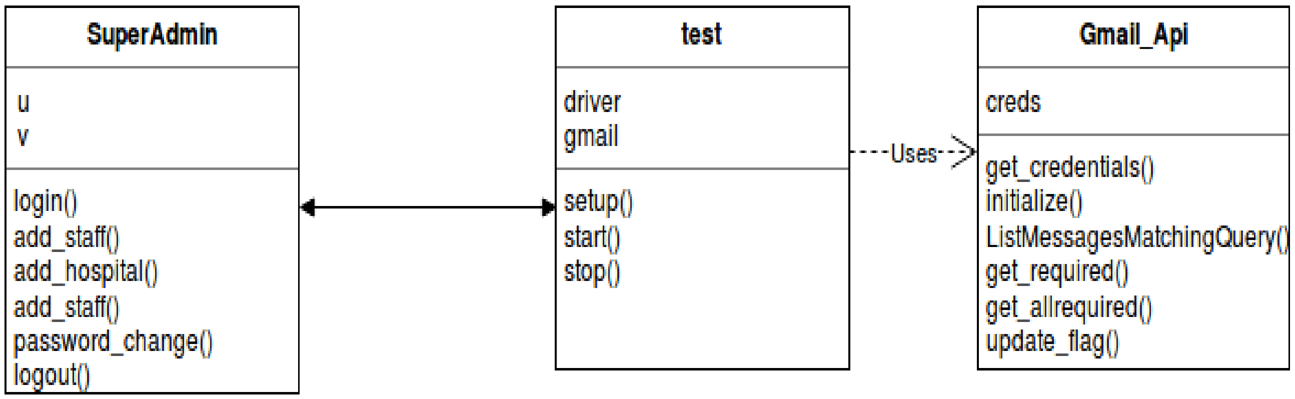
Product Design

Team Number	**Team 38**
Team Name	**SSAD31**
Team members	**1. Sri Harsha Vuyyuri**
	2. Karthika Ramineni
	3. Rishika Sharma

Design Model

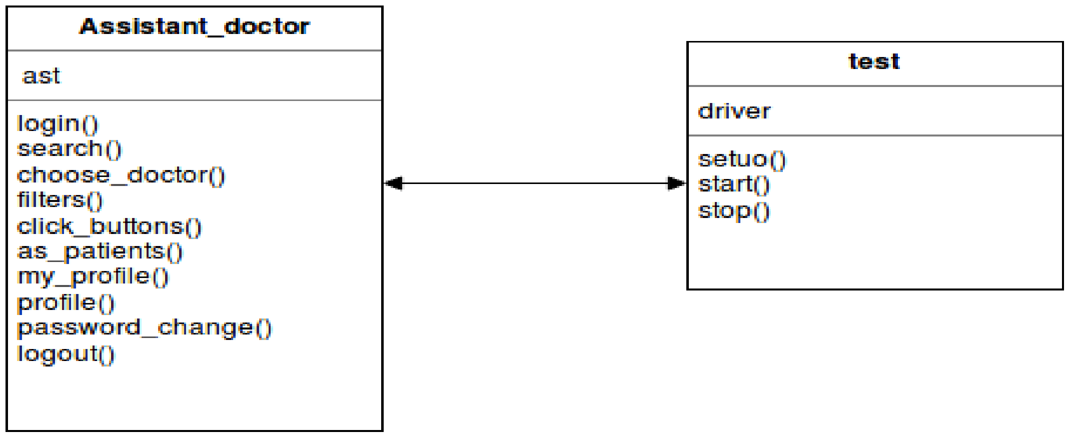
Super Admin role:



1. SuperAdmin	Class state • It contains the email id of the hospital • It contains the email id of the hospital admin Class behavior • all the functions in this class automate the verification of following functionalities • login(): the super admin is logged in • add_staff(): call center staff get added and verified if they are displayed • add_hospital(): hospitals get added and verified if they are displayed • password_change(): the password for super admin gets changed

