Topic: Practice Management System

Domain – HealthCare

Problem Statement

Today most of the large hospitals have hospital management system but still lot of local practitioners lacks a sophisticated system for patient data management.

System should provide Physicians, Nurses and Patient a platform where they can capture and view the data.

The system should allow Nurses to capture the patient data using a tablet or mobile phone, in case of low internet connectivity it should allow to capture offline data and provide ability to Sync once you are back in network.

System should also have ability to schedule appointments and manage calendar.

A physician should be able to check his upcoming patient visits and able to re-schedule the same based on need. Physician should also have an ability to check patient history from his/her previous visits.

The system should also provide an interface to Patient to check his/her details and download the same in a CSV format. The system should also allow patient to request for appointment.

Patient can also propose any changes to his/her captured data and the data should go to physician for approval, before it reflects in the system.

The application should also allow Patient nominees to access patients details if formal acknowledgement is signed.

Mandatory Modules

Inbox – All the tasks for a Dr or Nurse should appear in inbox, this is dashboard page or landing page for them. User should be able to traverse through this page to various modules like Scheduler, Visits, Notes etc.

Patient Visit Module – Capture visit data in Online and Offline mode. For every visit, the patient vitals like height, weight, blood pressure will be recorded. Any new allergies would be added. Also, his/her medications and diagnoses for a visit are recorded. If any procedures are performed, then those are also recorded.

Appointment Schedule Module – Allow Nurses and Dr to schedule appointment, show the calendar similar to the one we see in outlook. Patient should be able to schedule data collection appointment

Patient Portal module – Patient to check his details and also download the same.

Admin – Module for maintaining master data

User – Login, Reset Password, Register

Practice Management System

Project Abstract

Practice management System is a web application which provides multi device user interface and easy and efficient way to manage patient visits for local practitioner.

The application is currently catering to a single practice but should be extensible to cater to different practices.

The application allows day to day activities of practice through scheduling appointment, capturing data and allowing patient to access his or her data

Project Flow

The patient log-in / register in the system. Once register, patient adds basic demographic details. Patient schedules an appointment using scheduler module.

The scheduled appointment appears in doctor's inbox and he/she can action on the same. A notification is also sent to Nurse based on accepted appointments.

On the day of visit, nurse captures patient's vitals like blood pressure, sugar levels, weight, height. Also, notes the initial diagnosis.

For the first visit, Nurse also validates the data entered by patient while registration.

During doctor's consultation, doctor capture details like, diagnosis, any procedure suggested like blood test, scans and also adds medications

When patient comes back, he can export the data for his visit and carry it with him for further procedures. Patient can also check details of his current as well as previous visit.

Description of modules

User Module

User Registration

There would be three types of registration, physician, nurse and patient. Physician and nurse registration will happen through admin module and the patient registration will happen through register user screen.

For Physician and Nurse below details are captured by admin, on click of save an email will be sent on the provided email with a default password

- Title
- First Name
- Last Name
- Email Id
- DOB

- Role
    - Physician
    - Nurse
    - Admin
- Employee ID

For Patient registration below details are captured

- Title
- First Name
- Last Name
- Email Id
- Date of birth
- Contact Number
- Password
- Confirm Password

Login Screen

This is the default screen when application loads, post registration user can log-in. The log-in screen shows options for registration and forgot / change password

For first time login (For Nurses, Admin and Physician only) they should be automatically redirected to change password page.

Post 3 unsuccessful attempts the account will be locked and can be unlocked only by admin.

Change password screen

- Old Password
- New Password
- Confirm New Password

Forgot Password

- Send default password on registered email and then on login re-direct to change password

Patient Module

Patient Details Screen

This screen will capture all the patient specific data. Once a patient registers, he/she has to enter this data, below are the sections it will cater to

- Demographics –
    - First Name
    - Last Name
    - DOB
    - Age – Auto Populate based on DOB
    - Gender –
    - Race
    - Ethnicity

    - Languages Known
    - Email
    - Home Address
    - Contact Number
    - Emergency Contact Info
        - First Name
        - Last Name
        - Relationship
        - Email Address
        - Contact
        - Address – allow to select same as above
        - Do you need access to patient portal?
- Allergies
    - Type of allergies
    - Is allergy fatal

Patient Visit Details

This screen captures details when a patient visits the physician. This screen should have ability to save data offline. These details are captured during every visit.

- Patient Details - Should auto-populate all the fields from patient details screen. The content should be read only and Dr or nurse can edit if he finds things are not matching
- Visit Details
    - Vital Signs
        - Height
        - Weight
        - Blood Pressure (systolic / diastolic)
        - Body Temperature
        - Respiration Rate
- Diagnosis – Dr can enter multiple diagnosis
    - Diagnosis
    - Description
- Procedures– Dr can enter multiple Procedures
    - Procedures
    - Description
- Medication - Dr can enter multiple medications
    - Medication
    - Dosage
    - Description

Inbox Module

When a doctor or nurse logs-in they see the inbox module, the module should show below sections

- Upcoming appointment – Show only for a week's appointment/ Color code based on todays appointment vs rest of the appointment
- Any notes from nurse or other physicians – The Dr. can reply or close the message

- Patient's Inbox – Show upcoming appointment, show if some appointment is declined by Dr

Scheduling Module

Add / Edit/ Delete Meeting

The availability of the appointment should be checked on the Physician selected

- Meeting title
- Description
- Physician
- Date
- Time
- Reason – In case of edit or delete

View Schedule

A doctor should see display of all his appointments similar to Microsoft Outlook. Then he can select individual appointment and then go for Editing or Deleting it.

