|  |
| --- |
|  |

Registration

**Narrative Description**

Registration features is for super admin user and doctors. Patients, doctors, and community members are required to be registered users in order to access their accounts. Registration page is not available to visitors. Only doctors of the hospital and super admin user have the privileges to register new users, however, doctors are allowed to register only patients but a super admin user can register user under any role. After login super admin user or doctor will see a link to registration page on dashboard. First of all registration page will display all the existing users sorted by their roles and names. This page will have another link to the registration form. Now at registration time doctors will not be asked to select the role because doctors are allowed to register only patients but super admin user must have to select a role before going forward. User names, emails, Identification, and Health Card Numbers are to be unique and if any of these already exists in the database, registration will be unsuccessful. Now if registration is successful, doctors or super admin user will again see the lists of all users. If there is need to change the information of any user, it can be done on the same registration page. All the doctor or super admin user have to do is to click on the name of the user and the information will be loaded into fields on registration form. Here information can be updated. Doctors can change the information of patients only. Super admin user can lock or delete any user on the system but doctors are able to delete only patients.

All the information is stored in various relational data tables in the database. For the registration following tables are used:

Users

|  |  |
| --- | --- |
| User Id | Unique Identifier **(PK) (Auto Generated)** |
| Application Id | Unique Identifier **(FK)** |
| User Name | Text 256 characters |
| Mobile Alias | Text 16 characters |
| Is Anonymous | True / False |
| Last Activity Date | Date |

Users In Role

|  |  |
| --- | --- |
| User Id | Unique Identifier **(FK)** |
| Role Id | Unique Identifier **(FK)** |

Membership

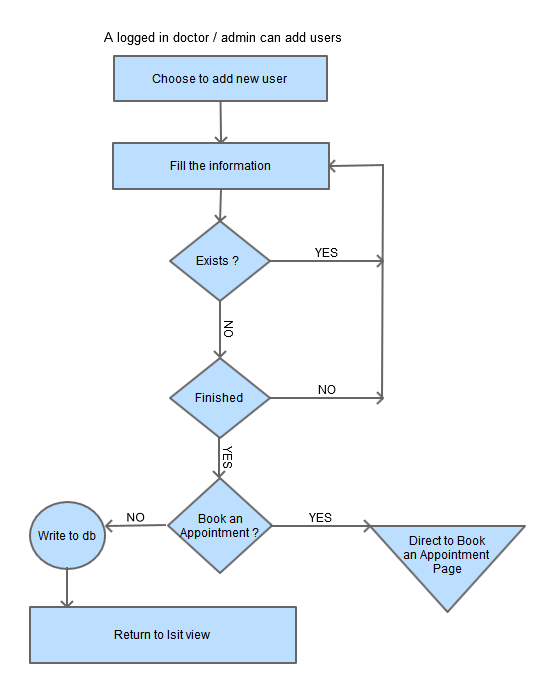
|  |  |
| --- | --- |
| User Id | Unique Identifier **(FK)(PK)** |
| Application Id | Unique Identifier **(FK)** |
| Password | Text 128 characters |
| Password Format | Numeric |
| Password Salt | Text 128 characters |
| Mobile PIN | Text 16 characters |
| Email | Text 256 characters |
| Lowered Email | Text 256 characters |
| Password Question | Text 256 characters |
| Password Answer | Text 256 characters |
| Is Approved | True / False |
| Is Locked | True / False |
| Create Date | Date |
| Last Login Date | Date |
| Last Password Changed Date | Date |
| Last Lockout Date | Date |
| Failed Password Attempt Count | Numeric |
| Failed Password Attempt Window Start | Date |
| Failed Password Answer Attempt Count | Numeric |
| Failed Password Answer Attempt Window Start | Date |
| Comment | Text Max characters |

UserBasicInfo

|  |  |
| --- | --- |
| User Basic Info Id | Unique Identifier **(PK)(Auto Generated)** |
| User Id | Unique Identifier **(FK)** |
| First Name | Text 50 characters |
| Last Name | Text 50 characters |
| Date of Birth | Date |
| Gender | Text 1 characters |
| Identification | Text 20 characters |
| Address | Text 100 characters |
| City ID | Unique Identifier **(FK)** |
| State Id | Unique Identifier **(FK)** |
| Postal Code | Text 7 characters |
| Phone | Text 15 characters |
| Fax | Text 15 characters |
| Family Doctors | Text 50 characters |
| Department | Text 50 characters |
| Join Date | Date |
| Speciality | Text 50 characters |
| Community Group Name | Text 100 characters |

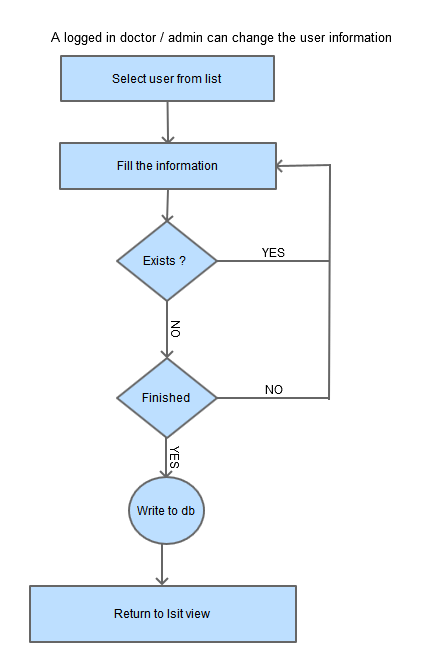
**The Dataflow Models**

Here is process of new registration in terms of data flow models

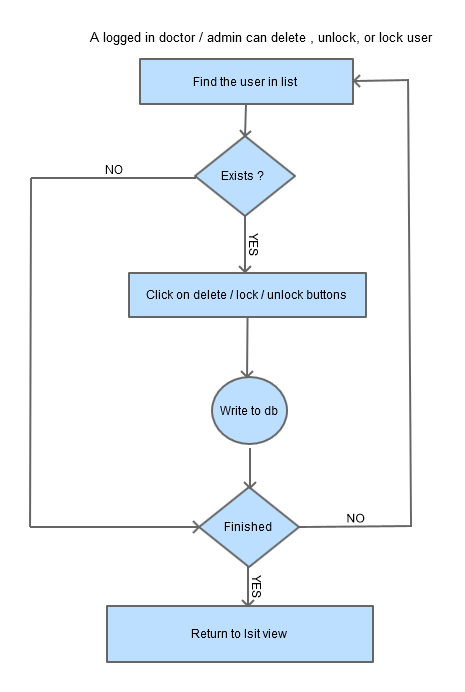


It is expected that admin or doctors are logged in. They provide the required information and if this information already exists in the database then user will be asked to re-enter new information or to quit. Once user is done with filling with information, a query will execute to save the information into the database. But if admin or doctor is registering patient then they will see another button on the form page which is a link to appointment page. Now if it is patient they might need to book an appointment for the patient and then they will click on this button which will redirect them to appointment booking form page.

