HIDOCTOR EDETAILING CLIENT APPLICATION

Specification Document v6.0

vijay prasad
vijay@swaas.net

Contents

HiDoctor EDetailing Client Application – Android Version

Purpose

Definitions, Acronyms and Observations

- HD –HiDoctor
- HD DB – HiDoctor Custom SQL Database
- Device – Android / IPad / Windows 8 tablet devices.
- Digital assets – Video files, Audio files, Microsoft office suite files including excel, word, power point files
- Digital assets source – Refers to a source digital media asset system that is responsible for uploading of content and serving content either offline or online
- Client System – Refers to the proposed application that is being built

Coding – Standards, Auditing & Exception handling

1. All messages, labels and alert strings need to come from a settings file / resource file / configuration file. This will allow us to make regional changes / language changes to the application as and when required by releasing an update pack or a resource file pack.
2. Date controls wherever applicable (both display and input) need to read the format from the configuration, the Indian date format must not be hard coded in the system anywhere
3. Exceptions need to clearly logged in a separate file with detailed stack trace, screen name, functionality that was executed (method name usually), relevant input strings that was used during the exception time.
4. System must handle any errors gracefully and request the user to retry the failed operation wherever applicable instead of crashing
5. During AUTO SYNC download activity, system must constantly monitor the download progress and in any case the internet drops off, system must have the ability to restart the download where it left off automatically. This includes for master data / digital asset downloads.
6. System must allow multiple downloads to be queued instead of forcing the user to download one at a time. The download progress must be clearly shown to the user in terms of a progress bar.

