

HEALINT

Migraine App Technical Specification

Version 0.11

April 17, 2014

# Change History

|  |  |
| --- | --- |
| Version | Changes |
| 0.9 | * Updated HEAD /v1/authTokens/${id} in profile service to take 2 query parameters appId & userID * Updated GET /v1/accessCode/${code} in profile service to /v1/accessCode/${code} * Updated DELETE /v1/successCodes/${code} in profile service to DELETE /v1/accessCodes/${code} * Removed The optional parameter accessCode from POST /v1/users in profile service * Removed the JSON support for POST /v1/users in profile service * Added POST /v1/accessCodes/${code}/user in profile service to create a new user using an access code * Updated the overall architecture section to explicitly list supported formats and HTTP response codes |
| 0.10 | * Updated the “When” column for status codes 400 and 406. |
| 0.11 | * Update specifications of all the URIs of the event service. The userId is now passed as a query parameter instead of being part of the URI. |

# Overall Architecture

The diagram below shows a high level architecture. Each of the components below will be described in a separate section.

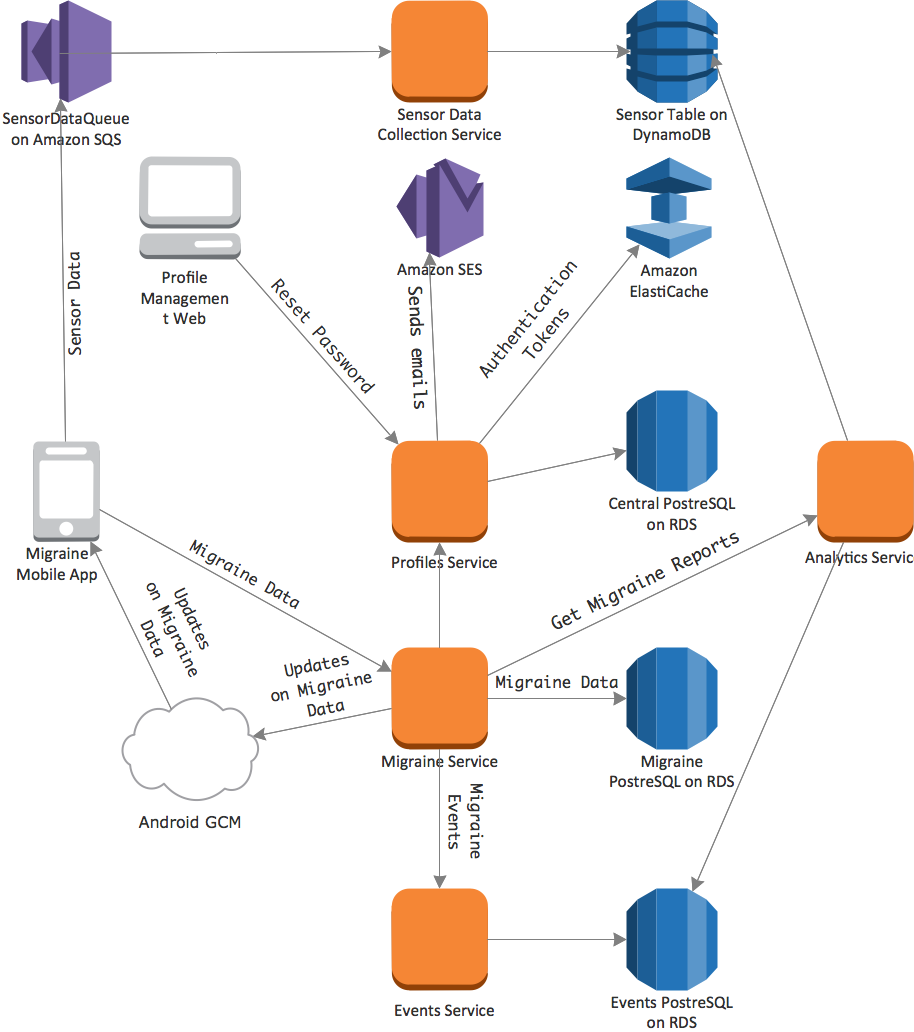


Figure - Migraine Mobile App Architecture

There are also 3 other clients: Doctors web, Admin web, and Profile Management web interfaces. The following 2 diagrams show architecture components that interaction with these clients.