Now if admin wants to edit the information of any existing user, then he will select the user from list and submit the form after changes. Here is data flow model to demonstrate the process of information change

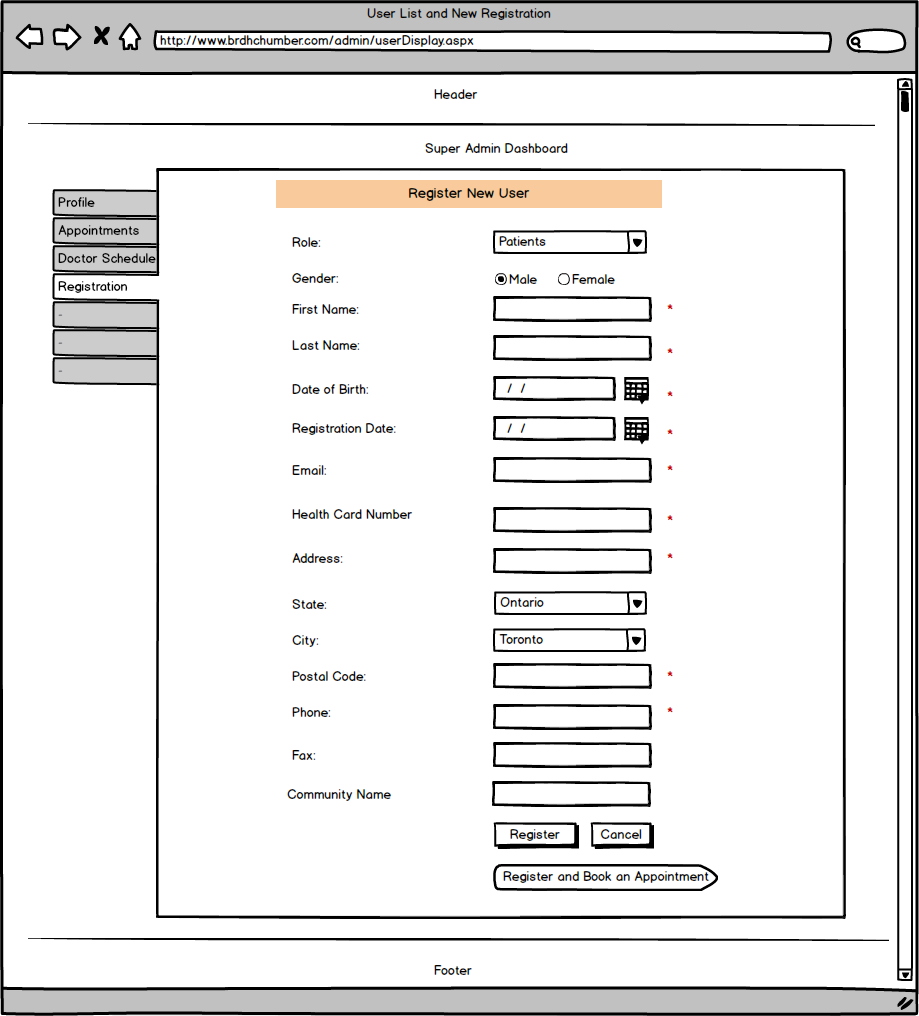


Following flow chart explains the process of deleting, locking, or unlocking of users by admin or doctors.

