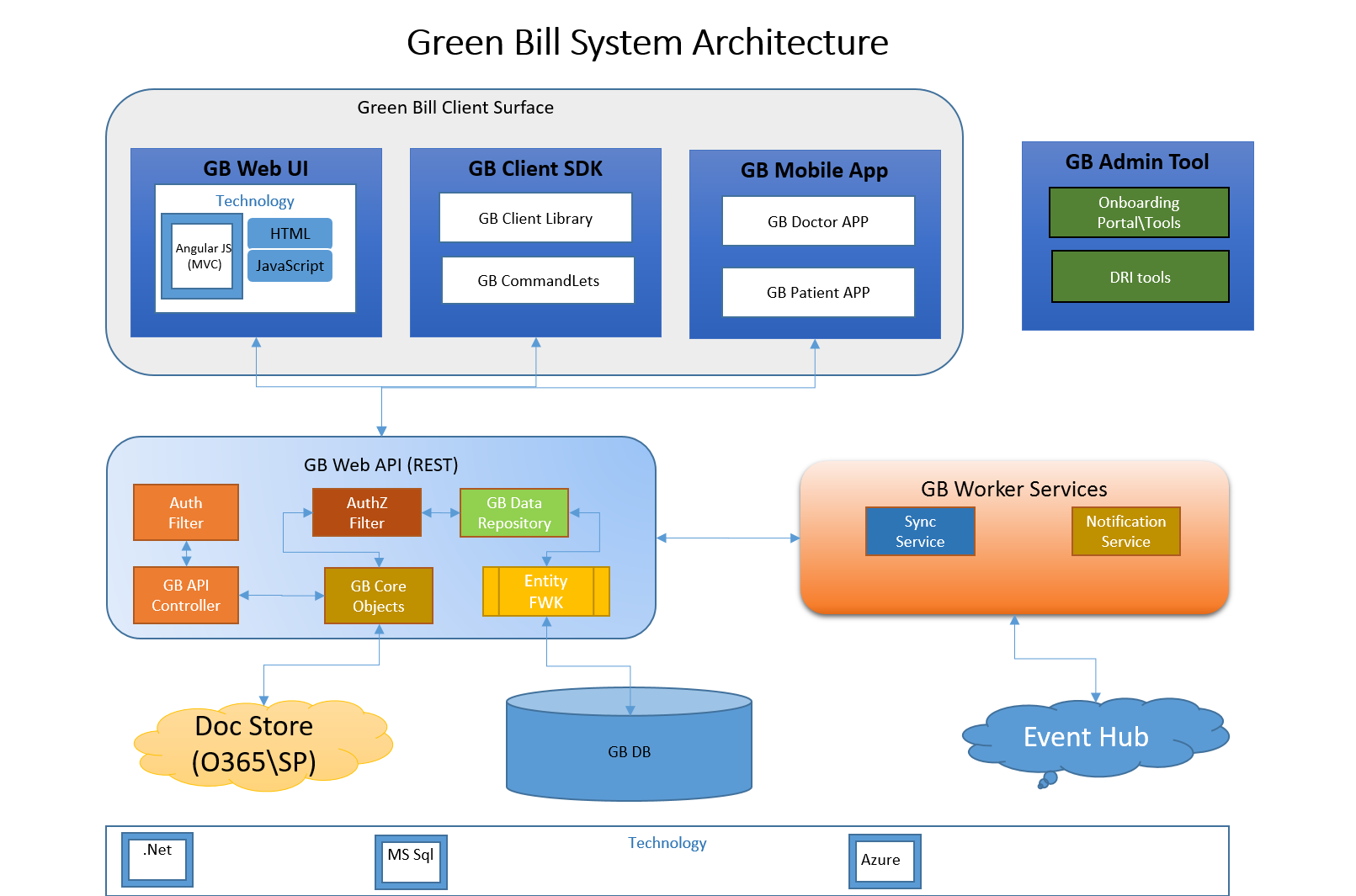