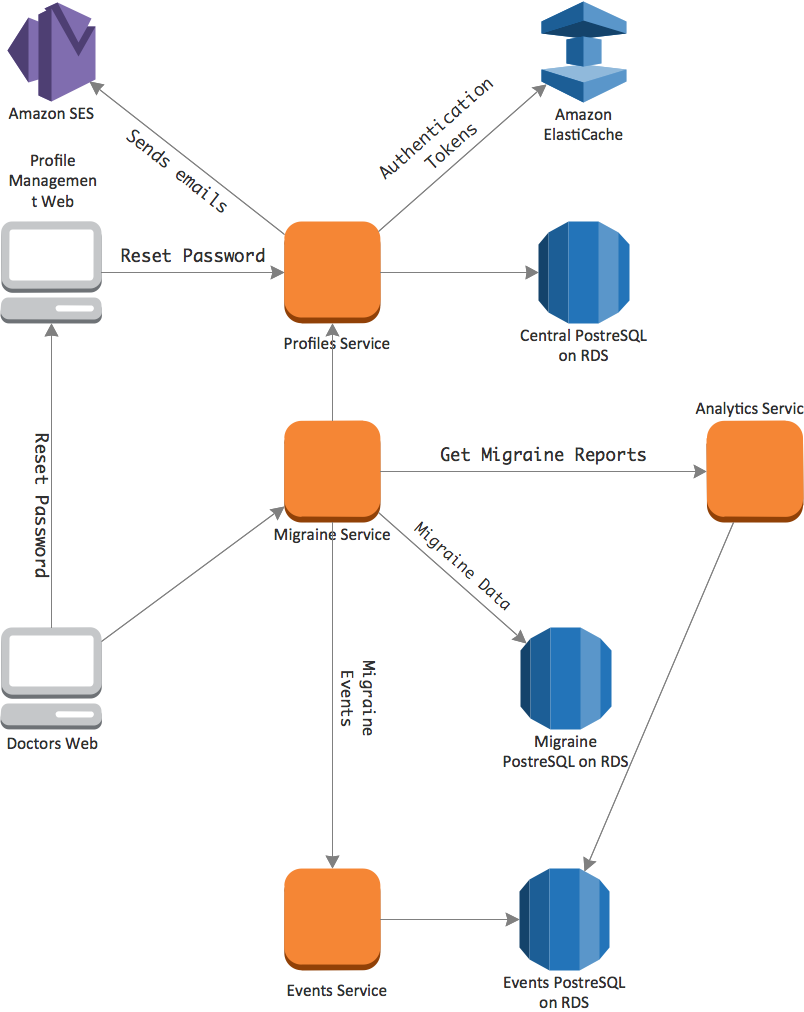


Figure - Doctors Web Architecture

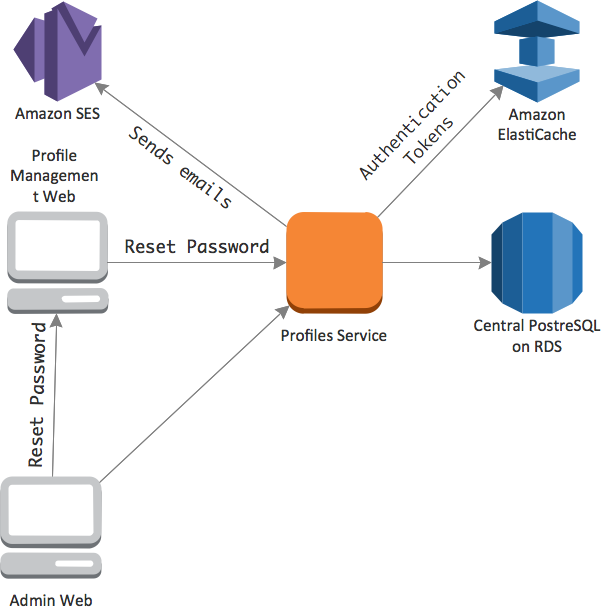


Figure - Admin Web Architecture

Backend REST services support only 2 types of formats HTML and JSON. Not all operations support the 2 formats, the table in each services specifies which format is supported for each operation. Clients that prefer using JSON format for communication must send the HTTP accept header and specify JSON as the first accepted MIME format.