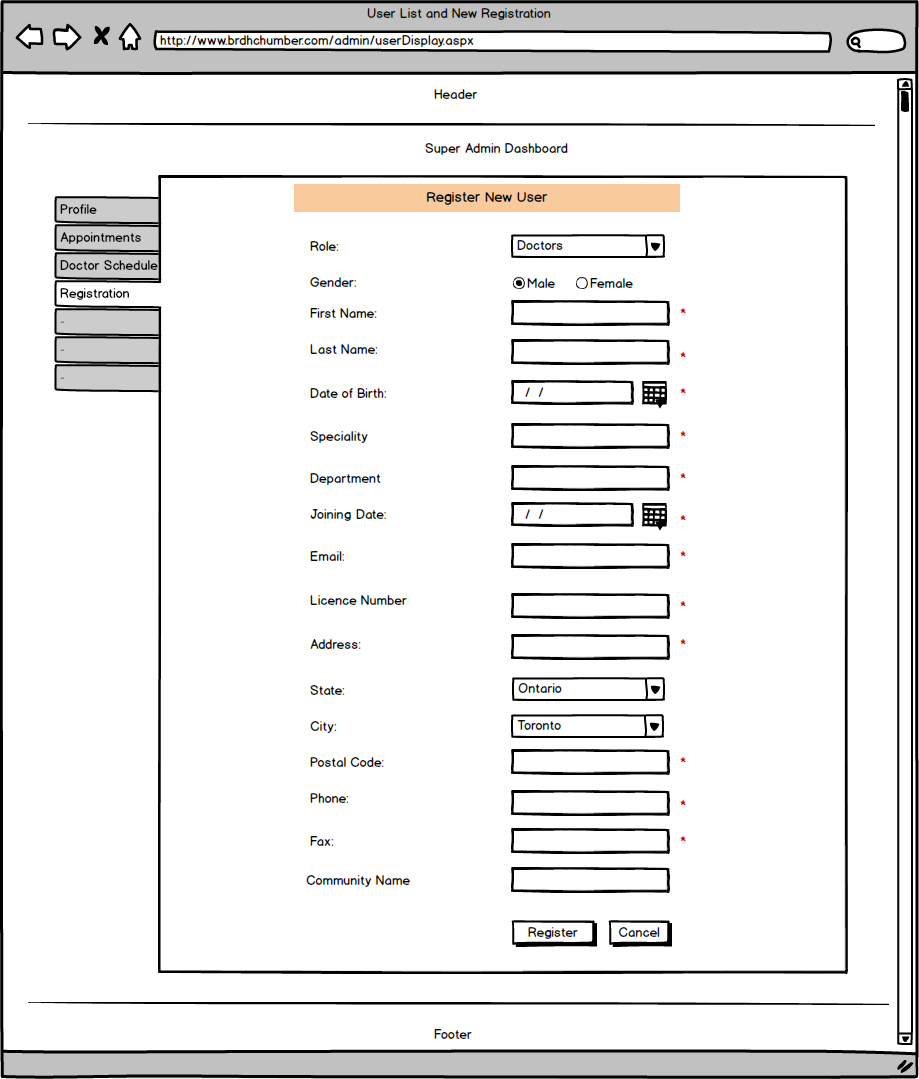


**Wireframes**

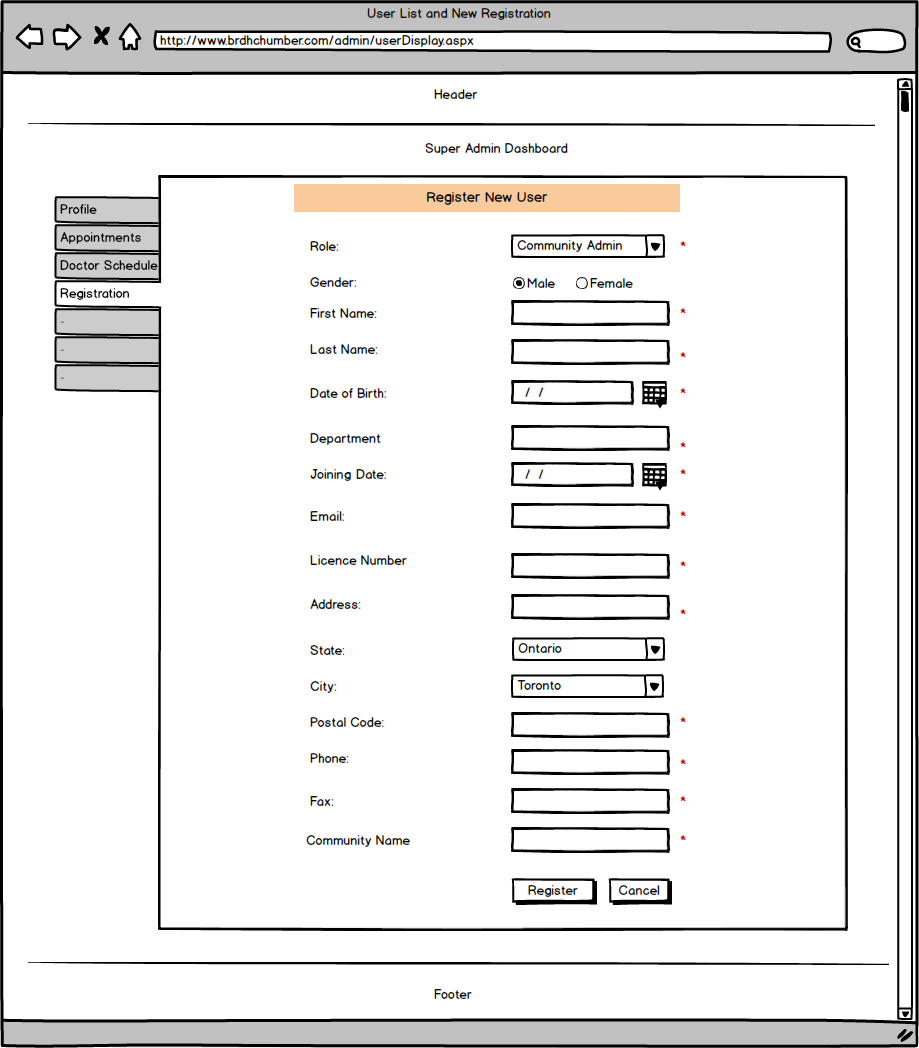
Here is the wireframe of adding new patient by super admin



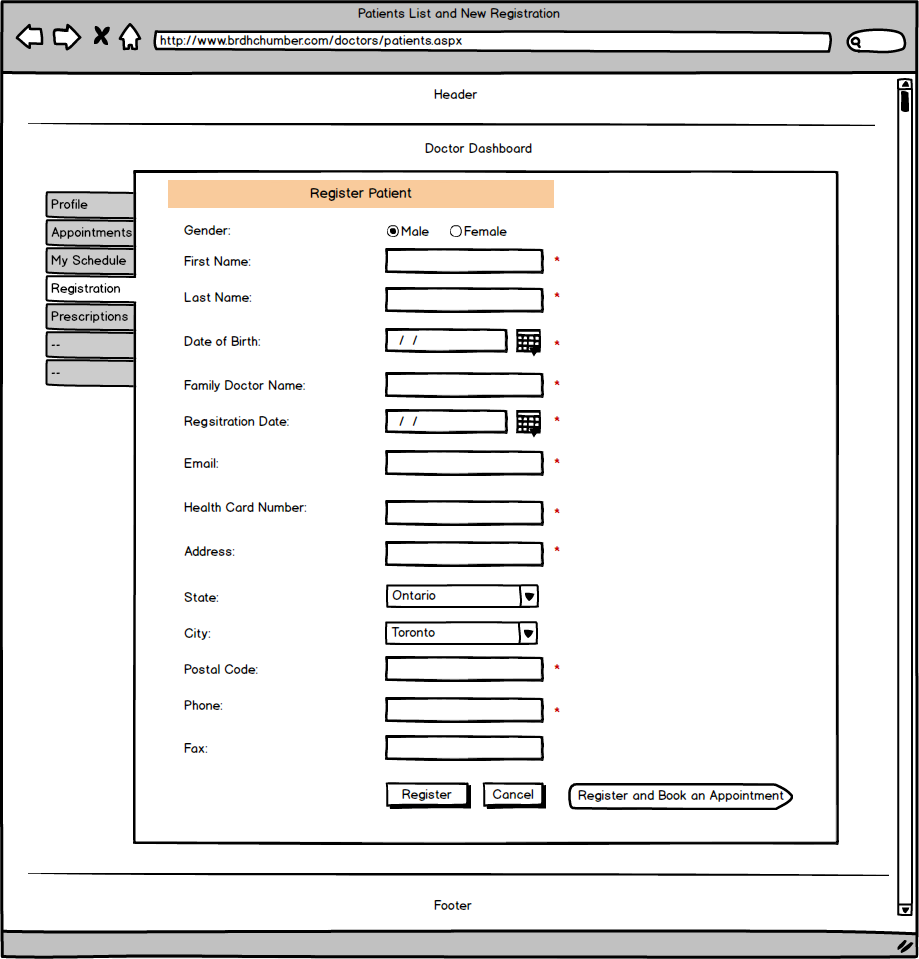
Here is the wireframe of adding new doctors by super admin