Admin Module

This module will take care of all the admin level features. It will have features like create Physician and Nurse users. Unlock patient users.

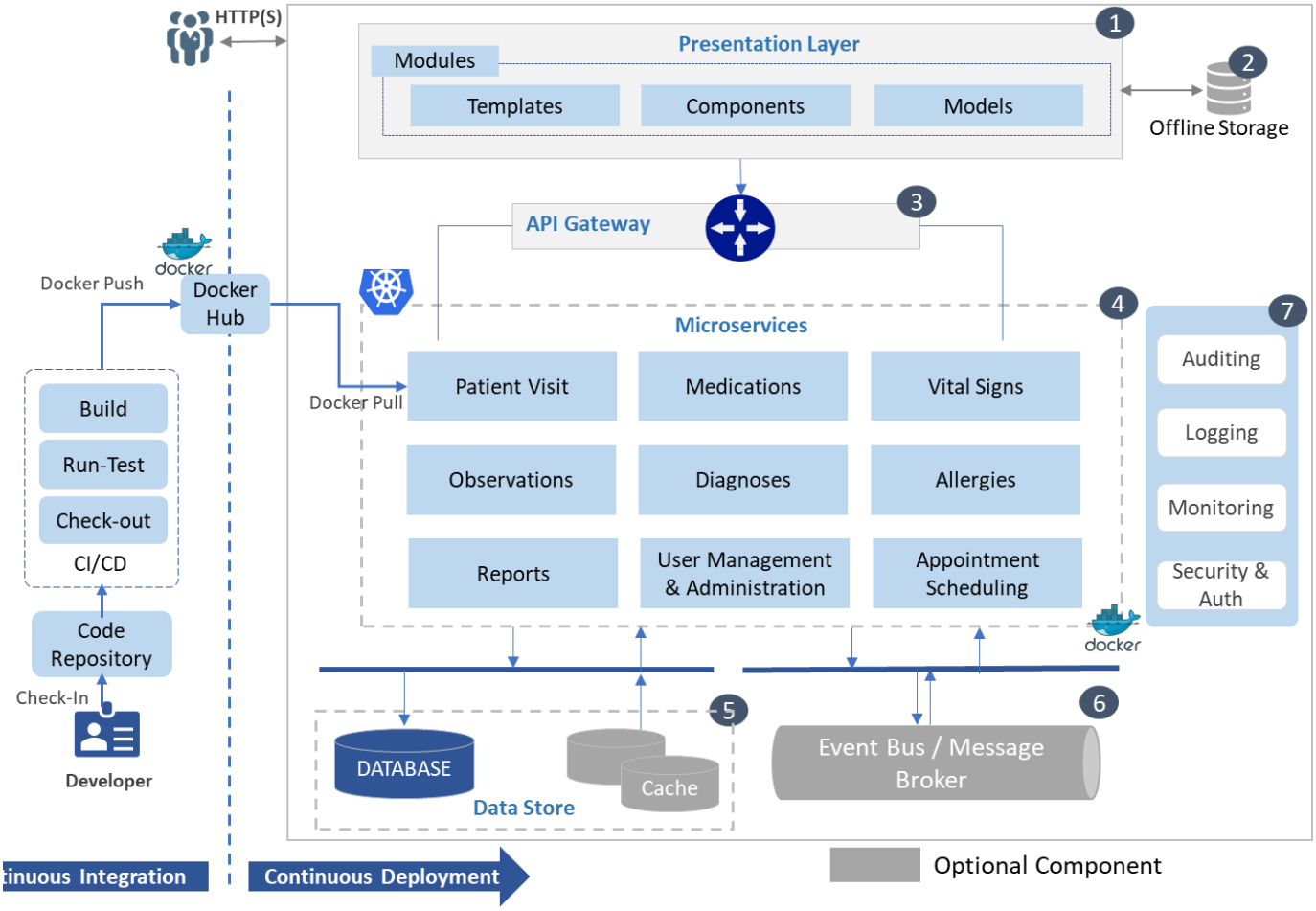
User Management

- Patient Users
    - Screen should show all the users with their details and status
    - Admin should be able to disable a user or unlock a user through the grid only
    - No hard deletes allowed
- Hospital Users
    - Screen should list all the hospital users
    - The grid should show option to edit or disable existing user
    - Admin should be able to Add the hospital user
    - Admin should be able to reset password for hospital user

