*Cloud-Based Healthcare Management System*

Design Specification Document

*Team Dream*

*American University of Sharjah*

College of Engineering

Computer Science and Engineering

Version #: App-DSD-mmddyyyy

**Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Author(s)** | **Change Description** | **Date** | **Notes** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Table of Content**

[1 Overview 4](#_Toc96588792)

[1.1 Software Context 4](#_Toc96588793)

[1.2 Terminology & Definitions 4](#_Toc96588794)

[2 System Architecture 5](#_Toc96588795)

[2.1 Node Architecture 5](#_Toc96588796)

[2.2 Component Architecture 5](#_Toc96588797)

[2.2.1 Component interface description 5](#_Toc96588798)

[2.2.2 Web service interface description 5](#_Toc96588799)

[2.3 Data Architecture 5](#_Toc96588800)

[3 Conclusions & Future Extensions 6](#_Toc96588801)

[4 Related Material 6](#_Toc96588802)

# Overview

The Cloud-Based Healthcare Management System (CBHMS) stands as a transformative solution tailored explicitly for the modern challenges and needs of the healthcare domain. In today's digital age, there is an ever-growing demand for cohesive, secure, and efficient healthcare platforms. Addressing this demand, CBHMS offers a comprehensive, centralized platform that amalgamates a diverse range of patient-related data and medical resources.

## Software Context

Positioning within the broader business and product line context, CBHMS is an evolution beyond traditional healthcare systems, which often present a fragmented or narrow scope. By contrast, CBHMS ensures a more holistic approach, centralizing everything from detailed patient demographics to intricate medical histories and diagnostic images. This design is not just about consolidation; it is about ensuring that healthcare providers can offer the best care by having immediate access to all necessary data.

Through well-crafted APIs, CBHMS can integrate with pre-existing Hospital Information Systems (HIS), ensuring that institutions can retain their legacy systems while benefiting from modern advancements. Furthermore, the platform is adept at integrating with a range of diagnostic tools, facilitating the automatic upload of radiology images, lab results, and other diagnostic data. This integration extends to a real-time notification system, ensuring that any significant changes or updates to patient data are immediately communicated to the relevant stakeholders.

At its core, CBHMS is a problem solver. It addresses the healthcare sector's intrinsic challenges by offering a unified repository for comprehensive patient data, ensuring that healthcare professionals have both ease of access and ease of management. The platform's emphasis on robust security protocols, including multi-factor authentication and role-based access controls, underscores its commitment to preserving data integrity and security. Additionally, real-time data management capabilities mean stakeholders can effortlessly add, modify, and retrieve patient data, fostering a culture of efficiency and precision. Lastly, the system's resource management capabilities empower administrators to deftly manage hospital listings, doctor specialties, schedules, and facility availability, ensuring that patients always have a transparent and informed choice.

However, no system is without its challenges. In the case of CBHMS, there are strategic issues to navigate. The sensitivity surrounding healthcare data means that data privacy and security are paramount. The system's design inherently focuses on these aspects, but continuous vigilance is necessary. Scalability is another concern; as healthcare institutions expand and evolve, CBHMS must scale seamlessly to cater to the growing user base and data volume. Lastly, the realm of healthcare is riddled with regulations. Different regions have their own standards and compliance benchmarks. Thus, CBHMS is not just about technical prowess but also about adaptability, ensuring it aligns with various regulatory standards across the board.

## Terminology & Definitions

**CBHMS (Cloud-Based Healthcare Management System):** A digital platform devised to centralize and streamline healthcare data management, including patient records, medical resources, and real-time data updates.

**Patient:** An individual who seeks or is currently receiving medical care or treatment. This individual's data, ranging from basic demographics to detailed medical histories, is stored within CBHMS.

**Doctor:** A licensed medical professional responsible for diagnosing and treating patients. Within CBHMS, doctors have profiles detailing their specialties, schedules, and affiliations with healthcare institutions.

**Administrator:** An individual or entity overseeing the CBHMS platform's functioning, ensuring data accuracy, adding, or updating hospital information, and managing user roles and permissions.

**Multi-factor Authentication:** A security process wherein users must verify their identity using multiple methods or factors before gaining access to the system. This often includes something they know like a password, something they have like a phone or a token, and something they are (like a fingerprint).