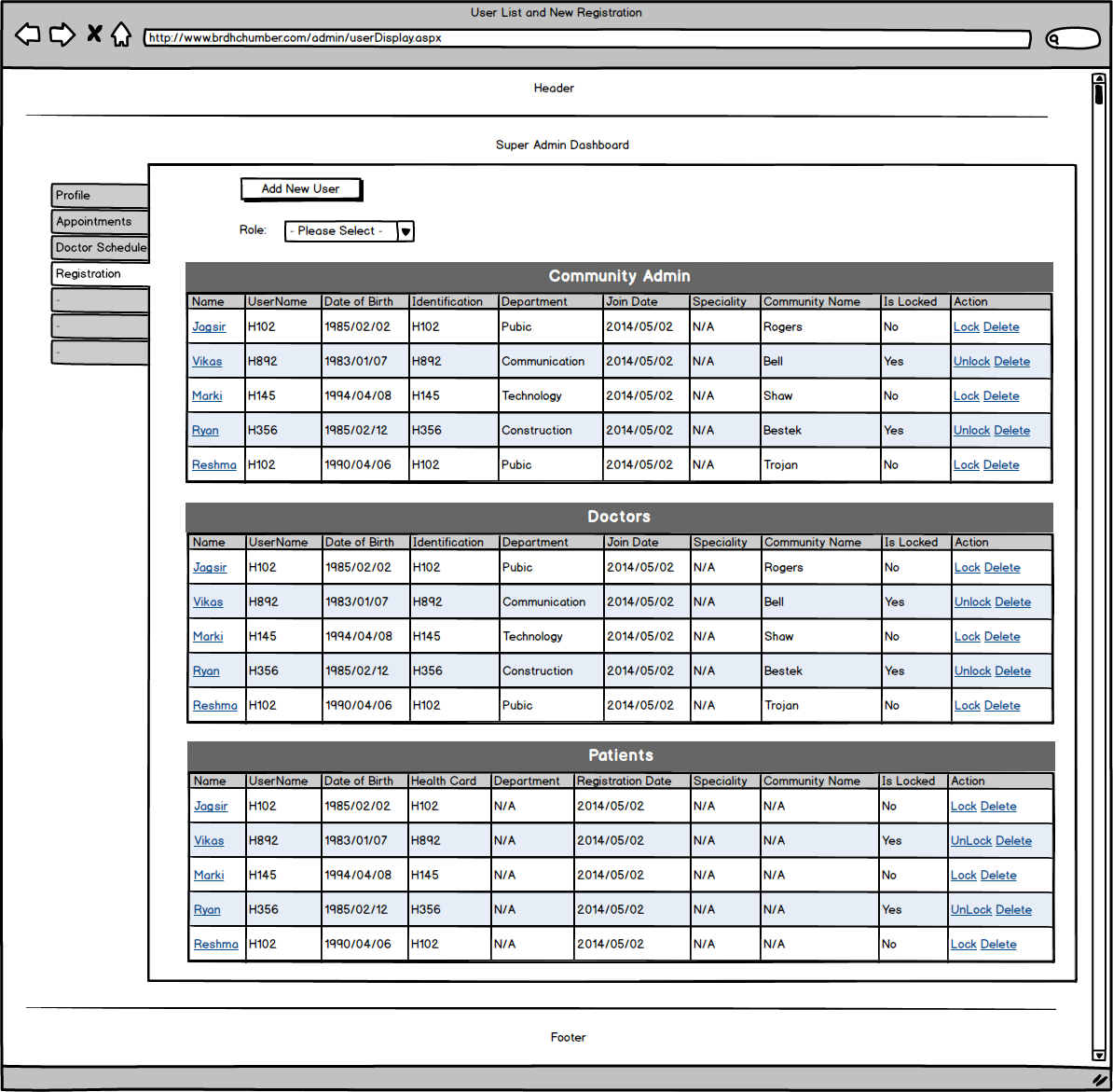
Here is the wireframe of adding new community members by super admin



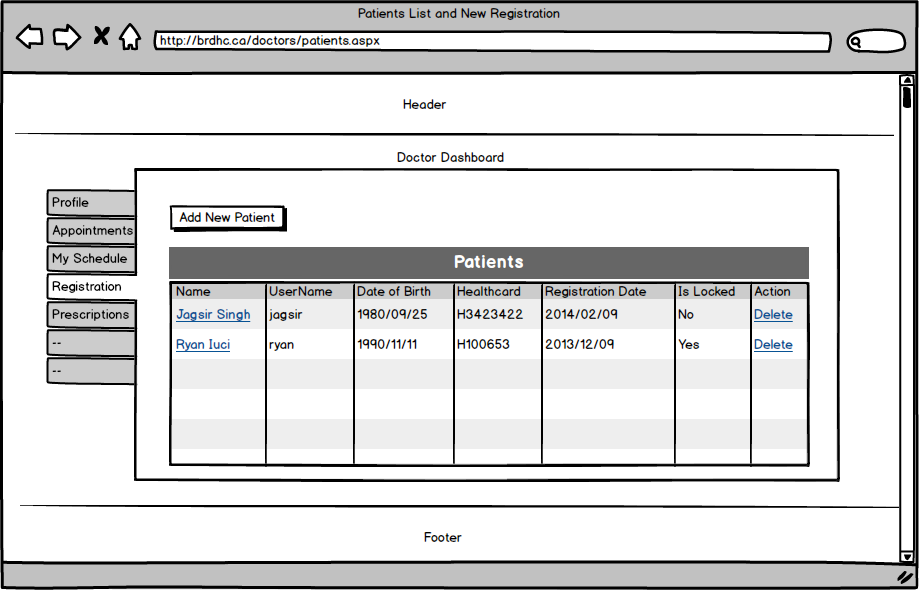
Here is the wireframe of adding new patient by Doctor



Here is the wireframe of all users list view to super admin account



Here is the wireframe of patients listing for doctors account



Book an Appointment

**Narrative Description**

This feature is for only admin and doctors. To book an appointment, patient must be registered first. So if patient is not registered, user cannot book an appointment. Admin and Doctors need to login into their accounts for this functionality. Once they logs in, they will see link to appointments page. First, this link will take them to appointments list page where they can see the all appointments.

Now in order to book an appointment, they need to click on Book an Appointment button. This will display the form. Now users will search the patient by Health Card number. Most important thing here is to check the doctor’s schedule. Only available date time can be allocated to patient.

Once the appointment is made, patients will get an email notification and doctor’s schedule will also be updated. Appointments made by super admin or doctors do not need to be approved but if patient request for the appointment it will need to be approved by doctors.

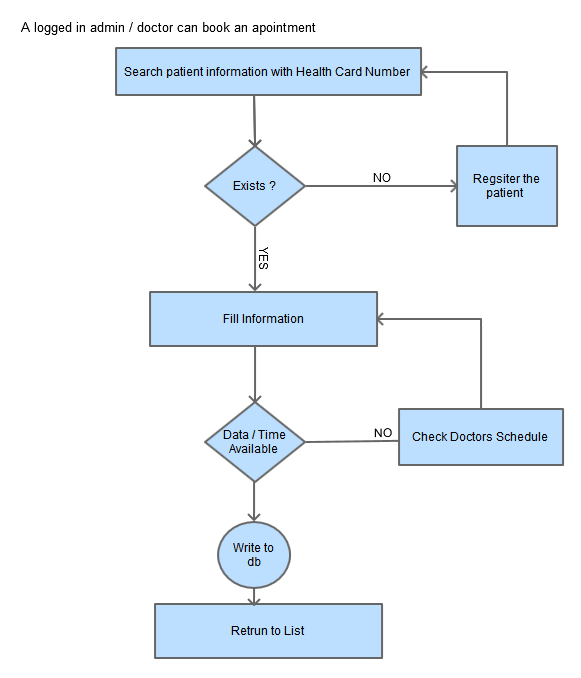
All the information will be saved into database tables that look like below.

brdhc\_PatientAppointments

|  |  |
| --- | --- |
| Appointment Id | Unique Identifier **(PK) (Auto Generated)** |
| Patient User Id | Unique Identifier **(FK)** |
| Doctor User Id | Unique Identifier **(FK)** |
| Appointment Date | Date |
| Appointment Time | Text 10 characters |
| Reason | Text MAX characters |
| Is Approved | True / False |