Configuration Items

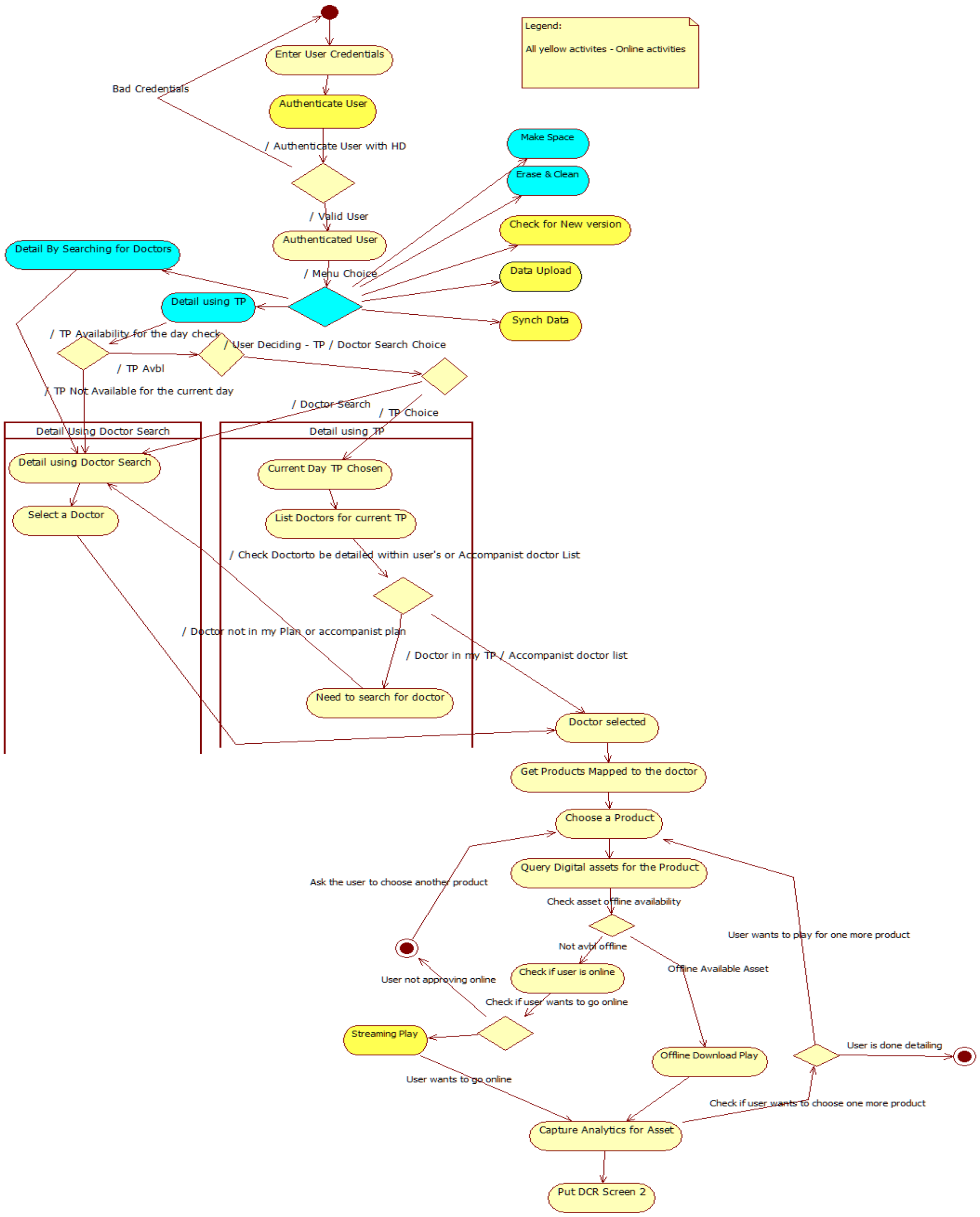
Table structure

COMPANY KEY	ACTION	INTENT	INTENT_TYPE
XYX	ALLOCATED_DB_SIZE_IN_GB	20	NA
XYX	CAN_ADD_OWN_TAGS	N	NA
XYX	DOWNLOAD_BITRATE	300	NA