	• logout(): the super admin is logged out
2. test	Class state • It contains the driver object • It contains the gmail api object which is used for fetching login credentials from gmail Class behavior • setup(): creates instances for the driver and gmail api classes • start(): this methos calls all the methods in SuperAdmin class and prints appropriate messages after automation of each feature • stop(): closes the driver connection
3. Gmail_Api	Class state • creds stores the login credentials of the user (super admin) Class behavior • this class is used to fetch the email ids and passwords of hospitals and call center staff from the emails sent by onward health

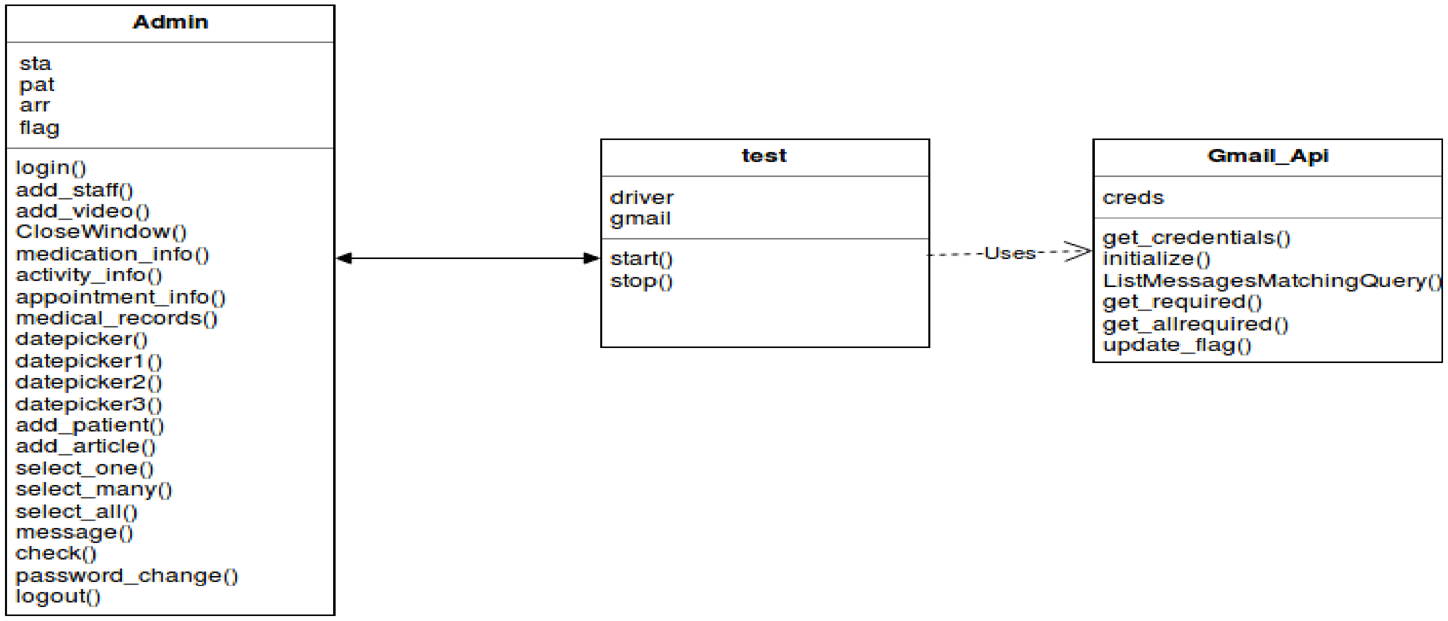
Assistant Doctor role:



1. Assistant_doctor	Class state • It contains the email id of assistant doctor Class behavior • all the functions in this class automate the verification of the following functionalities • login(): the assistant doctor is logged in • search(): searches for patients • choose_doctor(): doctor gets chosen • filters(): tests the working of search boxes • as_patients(): displays the patients under the doctor which we choose • my_profile() and profile() are used to edit the profile of the assistant doctor • password_change(): the password for super admin gets changed • logout(): the assistant doctor is logged out • click_buttons() is a helper function