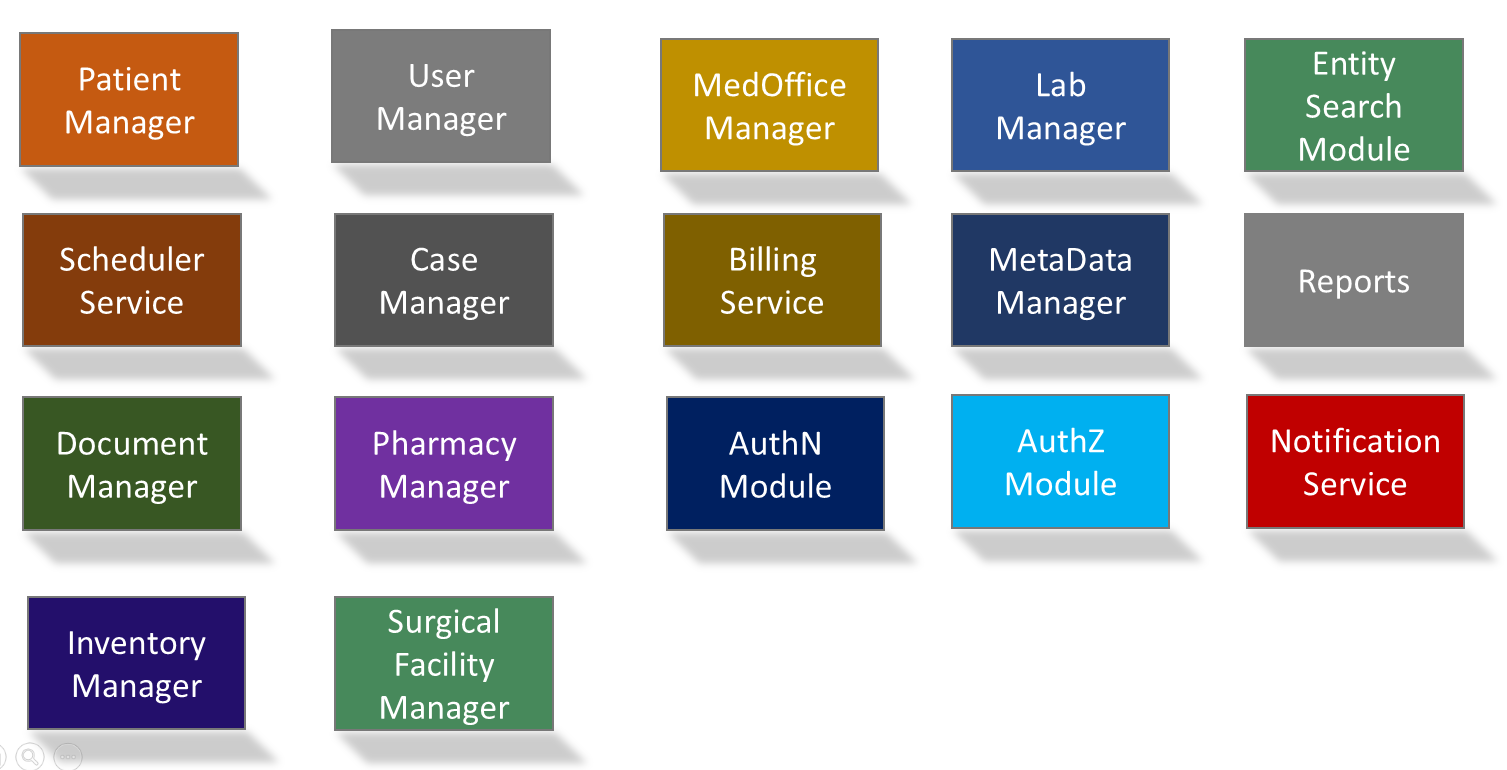
# Green Bill



