**General System Rules**

* Use URLs like “xxx.peerconnect.io” (taken, find a replacement domain name)
* Add Google Analytics tracking code to the site
  + Each program will have its own subdomain and GA tracking code
* All API calls will receive a static “tenant ID” parameter passed by the front-end
* **Program configuration options**
  + Tenant ID
  + GA Tracking Code (Store in Front-end Configuration)
  + Session Duration in minutes
  + Limit on the number of sessions a participant may participate in
  + Limit on the number of sessions a captain may participate in (30 for K.E.)
  + Limit on the number of times a participant may speak about a particular topic
  + Program has a start and an end date - ensure these are editable
    - Captain availability must be scheduled within the program start/end dates
    - Participants are able to schedule sessions within Captain availability
  + Support Email Address
  + List of Support Topics
* **Topics (Needs Review)**
  + Topics are per Tenant
  + Many-to-many relationship with Captains
  + Many-to-many relationship with Resources
  + Id - Int32 (PK)
  + Name  **-** need data type and data length
* **Resources**
  + Resources are per Tenant
  + Each resource is associated with zero or more Folder
  + Id - Int32 (PK)
  + Name  **-** need data type and data length
  + Type - (PDF, Video, etc.)
  + Url **-** need data type and data length
    - External and internal links
* **Roles**
  + Roles are per Tenant
  + Options:
    - MD/DO
    - DMD
    - DDS
    - PharmD/RPh
    - PA
    - CRNA/NP/APRN
    - RN
    - Other (Open text)
* **Segment**
  + Segments are per Tenant
  + Id - Int32 (PK)
  + Name  **-** need data type and data length
* **Specialities**
  + Specialities are per Tenant
  + Id - Int32 (PK)
  + Name  **-** need data type and data length
* **Timezones**
  + Timezone Label - Text, need data length
  + Timezone UTC Offset - ([+/-]00:00)
  + Timezone UTC DST Offset - ([+/-]00:00)
  + (AC to provide list of timezones)
* In notifications, instead of linking users to specific session IDs for each meeting, instead link to the “My Next Meeting” page.

**Forgot Password**

* Use existing Keystone password reset functionality
* Users will be able to access a page where they enter their email address, and if a match if found, email them a link to reset their password.
* The link in the email will contain a UID for the user that will authorize them to change their password.
* The link in the email will expire after a set period of time (12 hours).
* Use existing API calls where available

**Unsubscribe**

* Users click an “unsubscribe” link from an email, they will be taken to a confirmation page, and will be able to confirm that they no longer want to receive communications (email or text) for the associated program.
* Unsubscribed users will still be able to log in.
* Use existing API calls where available
* If someone unsubscribes, but wants to re-enroll, manually change their subscription flag back to active
  + If the user tries to re-enroll through the enrollment page, display an error that a record already exists for the email address

**Administrator Enrollment**

* Arches to provide endpoint to ensure that the Captain data can be uploaded to Keystone (Create/Update/Delete/Get by Id/Get by Email)
* Entrada to provide Administrator data to Steeltype
  + **First name\*** - need data type and data length
  + **Last name\*** - need data type and data length
  + **Email address\*** - need data type and data length
    - Going to be used for login
  + **Password\* -** need data type and data length
  + **Tenant Id\*** - need data type and data length
    - Hardcoded for each program’s front-end
* SteelType to upload Administrator profile data to the API endpoint

**Captain Acquisition**

* Arches to provide endpoint to ensure that the Captain data can be uploaded to Keystone (Create/Update/Delete/Get by Id/Get by Email)
* Arches to provide endpoint to ensure that Captain logins can be authenticated
* Entrada to provide Captain data to Steeltype
  + **First name\*** - need data type and data length
  + **Last name\*** - need data type and data length
  + **Title** - need data type and data length
  + **Email address\*** - need data type and data length
    - Going to be used for login
  + **Password\* -** need data type and data length
  + **Mobile number\*** - need data type and data length
  + **Affiliation**  - (string) need data type and data length
  + **Timezone\*** - need data type and data length
    - The list of options will be provided by the API
  + **Address Line 1 -** need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it