2. test	Class state • It contains the driver object • It contains the gmail api object which is used for fetching login credentials from gmail Class behavior • setup(): creates instances for the driver and gmail api classes • start(): this methods calls all the methods in Assistant_doctor class and prints appropriate messages after automation of each feature • stop(): closes the driver connection

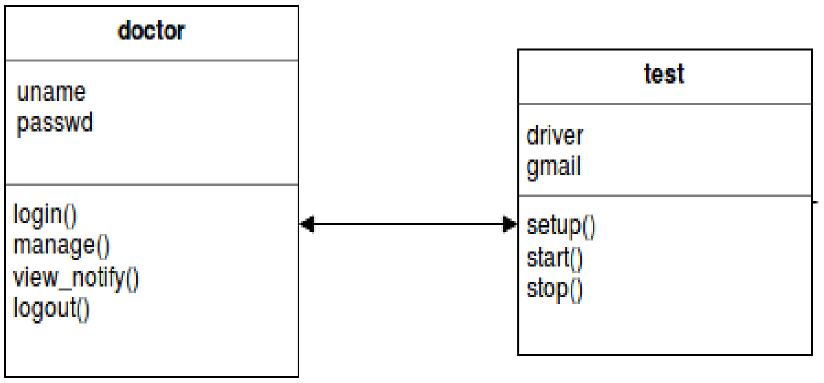
Hospital Admin role:



1. Admin	Class state • it contains the email ids of staff and patients • It contains the ids of patients • It stores the state of a patient (whether the patient is activated or deactivated) Class behavior • all the functions in this class automate the verification of the following functionalities • login(): the hospital admin is logged in • add_staff(): staff get added and verified if they are displayed • add_video(): videos get added and verified if they are displayed • add_article(): articles are added and verified if they are displayed, CloseWindow() is a helper funtion • message(): sends messages • password_change(): the password for hospital admin gets changed • logout(): the hospital admin is logged out • the rest of the methods are helper methods for the above functions
2. test	Class state • It contains the driver object • It contains the gmail api object which is used for fetching login

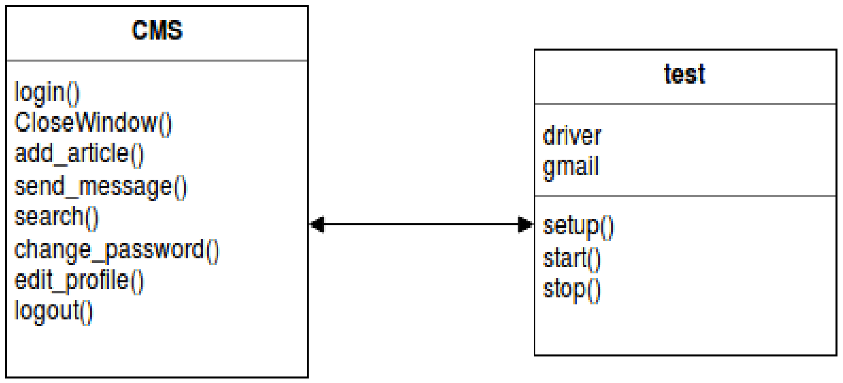
	credentials from gmail Class behavior • start(): this method calls all the methods in Admin class and prints appropriate messages after automation of each feature • stop(): closes the driver connection
3. Gmail_Api	Class state • creds stores the login credentials of the user (hospital admin) Class behavior • this class is used to fetch the email ids and passwords of the staff(doctors, assistant doctors, cms) from the emails sent by onward health

Doctor role:



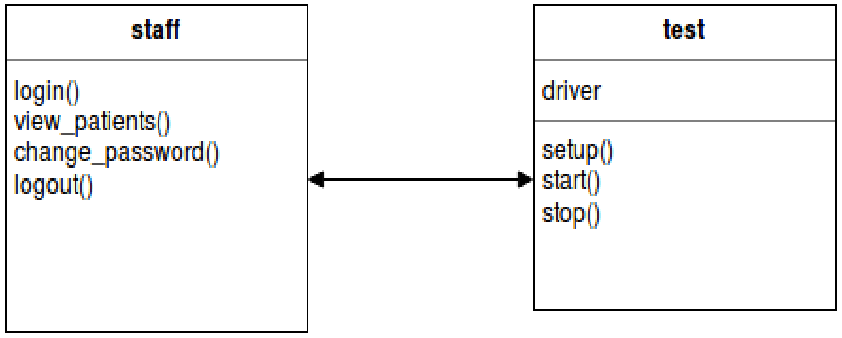
1. doctor	Class state • It has the login credentials of doctor Class behavior • all the functions in this class automate the verification of the following functionalities • login(): the doctor is logged in • manage(): this method can grant access to the assistant doctors • view_notify(): checks the view notifications page • logout(): the doctor is logged out
2. test	Class state • It contains the driver object • It contains the gmail api object which is used for fetching login credentials from gmail Class behavior • setup(): creates instances for the driver and gmail api classes • start(): this method calls all the methods in doctor class and prints appropriate messages after automation of each feature • stop(): closes the driver connection

CMS role:



1. CMS	Class behavior • all the functions in this class automate the verification of the following functionalities • login(): the CMS is logged in • add_article(): articles arr added and verified if they are displayed, CloseWindow() is a helper function for adding article • send_message(): messages are sent • search(): search boxes functionality is verified • change_password(): the password for CMS gets changed • edit_profile(): the profile gets edited • logout(): the CMS is logged out
2. test	Class state • It contains the driver object • It contains the gmail api object which is used for fetching login credentials from gmail Class behavior • setup(): creates instances for the driver and gmail api classes • start(): this method calls all the methods in CMS class and prints appropriate messages after automation of each feature • stop(): closes the driver connection

Call Center Staff:

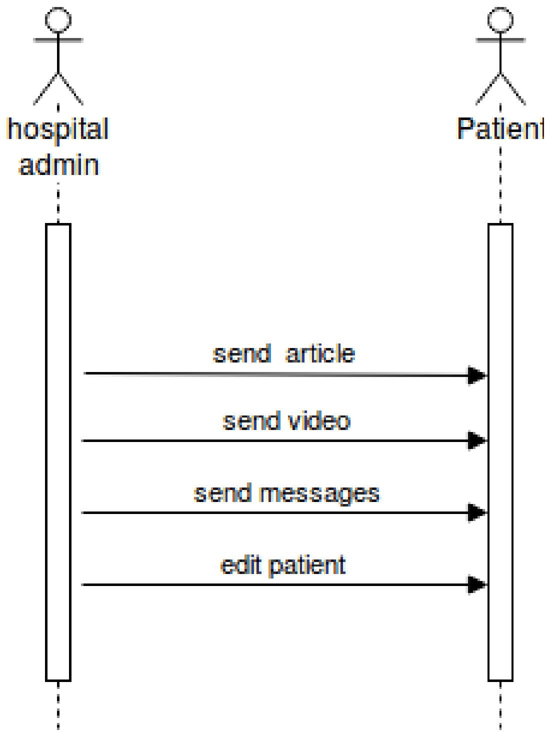


1. staff	Class behavior • all the functions in this class automate the verification of the following functionalities • login(): the call center staff is logged in • view_patients(): checks whether patients are displayed • change_password(): the password of call center staff gets changed • logout(): the call center staff is logged out
2. test	Class state • It contains the driver object • It contains the gmail api object which is used for fetching login credentials from gmail Class behavior • setup(): creates instances for the driver and gmail api classes • start(): this method calls all the methods in staff class and prints appropriate messages after automation of each feature • stop(): closes the driver connection