**Role-based Access Control (RBAC):** A method of restricting system access to authorized users based on their role (e.g., Patient, Doctor, Administrator). Different roles have varying levels of access to the system's data and functionalities.

**Hospital Information System (HIS):** A comprehensive software system designed to manage a hospital's administrative, clinical, and financial tasks.

**API (Application Programming Interface):** A set of tools and protocols that allows different software applications to communicate with each other. In the context of CBHMS, APIs facilitate integration with existing Hospital Information Systems and diagnostic tools.

**Radiology Images:** Digital visuals of the body's interior acquired through diagnostic equipment such as X-rays, MRIs, and CT scanners.

**Lab Results:** Documented outcomes from medical tests conducted in a laboratory setting. This can include blood tests, biopsies, and other diagnostic examinations.

**Notifications System:** A digital mechanism within CBHMS that informs or alerts users like doctors or patients about significant changes or updates to patient data.

# System Architecture

## Node Architecture

**Client Nodes:** These nodes represent the end-users' devices, be it computers, tablets, or smartphones. Through client nodes, users such as patients, doctors, and administrators interact with the system. They send requests to the server node and receive responses, ensuring real-time access to healthcare data. It will also support multi-factor authentication, and role-based access control.

**Diagnostic Tool Nodes:** Given the system's capability to integrate with diagnostic tools, these nodes represent various diagnostic equipment. They communicate directly with the server node, automatically uploading radiology images and lab results to the database.

**Server Node:** At the heart of CBHMS lies the server node, which hosts the core application logic. It processes requests, manages data operations, and ensures smooth interactions between various system components. To ensure robustness and cater to a growing number of users, the system uses a cluster of servers, providing load balancing and fault tolerance.

**Database Node:** This node is responsible for storing all patient-related data, medical histories, radiology images, and other relevant information. Given the sensitivity and volume of the data, the database node is optimized for fast read and write operations. Regular backups ensure data integrity and recovery in case of failures.

A diagram of a server

Description automatically generated

## Component Services Architecture

The Component Services Architecture of CBHMS represents the logical partitioning of the system into distinct components or modules. These components encapsulate specific functionalities and interact with each other to offer the comprehensive features of CBHMS. Dividing the system into components enhances modularity, scalability, and maintainability.

**User Interface (UI):** This component manages all communications between the CBHMS and external systems, such as hospital Information Systems and real time data updates. It goes through User Authentication Component to allow access and management of the external system.

**User Authentication & Authorization Component (UAAC):** This component is dedicated to ensuring that only authorized users can access the system. It manages the multi-factor authentication process and ensures that users have access only to the data and functionalities relevant to their roles, be it Patient, Doctor, or Administrator. It interacts with the Patient Records Management Component to fetch and display data based on user roles and controls access to the Medical Resource Management Component and the Real-time Data Update Component.

**UI & UAAC Mapping to FR:**

FR 1.1: Supports a multi-factor authentication process, ensuring Patients, Doctors, and Administrators undergo a secure login procedure.

FR 1.2: Provides role-based access controls. This ensures that users can access only specific data and functionalities relevant to their roles, be it Patient, Doctor, or Administrator. It controls access to the Patient Records Management Component, Medical Resource Management Component, Real-time Data Update Component, and potentially others based on the user's role.

**Communication Interface Component (CIC):** Residing on the Diagnostic Tools node, this component facilitates data exchange between CBHMS and external diagnostic tools such as radiology imaging tools and lab results. It ensures smooth communication and data transfer to and from external systems.

**Patient Records Management Component (PRMC):** As the name suggests, this component deals with all functionalities related to patient data. It allows for the storage, retrieval, and update of comprehensive patient information, ranging from demographics and medical histories to radiology images and lab results. Additionally, it provides search functionalities based on Patient ID. This component receives data updates from the Real-time Data Update Component and provides data for analysis to the Analytics & Reporting Component.

**PRMC & CIC mapping to FR:**

FR 2.1: Allows for the storage, retrieval, and update of comprehensive patient information, including demographics, medical histories, medications, allergies, radiology images, and lab results.

FR 2.2: Offers a feature to search for patient records based on Patient ID.

FR 2.3: Displays a comprehensive patient profile, showcasing demographics, medical histories, medications, allergies, radiology images, and lab results. It receives updates from the Real-time Data Update Component and provides data for analysis to the Analytics & Reporting Component.