Additionally, the backend services respond with different HTTP status codes depending on the outcome of processing requests. The following table lists the error HTTP status codes that could be returned.

|  |  |
| --- | --- |
| **HTTP Status Code** | **When** |
| 400 | If the request has an invalid syntax. |
| 401 | If the request requires the user to be authenticated but the request doesn’t contain the authentication header either is the session cookie or in the HTTP header |
| 403 | If the request requires the user to have a specific role that she doesn’t have |
| 404 | If the request is requesting a resource/entity that cannot be found |
| 405 | If the request’s HTTP method is not supported. Should be returned automatically by the play framework |
| 406 | If the requested response format based on the HTTP accept header is missing or not supported by this operation. For example, the operation only supports HTTP but the client is requesting JSON or vice versa |
| 408 | The service requires communication with an external service but that communication has timed out |
| 411 | If the length is not specified in the HTTP request. Should be returned automatically by the HTTP server |
| 414 | If the request URI is too long. Should be returned automatically by the HTTP server |
| 419 | If the authentication token provided in the request has expired |
| 422 | If the request cannot be processed due to semantic errors. For example, validation errors while creating or updating a resource/entity |

During the development, we will use git for source control management. We define a general overview of the branching in all our git repositories.

Some branches are restricted to persons to ensure quality

|  |  |  |
| --- | --- | --- |
| Branch | Responsibility of | Comment |
| nomovok | Ly / Chau (for iteration 2) | Developers create pull to submit code upwards |
| healint | Ali | After code has passed quality checks and unit tested |
| dev | Ali | After code has been deployed to development environment and checked by developers |
| test | Ali | After code has been tested on the Test environment and tested |
| master | Ali |  |

The checklist provided as a result of the PoC code review shall be used for code checks.

# Migraine Mobile App

This is the mobile application that patients use. Specification of user stories and GUI are described as cards in the project’s Trello board. The mobile app has a Sensors Data Collection Service as part of the application that collects sensor data over time and communicates with Amazon SQS over SSL. This background service will be developed by Healint and integrated in the application.

The following are general requirements for the app:

1. The App should support Android phones Gingerbread (V2.3 API 10) onwards.
2. The App should support screen sizes with resolutions between 480 X 800 mdpi and 1080 X 1920 xhdpi.
3. Google Analytics should be integrated in the app. Every action from users on a UI control button should be tracked by a custom event. As described here (<https://developers.google.com/analytics/devguides/collection/android/v3/events>) However, Strings should be retrieved from resources file and not written in the code as in the example. Reporting uncaught exceptions should be disabled as we plan to use HockeyApp for that.
4. Crash reporting will be done with HockeyApp
5. UserVoice should be integrated
6. SQLite database will be used for storing information. No local files will be stored separately without approval from Healint.
7. Upon logging in for the first time, the app retrieves all user data from the server and stores it locally in SQLite database. The app also stores the authentication token in a SharedPreferences so it doesn’t need to prompt the user to login again unless the server returned an authentication or authorization error as a response for one of the requests.
8. The server through Android GCM sends updates to the data. Upon receipt of such update, the app should update SQLite database and any application ui and update any ui if the app is running.
9. Updates to the data on the app are stored in the SQLite and sent the server if there is Internet connectivity. Otherwise, it will be sent to the server when the app starts again or when a GCM notification is received from the server.
10. In case of conflicts between local updates and server updates, the newest change overrides the older one. All records in the app will have a “Last Modified” timestamp to indicate when the update occurred.
11. The application should have the ant build capability functional
12. When the application is imported to ADT with eclipse, there should be no warnings. If the dev team thinks that they shouldn’t resolve the warning, they should communicate that with Healint.
13. All names used in the application should follow Java Naming Conventions.
14. Java Classes, variables and JSON objects representing objects in backend should have the same names as specified in backend specifications spreadsheets.
15. There should be NO String values stored written in Java Classes. All Strings should be in strings.xml and read the resources object
16. Mobile app should have a test suite that tests the app automatically either in isolation or with the service layer.
17. Charts will be drawn using ShinobiCharts
18. Follow the quality of code checklist provided during POC (see Quality of code checklist.pdf

During the signup or login process, the app checks the difference of time between the phone and the server. If the difference is more than 5 minutes time zone adjusted, a warning should be shown to the user asking him to correct the phone time. The difference that triggers this message should be a configuration parameter not hard coded.