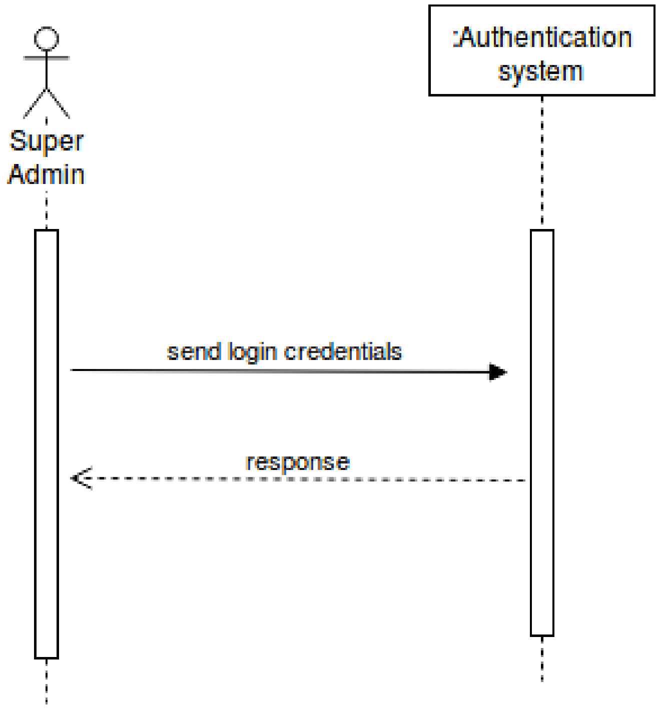
.

Sequence Diagram(s)

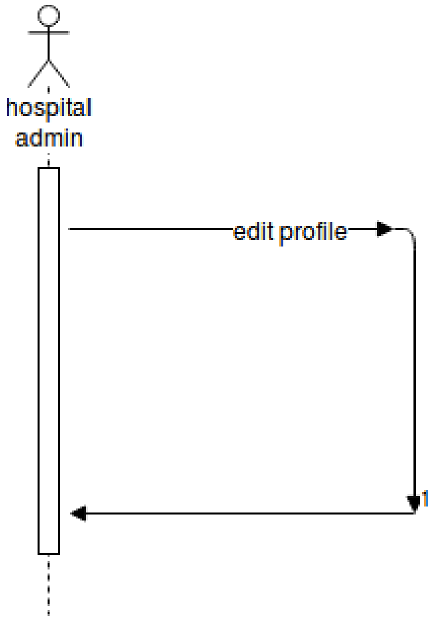
<u>Send messages, videos, article, edit patient:</u>



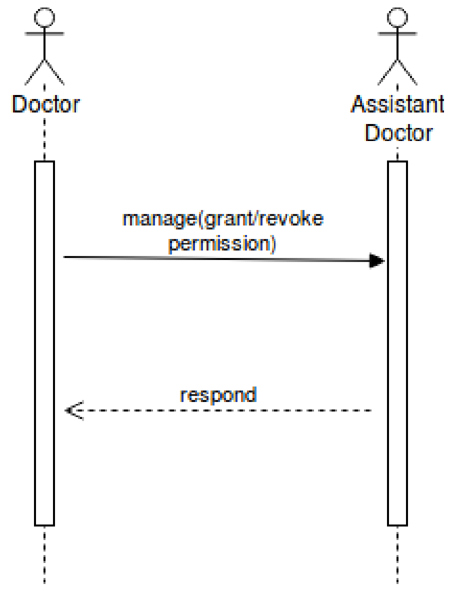
Login:



Edit profile:



Manage assistant:



Design Rationale

1) Used config file to store the URL, email ids, passwords, counts and various other counts. In the beginning we use to make changes in the actual files(search for the ids,….). Now it has become easy because we can directly make the changes in the config file.

2) Changed the if conditions to while loops in Hospital Admin add_staff function. This made our task easier because we can now get the login credentials and store them as a group instead of separately getting the login credentials and then combining them.

3) During the process of automation we found some things(surgery description was not getting added in patient, after password change we are getting redirected to an error page, user name getting changed only after re-login when profile is edited) which need to be changed in the website.