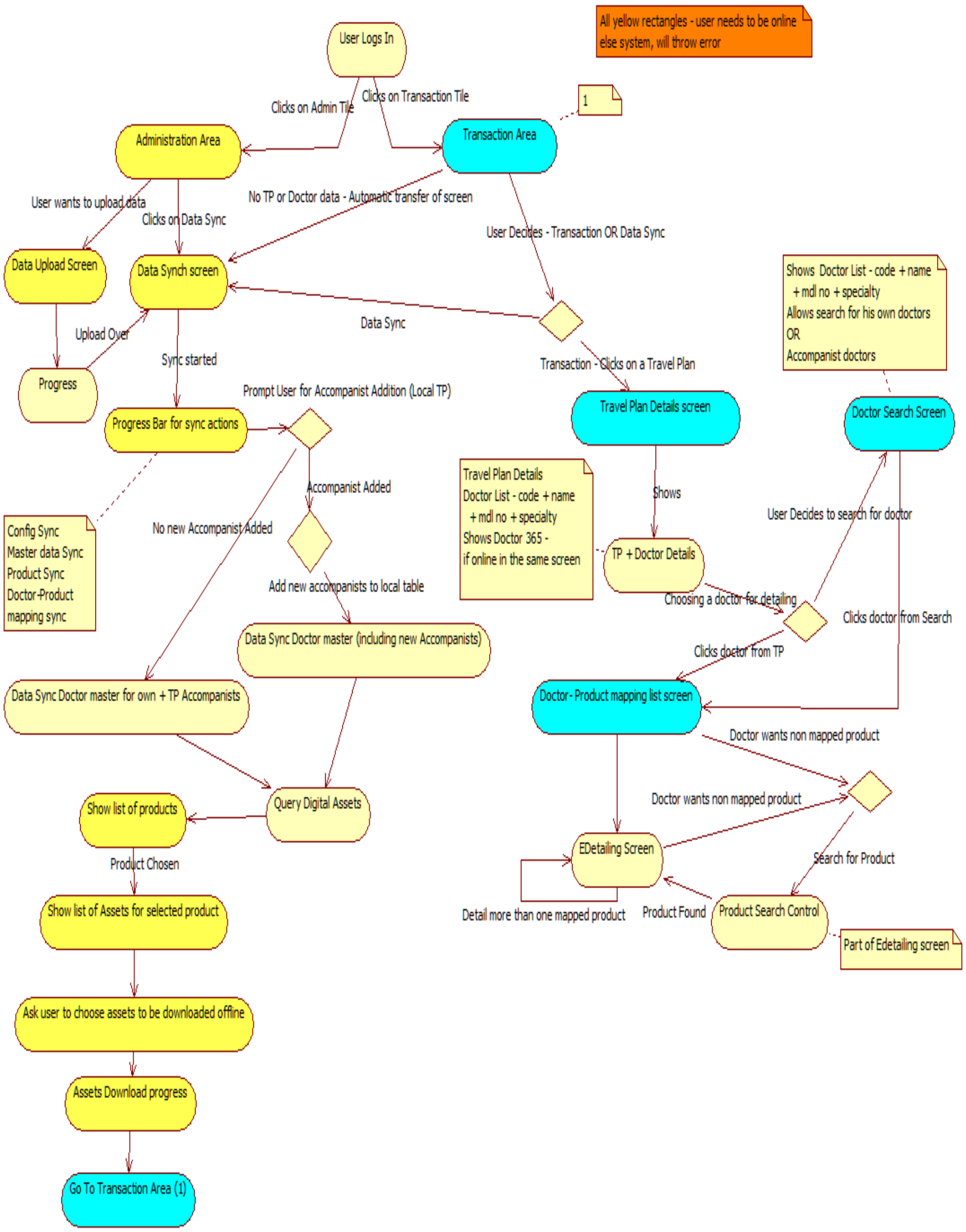
XYX	STREAMING_BITRATE	800	NA
XYX	SYNC_NO_OF_DAYS	7	NA
XYX	DATE_SETTINGS	dd/mm/yyyy	NA
XYX	TOTAL_FIELDS_TO_QUERY	2	NA
XYX	QUERY_PARAM_SPOTLIGHT	COL3	NA
XYX	QUERY_PARAM_ALL_VIDEOS_VARIABLE	COL1,COL2, COL3,COL4,COL5, COL6	
	QUERY_PARAM_ALL_VIDEOS_CONSTANT_TBLNAME	TBL_USER_INFO	NA
	TBL_NAME_SQLLITE_FTS_METADATA	TBL_FTS_EDET	NA
XYX	COL1	DOCSPE	
XYX	COL2	DOCCAT	
XYX	COL3	DOCMKT	
XYX	COL4	USRROL	
XYX	COL5	USRHIE	
XYX	COL6	USRDIV	
XYX	COL7	PDTCDE	
XYX	COL8	DOCCDE	

Comment [vp1]: Each one of this COL series must be interpreted for the INTENT and the intent specify the values that get tagged at the administrative UI

Business flow diagram


Legend:
All yellow activites - Online activities
Enter User Credentials
Bad Credentials
Authenticate User
/ Authenticate User with HD
/ Valid User
Authenticated User
/ Menu Choice
Make Space
Erase & Clean
Check for New version
Data Upload
Synch Data
Detail By Searching for Doctors
Detail using TP
/ TP Availability for the day check
/ User Deciding - TP / Doctor Search Choice
/ TP Avbl
/ TP Not Available for the current day
/ Doctor Search
/ TP Choice
Detail Using Doctor Search
Detail using TP
Detail using Doctor Search
Select a Doctor
Current Day TP Chosen
List Doctors for current TP
/ Check Doctorto be detailed within user's or Accompanist doctor List
/ Doctor not in my Plan or accompanist plan
/ Doctor in my TP / Accompanist doctor list
Need to search for doctor
Doctor selected
Get Products Mapped to the doctor
Choose a Product
Query Digital assets for the Product
Ask the user to choose another product
Check asset offline availability
Not avbl offline
User wants to play for one more product
Check if user is online
Offline Available Asset
User not approving online
Check if user wants to go online
Streaming Play
Offline Download Play
User is done detailing
User wants to go online
Check if user wants to choose one more product
Capture Analytics for Asset
Put DCR Screen 2