**Address Line 2 -** need data type and data length

* + - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **City** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **State** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **Zip Code** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **Segment** - need data type and data length
    - This will be captured as a multiple choice question on the front-end
    - The list of options will be provided by the API
  + **Profile image** - pass as URL - need data type and data length
  + **Biography** - need data type and data length
  + **Topics** (See “General System Rules”)
    - Specify which topics Captain can speak on
    - The list of options will be provided by the API
  + **Tenant Id\*** - need data type and data length
    - Hardcoded for each program’s front-end
  + **Specialty** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
    - The list of options will be provided by the API
  + **Role** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
    - The list of options will be provided by the API
  + **Opt-in** - (boolean)
    - Not sent by front-end, assumed to always be “true”
  + **Terms & Conditions** - (boolean)
    - Not sent by front-end, assumed to always be “true”
  + **MediaSourceId** (Hidden Field, Default to Null)
    - This is used to identify the referral source (if known)
    - Note: Client plans to send solicitation emails to prospects in 2019
* SteelType to upload Captain profile data to the API endpoint

**Participant Acquisition / Enrollment**

* Arches to provide endpoint to ensure that the Participant data can be captured from signup form (Create/Update/Delete/Get by Id/Get by Email)
* Arches to provide endpoint to ensure that Participant logins can be authenticated
* Data fields captured within signup form
* Entrada to determine how to breakdown the enrollment into steps
  + **First name\*** - need data type and data length
  + **Last name\*** - need data type and data length
  + **Title** - need data type and data length
  + **Specialty** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
    - The list of options will be provided by the API
  + **Email address\*** - need data type and data length
    - Going to be used for login
  + **Password\* -** need data type and data length
  + **Confirm Password\*** - (Front End Only)
  + **Mobile number\*** - need data type and data length
  + **Affiliation**  - (string) need data type and data length
  + **Role** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
    - The list of options will be provided by the API
  + **Timezone\*** - need data type and data length
    - The list of options will be provided by the API
  + **Address Line 1 -** need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it

**Address Line 2 -** need data type and data length

* + - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **City** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **State** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **Zip Code** - need data type and data length
    - Not visible on front-end form for Knowledge Exchange, however back-end should support it
  + **Segment** - need data type and data length
    - This will be captured as a multiple choice question on the front-end
    - The list of options will be provided by the API
  + **Tenant Id\*** - need data type and data length
    - Hardcoded for each program’s front-end
  + **Opt-in** - (boolean)
    - Not sent by front-end, assumed to always be “true”
  + **Terms & Conditions** - (boolean)
    - Not sent by front-end, assumed to always be “true”
  + **MediaSourceId** (Hidden Field, Default to Null)
    - This is used to identify the referral source (if known)
    - Note: Client plans to send solicitation emails to prospects in 2019
* Upon enrollment, redirect the participant to the session signup
* On the front-end, display help contact information
* On the front-end, display a progress bar indicating profile completion percentage
* We will not be adding in profile segmentation information in this version
* If an email address already is enrollment in the system, do not let them enroll again.
* Need an API call to do real-time checks on whether a user’s email address already exists in the system.

**Captains Availability**

* Captains are able to define availability timeslots (the frontend will display the timeslots in the Captain’s local time)
  + A timeslot is a start date/time (UTC) and a duration
  + Timeslots may only be defined within the program start/end dates
* Captains may edit and delete timeslots
* Frontend will retrieve the Captain’s timezone information from their profile
* Need an API endpoint which is able to batch create timeslots, edit a single timeslot, and batch delete timeslots (all start times will be UTC)
* Need an API endpoint which is able to retrieve all timeslots for a Captain within a given start date and end date (start date and end date will be UTC)