Component Architecture



Patient Manager : This module provides the following functionalities

1. Create \ Update Patient Information
2. Bulk Upload Patient Information
3. View Patient Information
4. View Patient Visit History
5. View Patient Future Schedule

User Manager : This module provides the following functionality

1. Create\Update\Delete Users
2. View User Information
3. Set User Profile Settings
4. Set User Roles

MedOffice Manager: This module provides the following functionalities

1. Create\Update\Delete Medical Offices
2. Create Update\Delete Medical Providers for a Medical Office
3. Assign Owners to Medical Providers
4. View\Generate Reports for MedOffice

Lab Manager :

1. Create\Update\Delete Medical Labs
2. Set\View Medical Lab Schedule
3. Create\Update\Delete Medical Lab Machine Info.
4. Assign Owners to Medical Lab
5. View\Generate Reports for Medical Lab

Scheduler Service : This should be designed as a generic module for scheduling appointments for entities (like patients\doctors) to an entity with a schedule/

This module should provide the following functionalities

1. Create\Update\Delete a Schedule for Patient with a specific Entity . For e.g. Patient can be scheduled for a visit with a doctor or Lab for an MRI visit.
2. Integration with popular calendar management clients, i.e. the user who has an appointment should be able to view the same appointment to his\her Gmail\Hotmail calendar too.

Case Manager : This module provides the following functionalities

1. Create\Update\Close a Case
2. Create Case Document

.Patient Treatments related to a specific case



Billing Service : This module will provide the following functionalities

1. Generate Bills for Medical Service

A bill is a document generated which would list the services\items that have been provided to a patient. The Bill is also issued to an Insurance Provider

Master Data Manager \ Metadata Manager :

This module will provide functionalities to create\manage system master data for following entities

1. Insurance Providers
2. Specialty
3. Procedure Code

Document Manager : Document Manager needs to be developed as a generic service which will provide the following functionalities

1. Create\Manage Document templates
2. Associate tags with documents.
3. Create\Update\Delete Documents based on a template. The documents should have standard place holder for known tags and a new document should be prepopulated with the values for the known tags.
4. Create Documents by Scanning Documents using a scanner.

The document manager should be saved online and not locally.

Authentication Module: Authentication module will provide multi factor authentication to authenticate users to access Green Bill.

Authorization Module :

Authorization module should be a Role Based AccessControl.

Notification Service :

Green Bill Entity Model

1. EntityType

EntityType defines all the entities defined in Green Bills. Every Entity in GB will be classified as a specific entityType. Actions can be defined on specific Entity Types.

An EntityType can also have the

1. Actions

In GB a User can perform the following actions against an entity

1. Create
2. Update
3. Delete
4. Read
5. Schedule
6. Report
7. Bill
8. Roles

Green Bill have standard and custom roles. Standard Roles are system defined roles and system is aware of the access the user will have for these roles. Standard Roles would be defined for a specific Entity, for e.g. Account Owner Role will be for Account and is a standard role and users having this role would have access to all features in Green Bill and access every Entity under that Account.

1. Account

Every green customer has a Green Bill Account. Account is the topmost entity in the Green Bill entity hierarchy. As part of sign up a Green Bill account is created for a customer, the user who has signed up is the Owner of the Account and can create other entities and Users under the Account.

A green Bill account is equivalent of a company which owns one or more Medical Facilities.

1. User

A green bill user can log into Green Bill and perform actions based on the role the user has on various entities.

Users can also be classified as the following

1. Staff
2. Doctor
3. Patient
4. Attorney
5. Medical Facility

A medical facility is an office under an account which provides a certain type of Medical Service. There can be one or more Medical Facilities under an account. A Medical Facility can be of the following types

1. Medical Office

E.g. CitiMedQueens

1. Medical Lab

E.g. CitMedQueens Diagnostic

1. Pharma

E.g. DrugsRUS

1. Surgical Center

E.g. Excel surgery

1. Medical Providers

A Medical Office can have one or more medical providers who practice a specific set of Specialty. A Medical Provider will have doctors and one owner who is a doctor.

1. Medical Case

A medical case is always opened for a patient who is being treated by a Green Bill Facility.

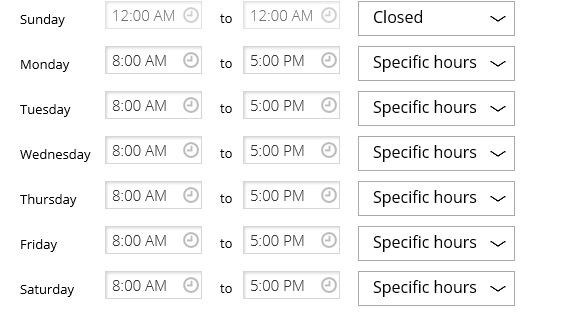
Medical Case is used to track the treatments and visits of a Patient and also for Billing the Treatments\Services provided to the customer. Documents related to Patient evaluation and Treatment are stored under a Case. Only Users associated with the Case will have access to these documents.