Screen flow and transitions


All yellow rectangles - user needs to be online else system, will throw error
User Logs In
Clicks on Admin Tile
Clicks on Transaction Tile
1
Administration Area
Transaction Area
User wants to upload data
Clicks on Data Sync
No TP or Doctor data - Automatic transfer of screen
User Decides - Transaction OR Data Sync
Data Upload Screen
Data Synch screen
Data Sync
Shows Doctor List - code + name + mdl no + specialty
Allows search for his own doctors
OR
Accompanist doctors
Upload Over
Sync started
Transaction - Clicks on a Travel Plan
Progress
Prompt User for Accompanist Addition (Local TP)
Progress Bar for sync actions
Travel Plan Details screen
Doctor Search Screen
Accompanist Added
Travel Plan Details
Doctor List - code + name + mdl no + specialty
Shows Doctor 365 - if online in the same screen
Shows
Config Sync
Master data Sync
Product Sync
Doctor-Product mapping sync
No new Accompanist Added
User Decides to search for doctor
TP + Doctor Details
Add new accompanists to local table
Choosing a doctor for detailing
Clicks doctor from Search
Data Sync Doctor master (including new Accompanists)
Clicks doctor from TP
Data Sync Doctor master for own + TP Accompanists
Doctor - Product mapping list screen
Doctor wants non mapped product
Doctor wants non mapped product
Show list of products
Query Digital Assets
EDetailing Screen
Search for Product
Product Chosen
Detail more than one mapped product
Product Found
Product Search Control
Part of Edetailing screen
Show list of Assets for selected product
Ask user to choose assets to be downloaded offline
Assets Download progress
Go To Transaction Area (1)

Scenarios

HD-ED-CLIENT-001 Authentication & Authorization

Purpose: The logged user needs to be authorized on every login and his current status and role needs to be authorized on every logical set of transactions.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-001-01	The system on start of the application need to validate if the login credentials are already entered and present. If not the system should halt any activity and prompt the user to enter login information along with URL	Local
REQ-CLIENT-001-02	User id / password / URL for the user who is in-charge of the tablet needs to be stored in the tablet such that during subsequent logins the system does not prompt for user id and password	Local
REQ-CLIENT-001-03	System must authenticate the user for his validity for user id / password / url combination and thus store the data permanently in a data store	
REQ-CLIENT-001-04	User must be able to erase the user id / pwd / url combination at will.	Local

HD-ED-CLIENT-002 ERASE & CLEANUP

Purpose: When a tablet device is being shared, the user who was owning the device must be able to clear his credentials and all the offline downloaded items before handing over the device.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-002-01	**ERASE:** The system will have an administrative menu called as "Erase and Clean". Clicking on this button will confirm with the user that "All data including Login credentials and offline downloaded files will be deleted. Are you sure you want to continue". If the user confirms then the system will 1. Clear the offline downloaded files including video / audio / images / offline downloaded files etc 2. Clear his login credentials including user id / password / url 3. Clear all configuration values 4. Clear the tags and tag related information Once the activity is done, a confirmatory message will be shown to the user that the device can now be handed over,	Local

HD-ED-CLIENT-003 MAKE SPACE

Purpose: When the user runs out of space during regular operations of the tablet or during attempt of a download, this administrative menu can be used to clear some more space. This is not a replacement for a general tablet file explorer system utility, but this can clear data from the HiDoctor EDetailing client folder only.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-003-01	**CLEANUP:** The user might have run out of space and would wish to clear the existing offline downloads one by one or all at one click. To enable this, in the administrative section, the system must list all existing offline downloaded files in the form of a grid showing the following data: Name of the digital asset Type of digital asset Size Download date / time Last updated date / time The user must be given an option to select one or more digital assets and click on "Delete". The system should perform the following actions 1. Remove the digital asset from storage / sd card 2. Reach to SQL Lite and query by the Digital asset id 3. In the resulting record, change the "FileMode" field to "Offline" and clear the value for field "OfflineURL"	Local