# Sensor Data Collection Service

This service is developed by Healint to collect sensor data from devices.

# Migraine Service

This is the main Migraine service that handles requests from Migraine Mobile App, Doctors Web Interface. The following are general requirements for the app:

1. Play framework should be used.
2. All Communication to the Migraine service must be done over https, except the landing page of the doctor web interface.
3. The Service layer should have a test suite to test it in isolation or with integration with any or all of Profile service, Events Service, or Migraine Database
4. All resources in the app return either JSON response or HTML response depending on the HTTP Accept header.
5. The signature of the authentication token is verified and the user roles in the token are checked before processing requests that access resources that need specific roles as per the table below. More details about the authentication token are described in the Profile service section.
6. All PUT, POST, and DELETE operations (Except the login and signup) need extra security check that is done by calling validateToken operation on the Profile service after the regular security check that is described in point 5.

The following is the list of all REST resources in the app:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| URI | Method | Parameters | Required Roles | Doctors Web HTML Response | Mobile App JSON Response |
| /v1/time | GET | None | None | Not applicable | Current time in UTC as recorded by the server |
| /v1/login | GET | < Optional > nextPage: the uri of the page to send in the next POST request | None | Login page | Not applicable |
| /v1/login | POST | email and password as parameters in the post body  < Optional > nextPage: the uri of the page to redirect to after login | None | Authentication token as a Session cookie and HTTP redirect response to /v1/patients and passing doctorId if the user has the MIGRAINE\_DOCTOR Role. Otherwise, return authorization error response code. | Authentication token and signature |
| /v1/login/reset | GET | None | None | The password reset page | None |
| /v1/login/reset | POST | Email of the user to reset the password for | NONE | Password Reset Success page or Reset page again with error message | Confirmation of success or error |
| /v1/patients | GET | doctorId | MIGRAINE\_DOCTOR | List of patients managed by this doctor | Not applicable |
| /v1/patients | POST | New patient information and Access Code to apply | NONE | Not applicable | New patient details |
| /v1/patients/${id} | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Patient profile. |
| /v1/patients/${id} | PUT | Update patient information | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id} | DELETE | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of deleting patient profile. Id of the currently logged in patient must match the id in the uri |
| /v1/patients/${id}/migraines | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | None | Summary of journal entry for the user |
| /v1/patients/${id}/migraines | POST | New event information | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/migraines/${id} | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | None | Details of migraine event |
| /v1/patients/${id}/migraines/${id} | PUT | Update event details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/migraines/${id} | DELETE | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/triggers | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | List of triggers. Includes static and personal triggers |
| /v1/patients/${id}/triggers | POST | New trigger details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/reliefActions | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | List of relief actions. Includes static and personal relief actions |
| /v1/patients/${id}/reliefActions | POST | New relief action details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/reliefMedications | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | List of relief medications. Includes static and personal relief medications |
| /v1/patients/${id}/reliefMedications | POST | New relief medication details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/affectedActivities | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | List of affected activities. Includes static and personal affected activities |
| /v1/patients/${id}/affectedActivities | POST | New affected activity details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/symptoms | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | List of symptoms. Includes static and personal symptoms |
| /v1/patients/${id}/symptoms | POST | New affected activity details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/auras | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | List of auras. Includes static and personal symptoms |
| /v1/patients/${id}/auras | POST | New affected activity details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/locations | POST | New location details | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/locations | GET | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | List of locations. |
| /v1/patients/${id}/device | POST | Details of device and GCM registration | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/patients/${id}/device | DELETE | None | MIGRAINE\_PATIENT and Id of currently logged patient must match the id in the uri | Not applicable | Confirmation of success |
| /v1/doctors/${docId}/${patientId} | POST | None | MIGRAINE\_DOCTOR |  |  |
| /v1/accessCodes/${code} | GET | Access Code | None | Not applicable | Access Code details |

# Events Service

Communications with the events service is only done on HTTPS with SSL mutual authentication enabled (Requires using Apache httpd as a reverse proxy for play server). User authentication and authorization for event requests is the responsibility of the calling service not the events service.