Non-Functional Requirements

- Application should be hosted as HTTPS
- The application UI should be supported on google chrome and Edge
- The application should be responsive and should be able to access on Desktop, Mobile
- All the coding guidelines and code quality standards should be followed
- Unit Testing code coverage should be 80%
- Optional NFR
    - Should have endpoints to consume patient data (Optional)
    - Tablet support is optional
    - Unit Testing for UI is optional

Reference Architecture



Key Highlights

- Business Logic Isolation: Each service implements business logic for a specific function.
- Scalable: Each microservice can scale independently.
- High Availability: Kubernetes cluster ensures that the entire application is highly available and adds auto scaling capabilities.
- Fault Isolation: Microservices architecture makes it possible to isolate failures through well-defined service boundaries.
- Maintainability and Deployment: Each microservice can be deployed and maintained independently.
- Cross Platform: The application will be compatible across different OS platform.

Component Description

#	Component	Description	Tech Stack
1	Presentation Layer	User Interface (UI) will be developed using HTML5 , CSS3 and Angular 9 JavaScript Framework. These application components will be responsible for rendering User visualization elements, UI processing, data binding, event wiring and command dispatching.	▪ Angular 9 and HTML 5 with CSS3 ▪ Angular Material Controls ▪ Toaster notification ▪ **UT:** Protractor ▪ Bootstrap or Material design for responsive UI
2	Offline Storage (Optional)	Few screens should have ability to store the data offline	• Optional (Browser Based or Offline storage)
3	API Gateway	API Gateway will manage routing, API composition, caching, logging, authentication, and rate limiting	▪ Zuul API Gateway / Istio
4	Microservices	Microservices are collection of services which represent business capabilities. And are highly maintainable, testable, loosely coupled and independently deployable.	• .NET Core Web API 3.1 with Entity Framework Core or Spring Cloud Microservices • API Documentation: Swagger-Swash buckle 3.0
5	Kubernetes	A Kubernetes cluster is a set of node machines for running containerized applications.	• AKS
6	Docker	Docker is a container technology that allows a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and deploy it as one package.	• Docker
7	Cache (Optional)	Cache is an in-memory data store which will help in improving the performance of the application by keeping frequently accessed data in the server memory that can be written to and read from quickly	• Redis
8	Database	Database to the relation data and maintain users User may think of using non-relational database for storing patient reports	• SQL / NoSQL
9	Cross Cutting	**Authentication** – The services would be secured by Token based authentication. KeyCloak or Azure IAM or other Open Source would be used as a Token Provider service **Authorization**: Group based authorization would be happening at API service level (Screen level). Users mapped with a group would be able to view/edit/update/delete items based on Group permissions	• OAuth2 • NLog or Log4J • AppInsights

		Audit Logging: To maintain the history of actions(view/modify) performed across various modules **Exception Management**: Handling and logging all exceptions and displaying user friendly messages to deal with error conditions	
11	CI/CD	Continuous integration through code check-ins, run unit testing, create build to deploy. Continuous deployment using docker registry	• GitHub / GitLab • Azure DevOps

Project Plan

Sr No	Milestones	Deliverables	Actors	Estimated Days
1	M1	• Infrastructure Setup • Git repository folder structure • Database access and user creation • Project team finalization • Set-up Azure DevOps for tracking the project • Finalize the database tables	Trainer Dev Team	5
2	M2	• Design UI wireframes for the identified modules • Populate master tables like medication, allergies, diagnosis etc. • Identify and define authentication mechanism for login module • Define mock JSON structure for UI development for login module • Complete the UI functionality for Login module	Dev Team	6
3	M3	• Define Mock JSON for Admin and Dashboard module • Develop UI screens for Dashboard • Develop UI screen for ADMIN module	Dev Team	5
4	M4	• Define Mock JSON for Visit, Scheduling, Admin and Dashboard module • Develop UI screens for Visit module • Develop UI screen for scheduling module • Develop UI screen for ADMIN module	Dev Team	6
5	M5	• Create REST API for login module • Integrate it with UI • Perform UT for login module • Set up a CI/CD pipeline	Dev Team	5
6	M6	• Create REST API for admin and dashboard module	Dev Team	6

		• Integrate with UI • Perform UT • Server-side validations if any		
7	M7	• Create REST API for visit module • Integrate with UI • Perform UT • Server-side validations	Dev Team	6
8	M8	• Create REST API for scheduling module • Integrate with UI • Perform UT • Server-side validations	Dev Team	6
9	M9	• Integration of all the modules • Containerization • Deployment	Dev Team Trainer	5
TOTAL				**50**

Definition of Done (DoD)

- Completed activities should be demo ready
- Developed code should deployed and demoed
- All the code is unit tested (Manual or Automated)
- Modules should pass all the validations (UI and Business)
- At end of each milestone a demo will be given to Mentor

Assumptions

- Training team will provide the environment for development and hosting
- Training team to provide master tables for Medication, Allergies, Diagnosis, Procedure
- Project team would be of minimum 3 members
- Each day project team will spend approximately 3-4 hours on project work
- At end of each milestone a demo will be given to Mentor
- Trainers and Mentors can help to resolve doubts and provide guidance