | PatientId, FollowUpDate, FollowUpID: Each follow has the patient ID so the system can identify to whom the follow-up is pertaining to. In addition, the follow-up date is used to indicate the date the patient will come in. |
| **Description of Module Function** | The function of this module is to edit new follow-ups. |
| **Input** | Click: “Save” or “Cancel” |
| **Output** | Display Patient Center – Follow-Ups screen. |
| **Called Modules** | Patient Center – Follow Ups |
| **Story** | The referring doctor calls the Provider to provide a new referrals. The Provider takes the information about the potential new patient and creates the new referral. Then the provider just has to wait until the referred patient walks through the door. |
| **Error Msg.** | If required input fields are missing, inform the user and system should not process the follow-up. |

|  |  |
| --- | --- |
| **Model Name** | **Patient Center – Referrals – Create New Referral** |
| **Parameters Passed & Meaning** | PatientId, FollowUpDate, FollowUpID: Each follow has the patient ID so the system can identify to whom the follow-up is pertaining to. In addition, the follow-up date is used to indicate the date the patient will come in. |
| **Description of Module Function** | The function of this module is to create new follow-ups. |
| **Input** | Click: “Save” or “Cancel”. |
| **Output** | Display Patient Center – Follow-Ups screen. |
| **Called Modules** | Patient Center – Follow Ups |
| **Story** | The referring doctor calls the Provider to provide a new referrals. The Provider takes the information about the potential new patient and creates the new referral. Then the provider just has to wait until the referred patient walks through the door. |
| **Error Msg.** | If required input fields are missing, inform the user and system should not process the follow-up. |

**Glossary**

**COTS**: Short for **c***ommercial* **o***ff-***t***he-***s***helf*, an adjective that describes [software](http://www.webopedia.com/TERM/S/software.html) or [hardware](http://www.webopedia.com/TERM/H/hardware.html) products that are ready-made and available for sale to the general public. For example, Microsoft Office is a COTS product that is a packaged software solution for businesses. COTS products are designed to be implemented easily into existing systems without the need for customization.

**Follow-Up:** A term used to refer to the process in which the patient is scheduled to revisit the provider.

**Medical Metric**: A unit of measure used to assess patient progress as it related to a specific treatment program.

**Patient**: Any individual who is suffering from a medical condition(s) and as a result, is seeking the help of the provider in order to alleviate the condition(s).

**Patient Progress**: Term used to determine if the patient’s health is improving in positive or negative direction within a treatment program.

**Medical Clinician**: Medical professional who is utilizing the HBMS in order to help a patient alleviate a medical condition(s).

**Medical Provider**: Medical professional who is utilizing the HBMS in order to help a patient alleviate a medical condition(s).

**Referring Doctor**: Medical professional who referrers a patient to a provider because treatment for the patient is outsider the professional’s scope of expertise.

**Report:** A written or automatically generated document which has information pertinent stakeholders of the HBMS

**Tablet**: Device used to access the HBMS

**Time Interval**: A point in time, at which a patient returns to the Provider for further medical treatment.

**Treatment**: The term used to describe medical care given to a patient for a medical condition.

**Treatment Program**: Treatment Program also referred to as simply program, is a set duration of time in which the medical condition(s) experienced by a patient is treated.

A

**Activity Diagrams**, 5, 90

appointments, 8, 22, 23, 77

Java Developer, 14

C

Java Programmers, 14

cholesterol, 8, 9, 10, 11, 15, 17

Cholesterol, 10, 11, 111, 125, 126, 130, 132

Cigarettes, 10

Clinician, 16

**Context Model**, 44

costs, 12, 24

**COTS**, 28, 29, 39, 135, 137

**Current System**, 15, 18, 19, 21

D

Database Specialist, 14, 39

diabetes, 8, 9, 10, 11, 15, 17

Diabetes, 10, 111, 125, 126, 130, 132

diagnosed, 9, 11

E

evaluation, 9, 11, 80

F

**Follow-Up**, 25, 26, 27, 77, 116, 117, 135

H

Hardware, 4, 6, 14, 17, 39, 40

healthcare, 8, 20, 24, 25, 28, 41, 78

I

**Intranet**, 43

L

Laptops, 14, 25, 27

M

**Medical Clinician**, 11, 12, 135

**Medical Metric**, 135

**Medical Provider**, 135

N

**Narratives**, 5, 74

Network Specialist, 14

**Networks**, 5, 43

O

obesity, 8, 9, 10, 11, 15, 17, 115

Obesity, 10, 111, 115, 125, 126, 130, 132

P

**Patient**, 1, 2, 8, 9, 10, 15, 16, 17, 28, 29, 39, 44, 45, 74, 75, 76, 77, 78, 79, 80, 82, 83, 84, 85, 91, 92, 93, 94, 101, 102, 105, 106, 107, 108, 109, 111, 112, 113, 114, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 135

**Hospital Bed Management System**, 1, 2, 8, 9, 28, 29, 39, 44, 45, 74, 75, 76, 77, 78, 79, 80

HBMS, 5, 8, 9, 11, 12, 20, 22, 24, 28, 29, 30, 39, 44, 73, 74, 75, 76, 77, 78, 79, 80, 81, 90, 95, 99, 101, 104, 135

program, 8, 9, 11, 15, 22, 23, 24, 28, 111, 113, 114, 115, 125, 126, 127, 128, 130, 131, 133, 135

progress, 10, 11, 12, 15, 16, 17, 22, 41, 76, 111, 112, 113, 115, 129, 135

R

Referrals, 12, 15, 16, 17, 26, 27, 88, 119, 120, 121

**Referring Doctor**, 79, 80, 89, 98, 101, 102, 122, 123, 124, 135

**Report**, 5, 16, 26, 27, 85, 94, 104, 105, 106, 107, 108, 109, 111, 112, 113, 114, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135

S

scheduled, 9, 11, 15, 16, 17, 25, 77, 117, 135

Security Analyst, 14

**Sequence Diagrams**, 5, 81

smoking, 8, 9, 10, 11, 15, 17, 111, 115, 130

SQL Database Server, 14, 39, 40

SQL Server, 12, 14, 39, 40, 41, 42, 43

System Analyst, 14

System Architect, 14

T

**Tablet**, 135

**Time Interval**, 135

treatment, 8, 9, 11, 15, 22, 23, 25, 28, 111, 113, 115, 125, 126, 127, 128, 135

**Treatment**, 10, 11, 17, 78, 103, 126, 127, 128, 135

U

**Use Case**, 5, 73, 74

V

Eclipse IDE, 14, 39

**Citations**

Beal, V. (n.d.). *COTS - Commercial off-the-shelf*. Retrieved from Webopedia: <http://www.webopedia.com/TERM/C/COTS.html>

Bochmann, Gregor v. "Non-Functional Requirements." 12 10 2010. *Site.Uottawa.ca.* PDF Document. 4 March 2016.

Burr, Alica. APRN Yale New Haven Hospital. Interview. n.d.

CDC. "CDC\_UP\_Non-Functional\_Definition\_Temaplate.doc." 12 Februaury 2006. *CDC.* Document. 4 March 2016.

Rahardjo, Edhy. "Decision Making in Information Technology Acquisition." 2006. *umsl.edu.* Document. 8 March 2016.

Rivera, Ivan. "Synopsis." Fairfield: Ivan@ivanrivera.com, 13 January 2016. Email.

Roberts, Robin. *MedicineNet*. 14 6 2012. medicinenet.com/script/main/art.asp?articlekey=6232. 16 February 2016.