**Patient Management Component (PMC):** This component ensures that the system is always up to date. It manages the real-time entry and retrieval of patient data and sends out notifications or alerts to relevant users when significant changes or updates are made to patient data. It interacts with the Patient Records Management Component for data synchronization and can communicate with external systems through the Communication Interface. It also interacts with analytics and reports component.

**RDUC mapping to FR:**

FR 3.1: Enables users to enter and retrieve patient data in real-time, ensuring the system always remains current.

FR 3.2: Sends push notifications or alerts to relevant users when there are significant changes or updates to patient data. It interacts with the Patient Records Management Component for data synchronization and can communicate with external systems through the Communication Interface.

**Medical Resource Management Component (MRMC):** This component manages resources related to healthcare facilities. Administrators use it to add hospitals to the system, update details about doctors, their specialties, schedules, and the availability of medical facilities. It also allows doctors to update their schedules and patients to check for available medical resources. This component may fetch or update data from external systems through the Communication Interface. This component also manages all communications between the CBHMS and external systems, such as Hospital Information Systems or diagnostic tools. It includes the API endpoints and any other necessary interfaces for data exchange. It facilitates data exchange between CBHMS and external systems, and the cloud service.

**Hospital Records Management Component (HRMC):** This new component, situated on the Database node, is responsible for the storage, management, and retrieval of all hospital-related records such as doctor records, hospital information, and facility availability. It ensures that all details pertaining to hospitals, such as their facilities, services, doctors, and departments, are stored and updated efficiently.

**MRMC & HRMC mapping to FR:**

FR 4.1: Allows administrators to add hospitals to the system, update details about doctors, their specialties, schedules, and the availability of medical facilities.

FR 4.2: Enables doctors to update their schedules.

FR 4.3: Allows patients to check different hospitals for the availability of doctors and medical facilities. It also manages communications between the CBHMS and external systems like Hospital Information Systems or diagnostic tools. This component facilitates data exchange between CBHMS and external systems, and the cloud service.

**Analytics & Reporting Component (ARC):** While not explicitly mentioned in the earlier requirements, an analytics component would be crucial for deriving insights from the stored data. This component can process data to extract correlations between demographics, medical histories, lab results, and medications. It aids in making data-driven decisions and predictions and may derive insights related to resource usage from the Medical Resource Management Component. It accesses the patient data through Patient Management Component (PMC).

**ARC mapping to FR:**

FR 5.1: Furnishes an analytics dashboard that extracts correlations between various patient data, aiding in making data-driven decisions and predictions. It can derive insights related to resource usage from the Medical Resource Management Component and offer correlations between demographics, medical histories, lab results, and medications.