This service is only accessible by Healint servers and doesn’t accept any connections from public Internet.

|  |  |  |  |
| --- | --- | --- | --- |
| URI | Method | Parameters | Mobile App JSON Response |
| /v1/events/ | GET | <Required> userId: the id of the user  < Optional > startId: parameter specifying an event id to start from.  < Optional > startTime: optional parameter specifying the time of the event to start from.  Note: The startId and startEvent are exclusive. i.e. The event with the exact startId and startTime will not be return in the response.  < Optional > limit: the maximum number of events to return  < Optional > attributes: the name of event attributes to return | All the user’s events regardless of the event type in descending order by time  < Optional > endId: The id of the last event returned if more results are available  < Optional > endTime: The time of the last event return if more results are available  To fetch next set of items, caller should call the operation again with startId = endId from response and startTime = endTiem from response |
| /v1/events/${eventType} | GET | < Required > eventType:  < Optional > startId: parameter specifying an event id to start from.  < Optional > startTime: optional parameter specifying the time of the event to start from.  Note: The startId and startEvent are exclusive. i.e. The event with the exact startId and startTime will not be return in the response.  < Optional > limit: the maximum number of events to return  < Optional > attributes: the name of event attributes to return | The user’s events that are of the specified type in descending order by time  < Optional > endId : The id of the last event returned if more results are available  < Optional > endTime: The time of the last event return if more results are available  To fetch next set of items, caller should call the operation again with startId = endId from response and startTime = endTiem from response |
| /v1/events/${time} | GET | < Required > userId: The id of the user  < Optional > attributes: the names of attributes to return | The details of the event specified |
| /v1/events | POST | Details of a new event | Confirmation of success or an error of the app has an event recorded for the specified user and application in the same millisecond |
| /v1/events/${time} | PUT | Update details of an event | Confirmation of success. It is not possible to update the event time. If this is needed, the app should delete the old event and create a new one. |
| /v1/events/${time} | DELETE | < Required > userId: the id of the user | Confirmation of success |