OPEN QUESTION : Will a Medical Case have a primary Medical Provider associated with it ?

A medical staff can create custom roles for users under the medical facility. These Roles should only be visible to the Medical Facility. The custom roles can be granted actions (Read\Edit\Create\Delete\Schedule\Bill) on specific entities (Medical Office\Mecical Lab) and reports.s

1. EntityScheduleSetting

All Entities that can be scheduled need to have a schedule setting which will be used to determine the availability of the entity.



1. EntitySchedule
2. Insurance Provider

An Insurance Provider is an external entity which will be billed for a specific Patients treatment. Green Bills will maintain the List of Insurance Provider which will be accessible to all Green Bill Accounts.

There would be cases where a specific Insurance Provider will not be present in the system and the Medical Office staff would need to create an Insurance Provider. We should let the staff create an Insurance Provider, this insurance provider should not be visible to other accounts other than the account which created the Insurance Provider. This custom created Insurance Provider would need to be evaluated by Green Bill and later on made accessible to all other accounts if it is a valid one.

General Rules for all Entities

1. All changes to an entity must be audited. The audited data needs to be retained based on a system specific configuration.
2. Entities should not be hard deleted, they should be hard deleted.

Green Bill Sign Up Flow

Document Manager

Overview : Document Manager as the name suggests is a service which will allow creating , editing and viewing documents online. It should support 2 most popular type of documents Word and PDF. The document management service should be a multitenant service. Each tenant should be able to register with the document service. The storage (known supported storage types only) of the documents should be configurable per tenant. If the location is not set then the document management service should use the default storage ( FTP server or GitHub) configured for the service.

Document Manager will compose of 3 distinct components

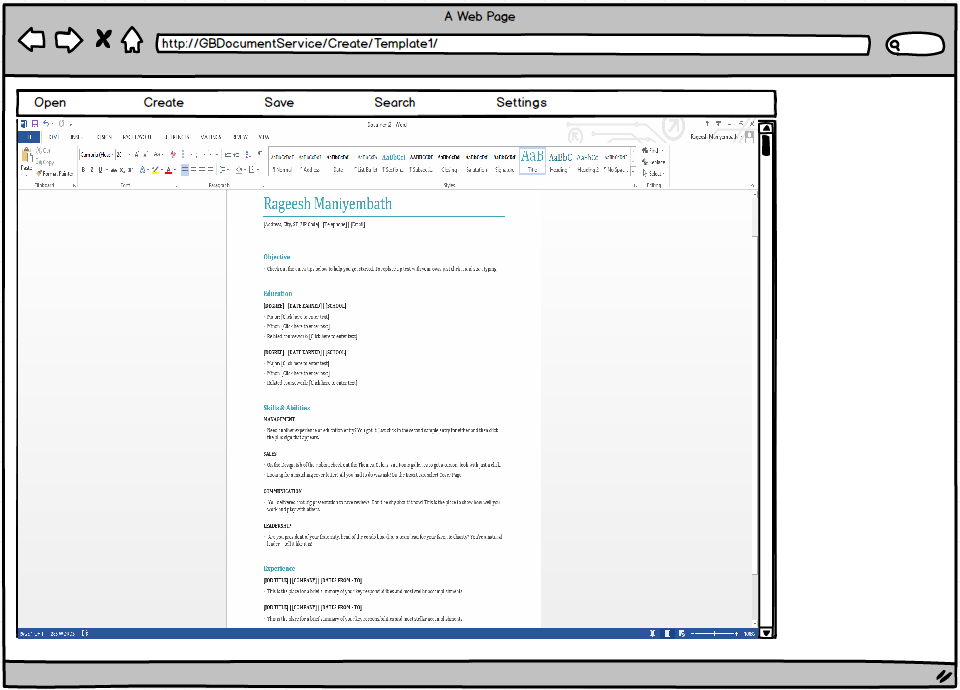
1. Document Management and Storage Web API.

This API should be a web service that would expose the necessary API’s to store and retrieve documents related to a tenant.

1. Document Management Client Lib

This could be server side library which is used to interact with the Document Service and is also responsible for rendering the HTML for the browser.