A diagram of a company

Description automatically generated

### Component interface description

**User Authentication & Authorization Component (UAAC) Interface:**

**AuthenticateUser Interface**

**Parameters:** username, password, token (optional)

**Return Type:** AuthenticationStatus (Success, Failure, RequiresToken)

**Description:** Authenticates users based on credentials and optional multi-factor token.

**AuthorizeUser Interface**

**Parameters:** userID, requestedFunction

**Return Type:** AuthorizationStatus (Authorized, Unauthorized)

**Description:** Verifies user permission for a given function based on their role.

**Patient Management Component (PMC) Interface:**

**StorePatientData Interface**

**Parameters:** patientID, patientData

**Return Type:** StorageStatus (Success, Failure)

**Description:** Stores comprehensive patient data.

**SearchPatientData Interface**

**Parameters:** patientID

**Return Type:** PatientData or Error

**Description:** Searches for patient records using Patient ID.

**RetrievePatientProfile Interface**

**Parameters:** patientID

**Return Type:** PatientProfileData or Error

**Description:** Retrieves a comprehensive patient profile.

**Patient Managment Component (PMC) Interface:**

**EnterPatientDataRealTime Interface**

**Parameters:** patientID, newData

**Return Type:** UpdateStatus (Updated, Error)

**Description:** Updates patient data in real-time.

**RetrievePatientDataRealTime Interface**

**Parameters:** patientID, dataFields (optional)

**Return Type:** PatientData or Error

**Description:** Retrieves real-time patient data.

**SendNotifications Interface**

**Parameters:** userID, alertCriteria

**Return Type:** NotificationStatus (Sent, Error)

**Description:** Sends notifications based on specific data criteria.

**Medical Resource Management Component (MRMC) Interface:**

**AddHospital Interface**

**Parameters:** hospitalDetails

**Return Type:** AdditionStatus (Added, Error)

**Description:** Adds a new hospital to the system.

**UpdateDoctorDetails Interface**

**Parameters:** doctorID, newDetails

**Return Type:** UpdateStatus (Updated, Error)

**Description:** Updates doctor details, such as their specialties or schedules.

**CheckHospitalAvailability Interface**

**Parameters:** hospitalID

**Return Type:** HospitalData or Error

**Description:** Checks doctor and facility availability within a hospital.

**Analytics & Reporting Component (ARC) Interface:**

**DashboardData Interface**

**Parameters:** None

**Return Type:** DashboardData or Error

**Description:** Fetches analytics and correlations for dashboard display.

**Appointment Scheduling & Management Component (ASMC) Interface:**

**BookAppointment Interface**

**Parameters:** patientID, doctorID, timeSlot

**Return Type:** BookingStatus (Booked, Error)

**Description:** Schedules an appointment.

**RescheduleAppointment Interface**

**Parameters:** appointmentID, newTimeSlot

**Return Type:** RescheduleStatus (Rescheduled, Error)

**Description:** Reschedules an existing appointment.

**CancelAppointment Interface**

**Parameters:** appointmentID

**Return Type:** CancellationStatus (Cancelled, Error)

**Description:** Cancels an existing appointment.

### Web service interface description

User Authentication & Authorization Microservice:

Endpoint: /accounts/login

HTTP Method: POST

Request Body: { "username": "string", "password": "string", "token": "string (optional)" }

Response: { "status": "Success/Failure/RequiresToken", "userID": "string (if authenticated)" }

Description: Authenticates a user.

Endpoint: /accounts/authorize/{UserID}

HTTP Method: GET

Request Parameters: userID, requestedFunction

Response: { "status": "Authorized/Unauthorized" }

Description: Authorizes a user for a specific function.

Patient Records Management Microservice:

Endpoint: /patients/record

HTTP Method: POST

Request Body: { "patientID": "string", "patientData": "object" }

Response: { "status": "Success/Failure" }

Description: Stores patient data.

Endpoint: /patients/record/{patientID}

HTTP Method: GET

Response: PatientData object or Error

Description: Searches for a patient record.

Endpoint: /patients/{patientID}

HTTP Method: GET

Response: PatientProfileData object or Error

Description: Retrieves a patient's profile.

Real-time Data Update Microservice:

Endpoint: /patients/{patientID}

HTTP Method: PUT

Request Body: { "newData": "object" }

Response: { "status": "Updated/Error" }

Description: Updates patient data in real-time.

Endpoint: /patients

HTTP Method: GET

Request Parameters: dataFields (optional)

Response: PatientData object or Error

Description: Retrieves real-time patient data.

Endpoint: /notifications/{userID}

HTTP Method: POST

Request Body: { "alertCriteria": "object" }

Response: { "status": "Sent/Error" }

Description: Sends alerts or notifications.

Medical Resource Management Microservice:

Endpoint: /hospitals

HTTP Method: POST

Request Body: { "hospitalDetails": "object" }

Response: { "status": "Added/Error" }

Description: Adds a new hospital.

Endpoint: /doctors/{doctorID}

HTTP Method: PUT

Request Body: { "newDetails": "object" }

Response: { "status": "Updated/Error" }

Description: Updates doctor details.

Endpoint: /hospitals

HTTP Method: GET

Response: HospitalData object or Error

Description: Checks availability in a hospital.

Analytics & Reporting Microservice:

Endpoint: dashboard/analytics/

HTTP Method: GET

Response: DashboardData object or Error

Description: Fetches analytics for dashboard display.

Appointment Scheduling & Management Microservice:

Endpoint: /appointments

HTTP Method: POST

Request Body: { "patientID": "string", "doctorID": "string", "timeSlot": "datetime" }

Response: { "status": "Booked/Error" }

Description: Schedules an appointment.

Endpoint: /appointments/{appointmentID}

HTTP Method: PUT

Request Body: { "newTimeSlot": "datetime" }

Response: { "status": "Rescheduled/Error" }

Description: Reschedules an appointment.