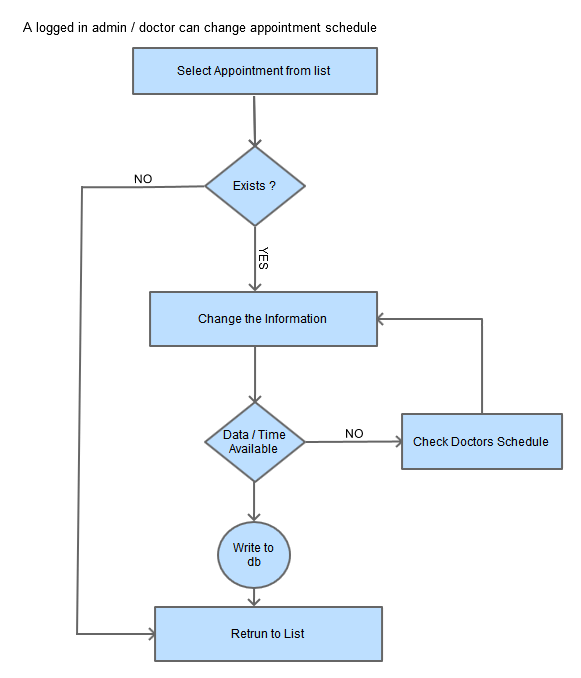
**The Dataflow Models**

Here is the data flow model to present the process of booking an appointment

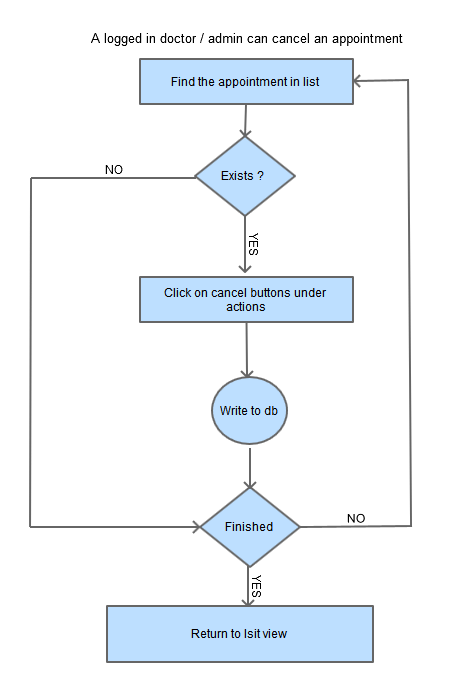


In case patient is not willing to come on the selected date, doctors or admin can reschedule the appointment. All they have to do is to select the appointment from listing and same form will display with appointment information in fields. Again admin or doctors have to check doctor’s schedule.

Here is the flow chart to explain the process steps

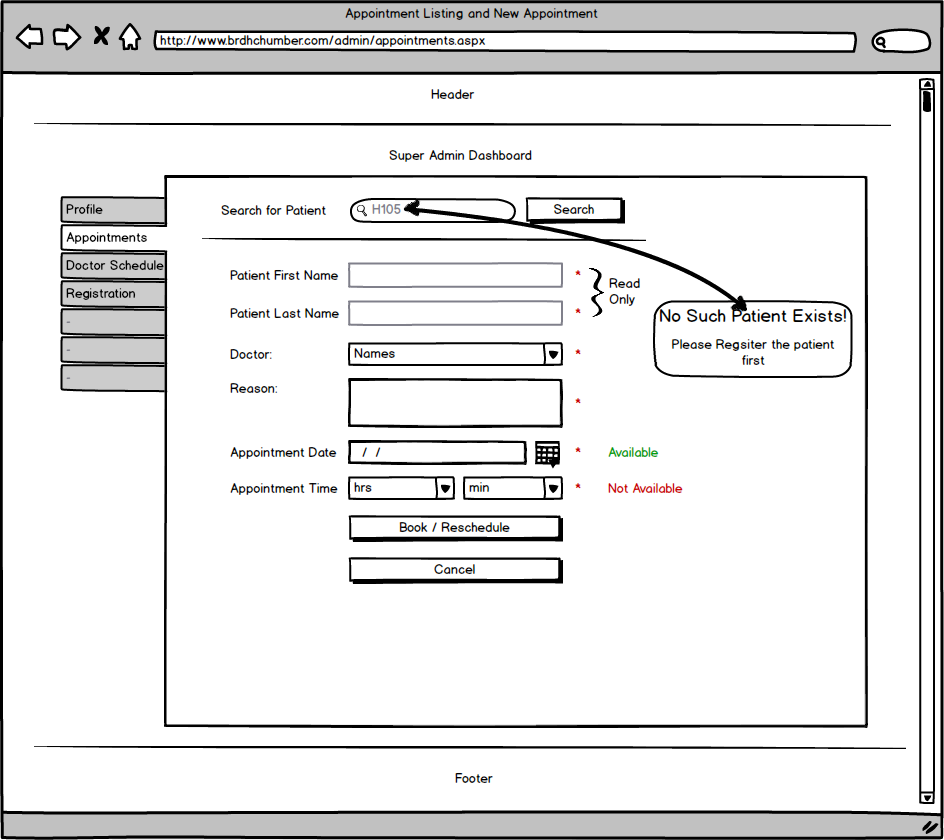


Patient can make request to cancel the appointment and following is the data flow model to define the process of appointment cancelation.