1. Document Management Client Plugin



The client should be a browser plugin and also an add-in. The server side info should be configurable for the client side plugin. The client plugin should be initialized with the info it needs to interact with the server side service. The browser plugin should have 2 frames

1. Menu Frame

The Menu frame should have the following options

* Create : Option to create new documents from existing document on the client box or a new document from an existing Template.
* Open : Load an existing document on the server , this document can be edited on the client side and the changes uploaded to Server when the document is Saved.
* Save : Option to Save document in the Document editor Frame. The saved document is saved remotely.
* Search : Option to search documents for the specific Tenant.

1. Document editor Frame

This is the frame which will load word or pdf editor , which will allow the user to edit the document. This requires the client to have the required software like MS word or PDF writer required for editing the type of document.

Template Based documents:

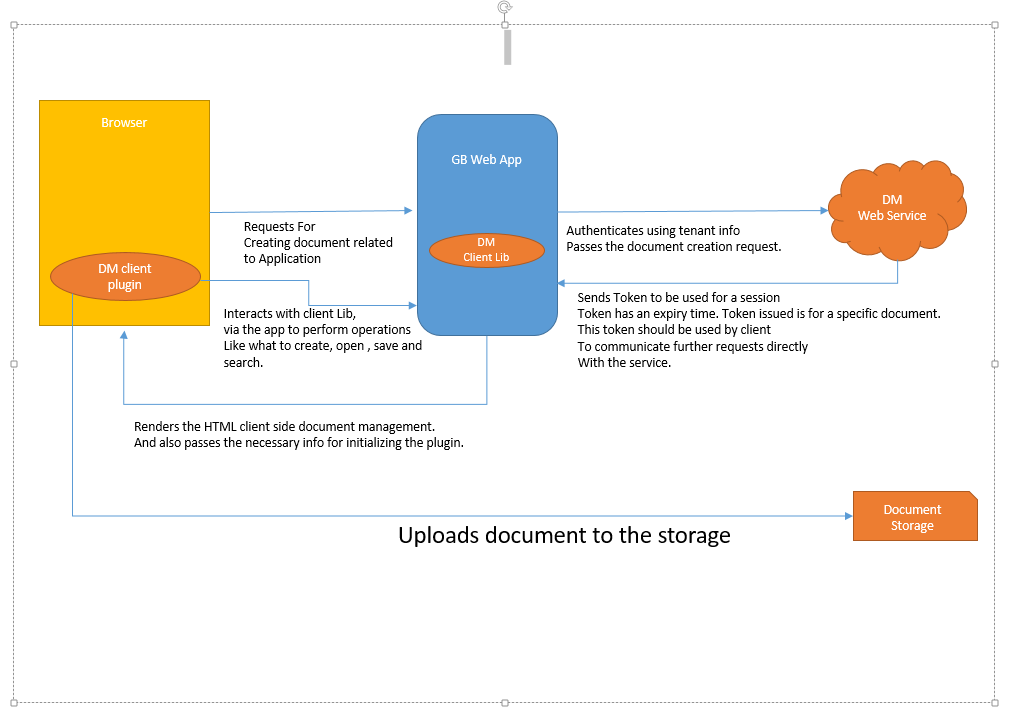
Document Manager should allow Tenants to define document templates. A user can define tags (place holders) in the document template. DM should allow to create a new document based on an existing template and the set of values for the tags. The new document should have the tags replaced with the values provided during creation.

Document versioning:

The service should track all the changes done to a document, every save should result into a new version of the document. One should be able to view the previous version of a document. Latest version of the document should be the default document to be used unless a specific version is requested.

Flow diagram for managing documents via document Manager

The components in Orange are part of the Document Manager.



DM

Web Service

Document Storage

GB Web App

Browser

Requests For

Creating document related to Application

Authenticates using tenant info

Passes the document creation request.

Sends Token to be used for a session

Token has an expiry time. Token issued is for a specific document.

This token should be used by client

To communicate further requests directly

With the service.

Renders the HTML client side document management.

And also passes the necessary info for initializing the plugin.

Uploads document to the storage

Interacts with client Lib,

via the app to perform operations

Like what to create, open , save and search.