Endpoint: /appointments/{appointmentID}

HTTP Method: DELETE

Response: { "status": "Cancelled/Error" }

Description: Cancels an appointment.

## Data Architecture

**User Authentication & Authorization Component**

**Table:** Users

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| UserID | Integer | Unique identifier for the user |
| Username | String | User's login name |
| PasswordHash | String | Hashed password for security |
| Role | String | Role (e.g., Patient, Doctor, Admin) |
| TwoFactorEnabled | Bool | Indicates if MFA is enabled |
| Email | String | User's email address |

**Patient Records Management Component**

**Table:** Patients

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| PatientID | Integer | Unique identifier for the patient |
| FirstName | String | Patient's first name |
| LastName | String | Patient's last name |
| DOB | Date | Date of birth |
| Gender | String | Gender |
| MedicalHistory | MedicalHistory[] | Medical history notes |
|  |  |  |
| Allergies | String[] | Known allergies |
| RadiologyImages | RadiologyImages[] | Radiology images (stored as binary data) |
| LabResults | LabResults[] | Laboratory results |

**Table:** MedicalHistory

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| MedicalHistory\_ID | Integer | Unique identifier for the History |
| Patient\_ID | Integer | Unique identifier for the Patient |
| Medications | Medications[] | Current medications |
| HistoryDate | Date | Date of history |

**Table:** Medications

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Medications \_ID | Integer | Unique identifier for the patient |
| Medications \_Name | String | Medication name |
| Medications\_Dose | String | Medication Dose |

**Table:** LabResults

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| LabResult\_ID | Integer | Unique identifier for the result |
| LabResult\_Type | String | Type of result |
| LabResult\_Date | Date | Date of the result |
| LabResultData | Blob | Data of the result |

**Table:** RadiologyImages

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| Image\_ID | Integer | Unique identifier for the Image |
| Device\_Capture\_ID | String | Machine Capturing ID |
| Image\_Date | Date | Data of Capture |
| Patient\_ID | String | Patient ID of whom it was captured |
| ImageData | Blob | Image data |

**Medical Resource Management Component**

**Table:** Hospitals

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| HospitalID | Integer | Unique identifier for the hospital |
| Name | String | Name of the hospital |
| Capacity | Integer | Total capacity of the hospital |
| Specialties | String | Available specialties |
| Location | String | Address of the hospital |

**Table:** Doctors

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| DoctorID | Integer | Unique identifier for the doctor |
| HospitalID | Integer | Identifier of the hospital |
| DName | String | Doctor's name |
| Specialty | String | Medical specialty |
| Schedule | Schedule | Doctor's availability and schedule |

**Table:** Schedule

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| DoctorID | Integer | Unique identifier for the doctor |
| TimeSlots | Slot [] | Identifier of the hospital |

**Table:** Slot

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| DoctorID | Integer | Unique identifier for the doctor |
| Patient\_ID | Integer/None | Unique identifier for the patient |
| Date | Date | Date of appointment |
| Time | Time | Time of appointment |
| Reason | String | Reason of appointment |

**Analytics & Reporting Component**

**Table:** Analytics Reports

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ReportID | Integer | Unique identifier for the report |
| ReportType | Integer | Type of report (e.g., usage, trends) |
| GeneratedOn | String | Date and time report was generated |
| ReportData | String | The actual report content |

# Conclusions & Future Extensions

In conclusion, this document has provided a comprehensive overview of the proposed Cloud-Based Healthcare Management System (CBHMS), detailing its system architecture, component services, and web service interfaces. The system is designed to meet key functional and non-functional requirements including secure and efficient management of healthcare data, robust user authentication, real-time data updates, and effective medical resource management.

The architectures and interfaces outlined emphasize scalability, security, and usability, addressing the critical needs of various stakeholders including patients, healthcare providers, and administrators. The system's modular design ensures easy maintainability and future scalability.

As healthcare technology evolves, the CBHMS can be extended to incorporate emerging technologies such as AI-driven diagnostics, telemedicine, and more integrated patient engagement tools. Regular updates and adaptations to this system will ensure it remains at the forefront of healthcare management technology.

The document serves as a foundational blueprint for the development, implementation, and future enhancement of the CBHMS. It aims to streamline healthcare operations, improve patient outcomes, and optimize resource utilization, ultimately contributing to more efficient and patient-centric healthcare services.

# Related Material

[1] requirement specification version filename.doc