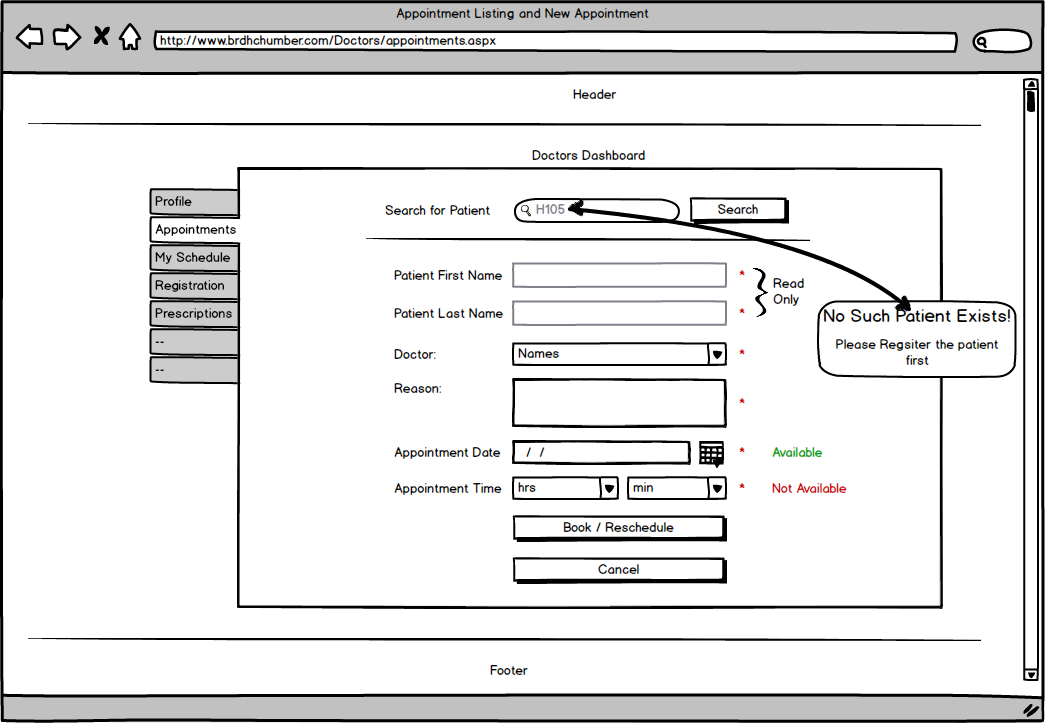


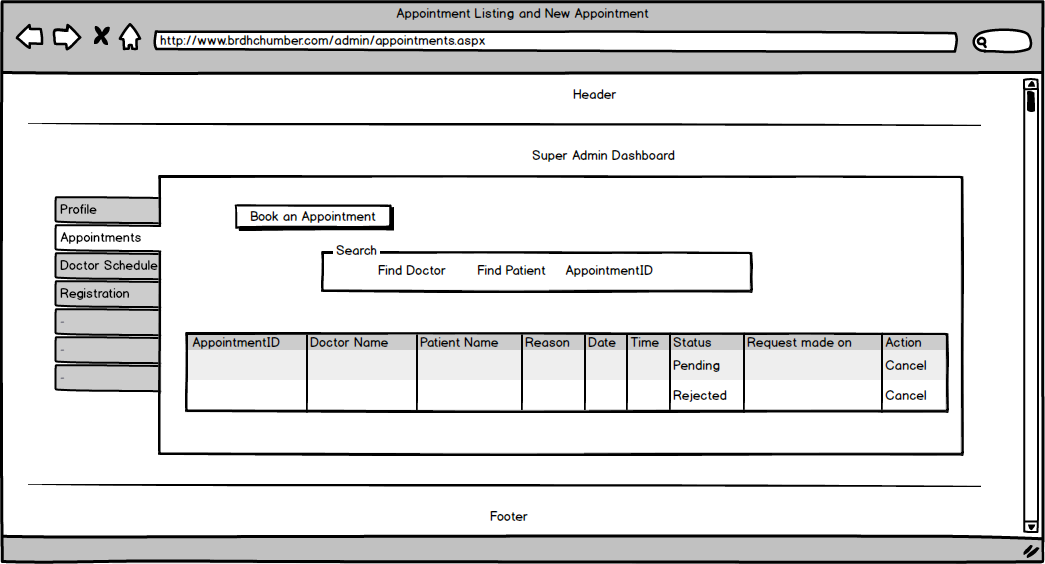
**Wireframes**

Here is the wireframe for booking an appointment by admin for the patients



The interface to book an appointment will be similar for the doctors. Here is the wireframe for doctors to book an appointment.



List view of all Appointments in admin’s account will be as in the following wireframe

Alerts

**Narrative Description**

Alerts have admin interface and public interface. A short message of the latest alert is shown on the screen and when visitor click on read more link button a separate page with complete information opens. Here users can see previous alerts also and can email these alerts to anyone.

Admin user can add, update, delete, publish, and delete. To update, publish and delete same form is used by admin. All the information is stored in database. Once the alert is added and published subscribers will get emails. To publish the alert on home page, admin will need to select publish no and if admin wants to hide any alert from visitors, he will need to select status removed.

The table schema used for alerts is like below:

brdhc\_HealthAlerts

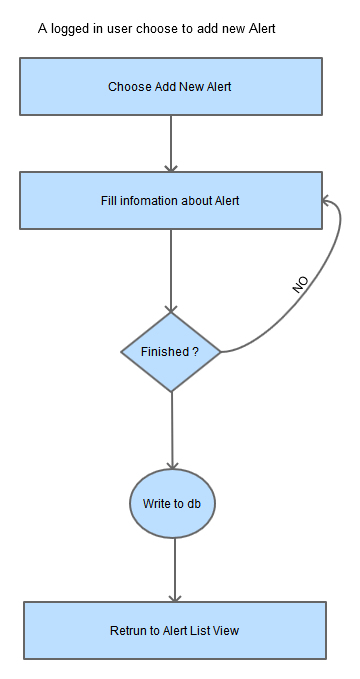
|  |  |
| --- | --- |
| Health Alert Id | Unique Identifier **(PK) (Auto Generated)** |
| Health Alert Title | Text 100 characters |
| Health Alert Description | Text MAX characters |
| Alert Date | Date |
| User Id | Unique Identifier |
| Status | True / False |
| Published | True / False |

brdhc\_AlertSubscribers

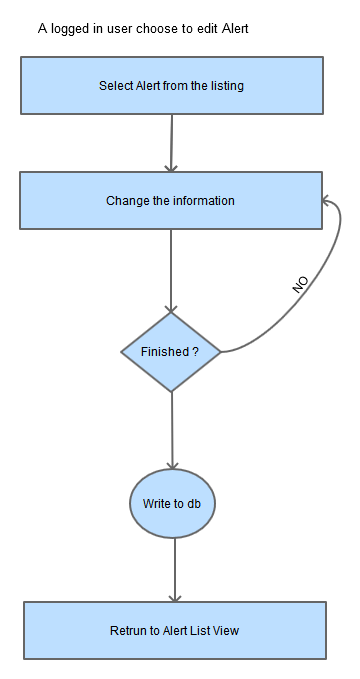
|  |  |
| --- | --- |
| User Id | Unique Identifier **(FK)** |

**The Dataflow Model**

After login into personal account admin user will see a link for alerts in the dashboard. Upon clicking this link, user will see alerts listing page and this page also contains the link to form where user can add new alerts. At this form user will provide the required information and all the information will be saved into database table. Following data flow model explains the process of adding new alerts in the database

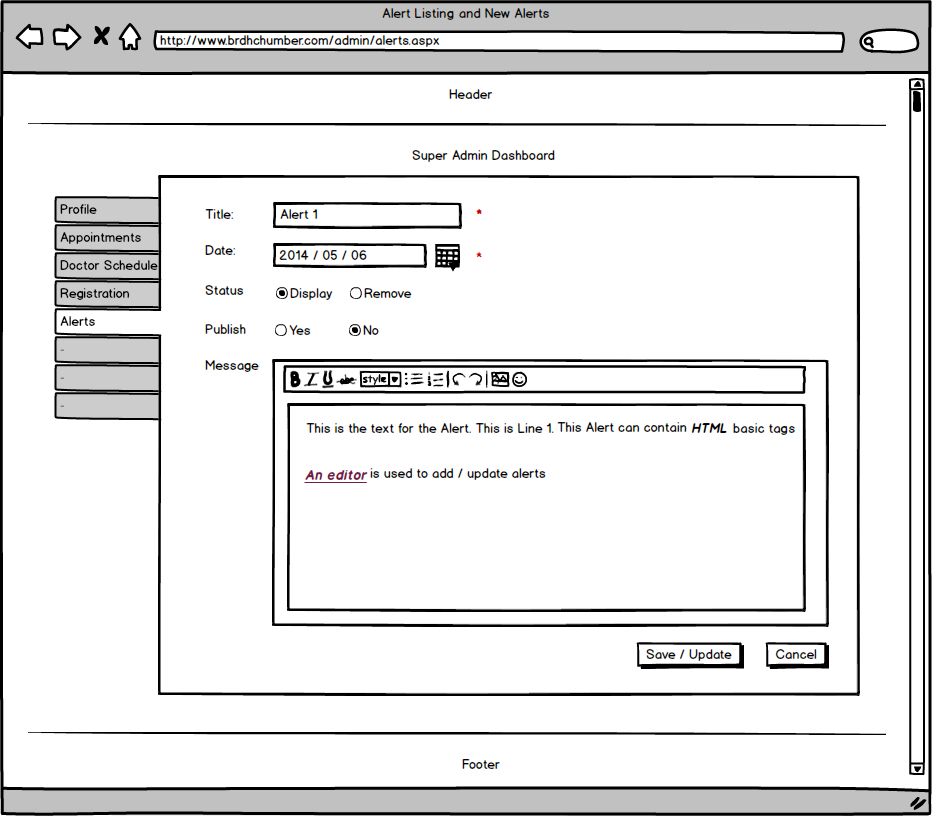


And admin can edit the information of alert like alert title, message, publish, or status. Once the user select the alert from list to update and click on edit button same form will display that is used to add new alerts. Following is the flow model for information updating.