**Upcoming Sessions Page (Participant)**

* This will be the default landing page after enrollment or login
* Display a notice that advises participants to schedule their session at least a day in advance
* This page will display a notice if the participant has spoken about all available topics
* This page will display a list of all upcoming sessions involving the participant
  + Display all session times/dates in participant’s timezone, ordered by soonest first
* Need an API endpoint which is passed a participant ID, and returns all of the upcoming sessions for that participant (order results by soonest first)
* Need an API endpoint which retrieves an availability timeslot by ID (datetimes in UTC)
* Display a link to the topic selection screen for participants to schedule new sessions
* Display links next to each session to:
  + Allow the participant to join the session
  + Allow the participant to add the session to their calendar (no API functionality required)
  + Allow the participant to cancel the session
    - Need an API endpoint to delete session by session ID and availability ID as a participant, and make the corresponding availability slot available for use again
    - Triggers email notification to the captain
* Display a list of all completed sessions involving the participant (not used in K.E.)
  + Display all session times/dates in participant’s timezone, ordered by most recent first
  + Need an API endpoint which is passed a participant ID, and returns all of the completed sessions for that captain (order results by most recent first)

**Upcoming Sessions Page (Captain)**

* This will be the default landing page after login
* This page will display a list of all upcoming sessions involving the captain
  + Display all session times/dates in captain’s timezone, ordered by soonest first
* Need an API endpoint which is passed a captain ID, and returns all of the upcoming sessions for that captain (order results by soonest first)
* Need an API endpoint which retrieves an availability timeslot by ID (datetimes in UTC)
* Display links next to each session to:
  + Allow the captain to join the session
  + Allow the captain to add the session to their calendar (no API functionality required)
  + Allow the captain to cancel the session
    - Need an API endpoint to delete session by session ID and availability ID as a captain, and make the corresponding availability slot available for use again
    - Triggers email notification to the participant asking them to reschedule
    - Triggers email notification to the admin team is captain has “excessive cancellations”
* Display a list of all completed sessions involving the captain (not used in K.E.)
  + Display all session times/dates in captain’s timezone, ordered by most recent first
  + Need an API endpoint which is passed a captain ID, and returns all of the completed sessions for that captain (order results by most recent first)

**Topic selection, Captain Matching & Session Scheduling**

* When a surgeon goes in to schedule a session, she will be taken through the following set of front-end screens - Topic selection, Captain matching, Session scheduling
* Topic selection page will display a list of all the topics that a participant hasn’t already scheduled a session to speak to a captain about
  + Need an API endpoint which returns all the ordered topics (id, ordinal, label) that a participant hasn’t already scheduled a session to speak to a captain about
  + Need an API endpoint for developers to create/edit/delete topics
  + Participant will select a single topic, and the selected topic ID will be passed to the Captain matching page
* Captain matching page will display a list of all captains who are available to speak and eligible to speak about the selected topic ID
  + Need an API endpoint which returns all captains who are associated with a topic ID, and who are available to speak in the future
    - Availability is limited to X number of sessions allowed per month
    - The Captain must have an unused availability timeslot in the future
  + Participant will select a single Captain, and the selected Captain ID and Topic ID will be passed to the Session scheduling page
* Session scheduling page will display a list of all the unused availability timeslots for a particular Captain and topic combination
  + Frontend will retrieve the Captain’s timezone information from their profile
  + Frontend will retrieve the Participant's timezone information from their profile
  + Need an API endpoint which is able to retrieve all unused timeslots for a Captain for a particular topic ID
    - Only show timeslots in the future
    - Endpoint should take optional parameters of a given start date and end date (start date and end date will be UTC)
  + Participant will select a single timeslot
    - Need an API endpoint to create a session with a participant ID, captain ID, topic ID, timeslot ID, tenant ID
    - Triggers an email notification to the participant about the session
    - Triggers an email notification ot the captain about the session
    - Display a session confirmation page with a button that takes them to their “upcoming sessions” page