# Profile service

This service is responsible for Authentication, Authorization, and Auditing. Currently, only Authentication and Authorization will be implemented. Communication with this service is only done over HTTPS. Only one authentication token is allowed per user per application. Authentication tokens do not have a time limitation and are stored in Amazon ElastiCache. when the user login to an application, the new authentication token replaces the old one (if it exists) in ElastiCache. Authentication token ids are generated using the Java’s UUID class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| URI | Method | Parameters | Admin Web HTML Response | Mobile App JSON Response |
| /v1/login | GET | < Optional > nextPage the nextPage parameter to pass to the next call of POST /v1/login  < Optional > Error Message to show | Login page | Not applicable |
| /v1/ login | POST | email and password, and application id as parameters in the post body  < Optional > nextPage: the uri of the page to redirect to after login | Authentication token as a Session cookie and HTTP redirect response to /v1/users | Authentication token and signature |
| /v1/login/reset | POST | Email to reset | Not applicable | Confirmation of success |
| /v1/login/${resetToken} | GET |  | Password reset page | Not applicable |
| /v1/login/${resetToken} | POST | New password | Password reset success page | Not applicable |
| /v1/tokens/${id} | GET | None | Not applicable | Authentication token details or error if the token is not valid. |
| /v1/users | GET | < Optional > startId: specified the id of the first user. This parameter is exclusive. i.e. the user with this specific parameter will not be returned in the response  <Optional > limit: maximum number of users to return  <Optional > sort: attribute to sort by  < Optional > order: asc or desc  < Optional > filter attributes  < Optional > error: an error message to display  < Optional > success: a success message to display  < Optional > notice: a notice message to display | User management page listing users returned from the query | List of users returned from the query |
| /v1/users/${id} | GET | None | User information page | User details |
| /v1/users/new | GET | None | New user page | Not applicable |
| /v1/users | POST | New user details | Redirects to /v1/users with success message | Not applicable |
| /v1/users/${id} | PUT | Updated user details | Redirects to /v1/users with success message | Confirmation of success |
| /v1/users/${id} | DELETE | None | Redirects to /v1/users with success message | Confirmation of success |
| /v1/roles | GET | < Optional > success: a success message to display  < Optional > error: an error message to display  < Optional> notice: a notice message to display | Page displaying all roles | List of roles |
| /v1/roles/new | GET | None | Page to create a new role | Not applicable |
| /v1/roles/${id} | GET | None | Page to view role details | Role details |
| /v1/roles | POST | New role details | Redirects to /v1/roles with success message | Confirmation of success and the id of the new role |
| /v1/roles/${id} | PUT | None | Redirects to /v1/roles with success message | Confirmation of success |
| /v1/roles/${id} | DELETE | None | Redirects to /v1/roles with success message | Confirmation of success |
| /v1/countries | GET | < Optional > success: success message to display  < Optional > error: error message to display  < Optional > notice: a notice message to display | Page displaying list of countries | List of countries |
| /v1/countries/${id} | GET | None | Page displaying country details | Country details |
| /v1/countries/new | GET | None | Page to create a new country | Not applicable |
| /v1/countries | POST | New country details | Redirects to /v1/countries with a success message | Not applicable |
| /v1/countries/${id} | PUT | Updated country details | Redirects to /v1/countries with a success message | Not applicable |
| /v1/countries/${id} | DELETE | None | Redirects to /v1/countries with a success message | Not applicable |
| /v1/accessCodes | GET | < Optional > startId: specified the id of the first access code. This parameter is exclusive. i.e. the access code with this specific parameter will not be returned in the response  <Optional > limit: maximum number of access codes to return  <Optional > sort: attribute to sort by  < Optional > order: asc or desc  < Optional > filter attributes  < Optional > error: an error message to display  < Optional > success: a success message to display  < Optional > notice: a notice message to display  < Optional > download: link to a file to automatically start downloading after the page is displayed | Page listing access codes | Not applicable |
| /v1/accessCode/new | GET | None | Page to generate access codes | Not applicable |
| /v1/accessCodes | POST | Access code details and the number of access codes to generate | Redirects to /v1/accessCodes with success message and a download parameter to a PDF file | Not applicable |
| /v1/accessCodes/${code} | GET | None | Access code details page | Access Code details |
| /v1/accessCodes/${code} | PUT | Updated accessCode details | Redirects to /v1/accessCodes with success message | Not applicable |
| /v1/accessCodes/${code} | DELETE | None | Redirects to /v1/accessCodes with success message | Not applicable |
| /v1/accessCodes/${code}/user | POST | New User Details | Not applicable | New patient details |
| /v1/hospitals | GET | < Optional > startId: specified the id of the first hospital. This parameter is exclusive. i.e. the hospital with this specific parameter will not be returned in the response  <Optional > limit: maximum number of hospitals to return  <Optional > sort: attribute to sort by  < Optional > order: asc or desc  < Optional > filter attributes  < Optional > error: an error message to display  < Optional > success: a success message to display  < Optional > notice: a notice message to display | Page displaying list of hospitals | List of hospitals |
| /v1/hospitals/${id} | GET | None | Hospital details page | Hospital details |
| /v1/hospitals/new | GET | None | New hospital page | Not applicable |
| /v1/hospitals | POST | New hospital details | Redirect to /v1/hospitals with success message | Not applicable |
| /v1/hospitals/${id} | PUT | Update hospitals details | Redirects to /v1/hospitals with success message | Not applicable |
| /v1/hospitals/${id} | DELETE | None | Redirects to /v1/hospitals with success message | Not applicable |
| /v1/authTokens/${id} | HEAD | AppId and UserId as Query Parameters | Not applicable | Status code 204 if the authentication token is still valid in the Elasticache or 404 if it is not. |

The authentication token that is returned or validated by some of the previously mentioned operations consists of the following:

1. tokenId
2. userId
3. appId
4. userRoles and their expiry dates if any
5. Timestamp at which the token was generated

Only users with ADMIN role are allowed to create new users with specific roles, create new roles, and generate access codes, add/edit/delete countries and hospitals. A new user can signup and specify an access code to grant himself/herself a role automatically.

Items in all drop down boxes in the wireframe should be sorted alphabetically.

Drop boxes specifying a doctor are populated with users who have the role DOCTOR.

It is not possible to delete or rename the roles ADMIN and DOCTOR.