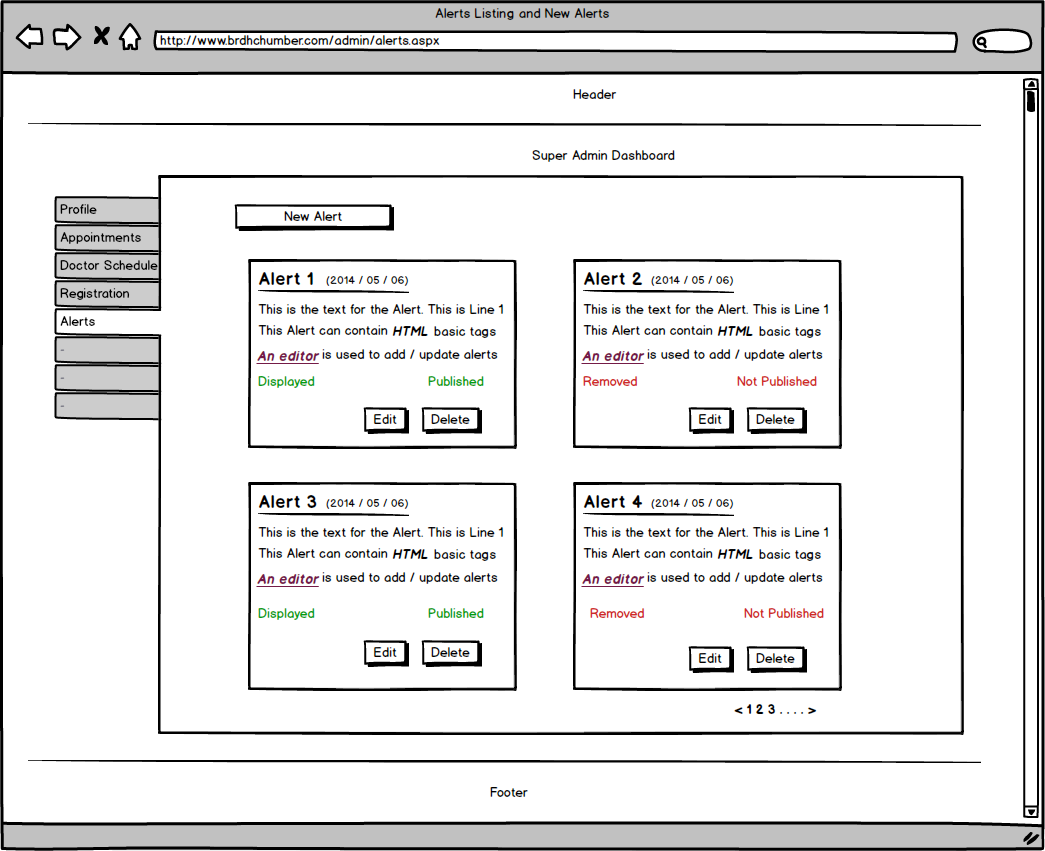


**Wireframes**

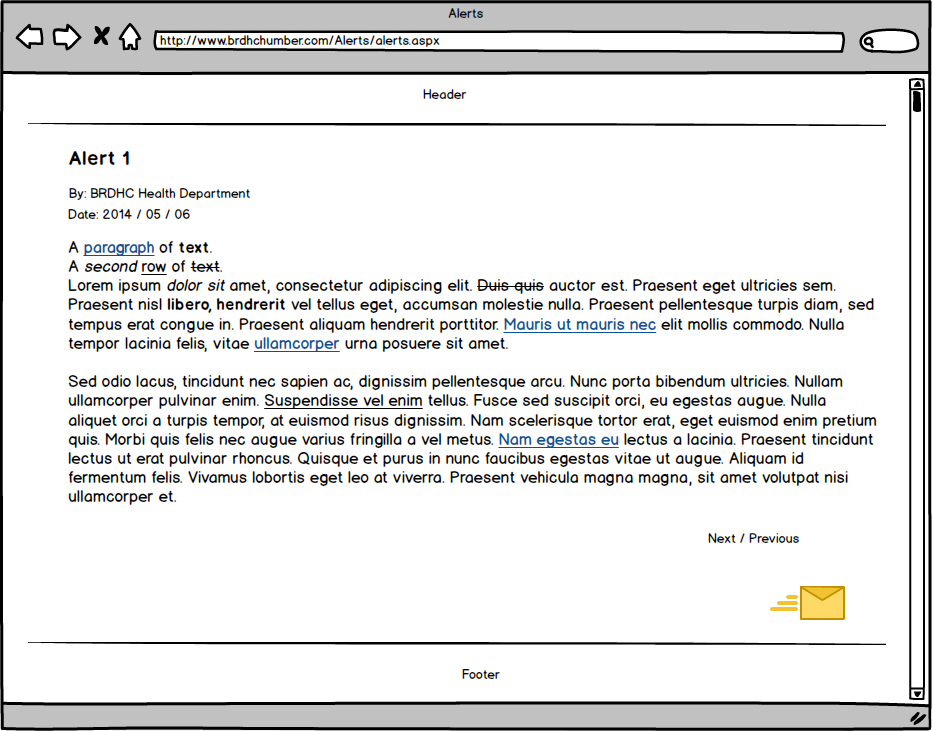
Interface to add new alerts or to update alerts looks like below:



Following is the wireframe to display all the alerts



And here is the wireframe that explains how the alerts page will look like when a visitor tries to see alert in detail.



Prescriptions

**Narrative Description**

Prescriptions are related to appointments. When patients meet doctors and doctors write prescriptions. The information of prescription is stored in system by doctors. This information contains appointment id, patient information, date, medicine details, repeat, medicine timings, and quantity to take at a time.

When prescription is saved in database, patients can view prescription details in their account; moreover, they can print schedule for medicines. Only doctors can add, update, and delete prescriptions.