**Notifications - Captains, Participants, Admin**

* Urls in text messages will be shortened using bit.ly
* User clicks on links in notifications will not be tracked
* Please see the linked document for details: <https://docs.google.com/spreadsheets/d/1p2JZfjMHwIF4ITqYIidIipLusA-h9NAVT3lSma0m95c/edit#gid=709912565>

**Resource Library**

* Resources will consist of:
  + Id (PK)
  + Resource Name - need data type and data length
  + Resource Description - (varchar, ~4000 chars)
  + Resource Absolute Url - (varchar, ~2000 chars)
  + Thumbnail Image Url - (varchar, ~2000 chars)
  + Category Id (referenced Id, optional)
  + Content Type (enum?) - need data type and data length
  + Folder Id (referenced Id, optional)
  + Tenant Id (referenced Id)
* Folders will consist of:
  + Id (PK)
  + Folder Name - need data type and data length
  + Is Restricted (bool)
  + Topic Id (referenced Id, optional)
  + Featured Resource (referenced Id, optional)
  + Tenant Id (referenced Id)
* Rating will consist of:
  + Id (PK)
  + User Id (referenced Id)
  + Score - (integer?)
* Resources may be grouped into folders
* Each folder may be associated with a single topic
* Each folder may be “locked” for participants, meaning that resources within it are not accessible until they have completed a session on the associated topic.
* All folders are “unlocked” for captains
* Resource materials will be preloaded during program setup
* Need API calls to:
  + Request a list of resources in a given folder (or with no assigned folder)
  + Request a list of folders
  + Request a single folder by Id
  + Request a single resource by Id
  + Request the list of topics completed by a participant
  + Record each user’s attempt to access a resource
  + Record a user’s rating for a resource

**Waiting Room**

* Before joining a session, users are directed to this page.
* Displays the offline camera feed, and allows users to select their audio and video devices.
* When a participant lands on the page, display the ISI in a scrollable pop-up.
  + Display a “I understand” checkbox after the user scrolls to the bottom of the ISI which must be checked before they are able to proceed.
* Display a button that allows users to indicate they are ready to start the session.
* Display a button to Captains that allows them to send a reminder notification to the participant.

**Sessions**

* When a user joins a session, the frontend will call the API to let the backend know that a user has joined the session. The backend will then log that the user has “shown up” to the session for reporting purposes.
* After a participant completes a session, present them with a 1-5 star rating question. The rating they select will be saved to the backend via an API call.
* Knowledge Exchange sessions will all be video and audio together
* Calls will be recorded once both parties join, the recorded sessions will be exportable as video files
* Display links to the ISI and PI in the footer of the page, open these as new tabs
* Display the complaints phone number
* Controls:
  + Captain:
    - Send MIRF notification button (Calls the API to request that the “Off label request” notification be sent to the participant)
    - Exit session button
  + Participant:
    - Exit session button (prompts user for call rating)
* When the Captain clicks the “exit session” buttons, information about the session, including a reference to the Twilio recording and the session length, are sent to the API. This will mark completion of the session.
* Need a “complete session” call, which should only be called once for a particular session, and should reject any further calls for that session

**My Next Meeting**

* Users will be transparently redirected to their next upcoming session
* This page will use the same API calls as the “Upcoming Sessions” page

**Session Archives**

* Accessible only to Administrators
* Displays a list of previous completed sessions
  + Include:
    - Captain First Name + Last Name
    - Participant First Name + Last Name
    - Session Date (Use Captain Timezone)
    - Session Duration
    - Topic
  + Sort by:
    - Session Date
    - Topic
    - Captain Name
* Each session will have a button next to it to allow the user to download the recorded video file
* NOTE: Conversion of the recording into a downloadable MP4 will happen automatically. (Steeltype to investigate)

**FAQs**

* Display a static list of FAQ entries
* Display a “Need Help” button that sends an email notification to Entrada informing them that someone needs help.
  + Users enter in an contact phone number that will be included in the notification body.
  + Users select a support topic to discuss from the list of support topics
  + Need an API call for this.