Structure of tables is below:

brdhc\_PatientPrescriptions

|  |  |
| --- | --- |
| Prescription Id | Unique Identifier **(PK) Auto Generated** |
| Appointment Id | Unique Identifier **(FK)** |
| Repeat | Numeric |
| Prescription Date | Date |

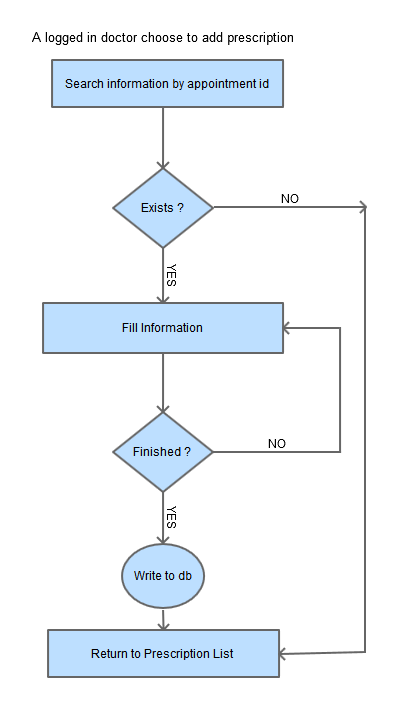
brdhc\_PrescriptionsDetails

|  |  |
| --- | --- |
| Prescription Id | Unique Identifier **(FK)** |
| Medicine | Text 100 characters |
| Timings | Text 100 characters |
| Days | Numeric |
| Quantity | Numeric |

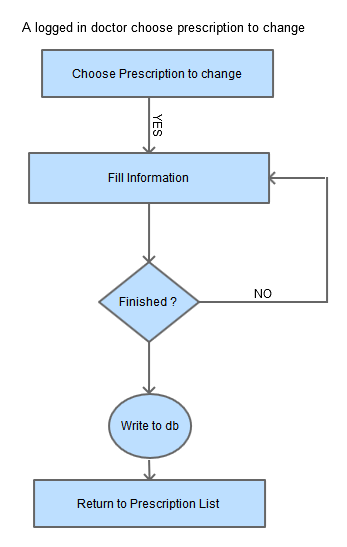
**The Dataflow Models**

To add prescriptions, doctors needs to be logged in. Doctor will see prescriptions link on dashboard. First it will display list of all the prescriptions. Doctors will need to click add new prescription button that in further will open a new form for prescription information. Here once doctor is done with filling all the information and click submit button, information will be saved into database.

Following data flow explains the process of adding new prescription into database.

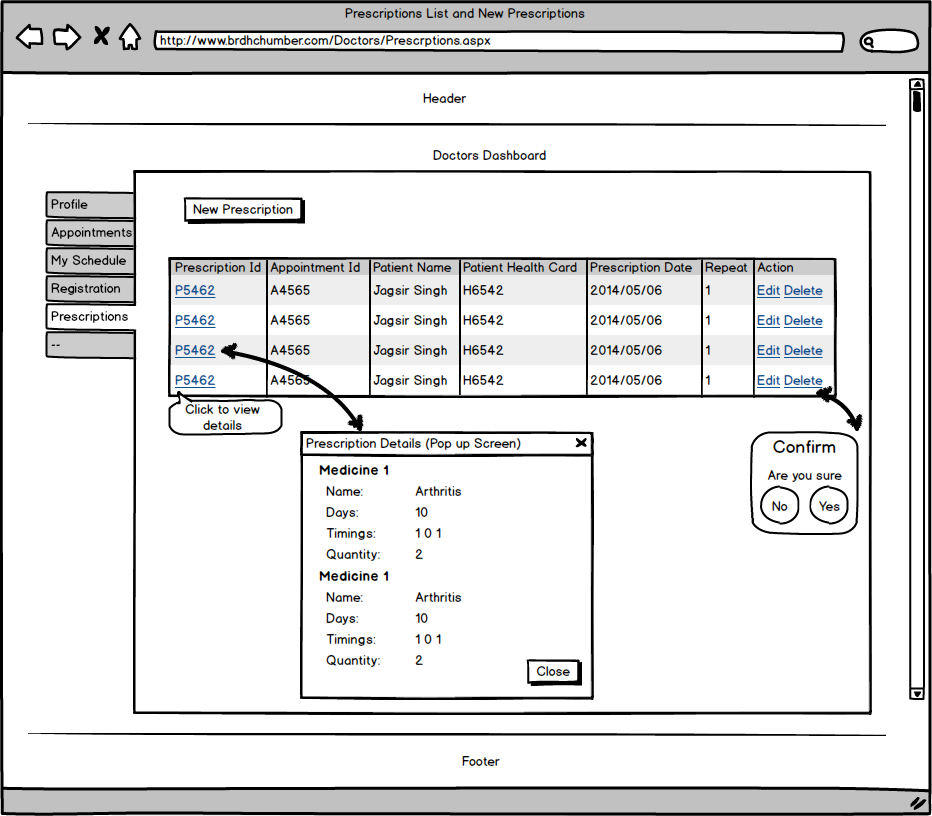


To update the prescription information, doctor will select the prescription from list and click on edit button. Same form which is used to add new prescription appears with populated fields. Now doctor can change information and click on update button to save the information in database. Here is the data flow chart for prescription update.

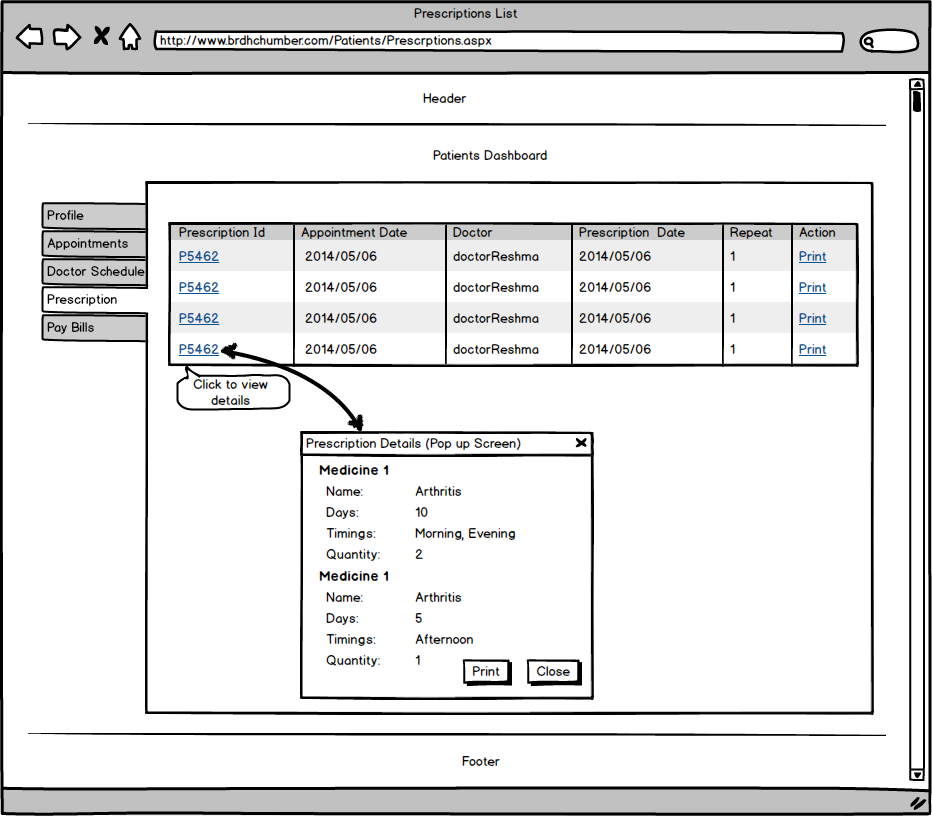


**Wireframes**

When doctors view list of prescriptions, it will look like below



Here is the wireframe for prescription in patients account



And here is the wireframe to add new prescription by doctors