HD-ED-CLIENT-004 Data Inbound sync tasks

Purpose: Data inbound sync tasks covers all the different data that needs to be downloaded to the client system so that the user can work offline with data and records that are retrieved from the HD source system.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-004-01	Every time Data sync has been clicked the system must call this API to ensure that the user is still in active status and his password is not changed. If the user's authorization details return a false, then the system must halt the user from performing any action and take him back to the login screen where he must be forced to enter his credentials.	
REQ-CLIENT-004-02	System must check if the logged in user has travel plans enabled and available.	
REQ-CLIENT-004-03	**Travel Plan Details (RP)** System should sync Travel Plan data from the HD source system. Travel plan data is for the current date + SYNC_NO_OF_DAYS. (Sundays included). The retrieved data needs to be stored in the local SQL lite database. The received data will contain 1. Travel plan Details in a delimited format 2. Accompanist details that were chosen as the part of the original travel plan 3. System must store these two details separately such that a relational key exists between them for joining purposes.	
REQ-CLIENT-004-04	**Logged on user details synch:** The system needs to the fetch the following details against the logged on user: User Id User Territory Hierarchy Division to which the user is attached to (Optional) Region Code of the user Priority product codes (Refer technical API for more details) The above details need to be set as the default context and that's to be used across the navigation screens of the system.	
REQ-CLIENT-004-06	**Configuration Synch:** The HD system stores configuration values that governs the client android system. These configuration values needs to be refreshed on a regular basis (or at least whenever the user synchs) such that any changes made at source HD system are are reflected in the client system.	

	The configuration values are shown on the top and they need to be persisted across the entire life time of the application. They need to be cleared only when the user attempts to do an “ERASE & CLEANUP”	
REQ-CLIENT-004-07	**Calendar data synch:** System will pass the logged on user ID to get the DCR calendar details for the month. Follow - HD-ED-CLIENT-007 Calendar - for details. The fetched data is to be internally stored such that when the calendar view is displayed, system can show the DCR status superimposed against every day.	
REQ-CLIENT-004-10	**Create more accompanists:** After the initial data sync, the system would prompt the user to create a local TP so that transactional data required for offline operations can be downloaded. When prompting, the system would traverse the TP downloaded and get a list of all accompanists and then list them out in the screen. Additionally the user is being prompted for adding any more accompanists. In the previous step, if the system finds that there are no accompanists, the system would just report “No Accompanists fund”, but the flow remains the same for a user who is of a MR or a manager role or already has a Travel Plan or note. If the user decides to select accompanists in this screen, the system will bring all users who are part of his reporting hierarchy (for Manager) OR one level up hierarchy (For medical rep) and then give an option to choose up to 4 accompanists (essentially people who accompany the medical representative during doctor visits). Once maximum of 4 accompanists are (optionally) chosen system will download all doctor data related to those 4 accompanists + logged in users doctors + accompanist doctors that are part of travel plan. The complete format would be **DOCSPE_docspe001#DOCCAT_supercore#USRROL_userroleRM#USRHIE_India~Zone004~TNRegion~Area001~Chennai001#DOCCDE_doc002# DOCMKT_mkt000000123** When the system gets this as output, system should perform the following steps: 1. Run a for each loop of every result set 2. Split by “#”, this is the top key – value pairs, further split by “underscore” – this is the key as the first part of array and values as second parts of array. 3. For each key-value pair result, system should query configuration values to find the correct COL to prepare an insert statement. For ex: for the key – DOCSPE, the equivalent COL is COL1 DOCSPE, Similar operation for all the distinct KEYS. 4. System gives me the flexibility to add and remove any number of new COL until COL10. Now we need to prepare an insert statement to insert in to Tbl_DENORM_DIGASSETS_QUERY_INPUTS. In this table column names are only COL1, COL2 etc. So based on the column names identified using each KEY the corresponding values are to be inserted. So for the above, insert statement would be something like **Insert in to** **Tbl_DENORM_DIGASSETS_QUERY_INPUTS**	

	COLUMNS (COL1, COL2, COL4, COL5,COL8) **VALUES (‘DOCSPE_docspe001’,’ DOCCAT_supercore’,’ USRROL_userroleRM’,’ USRHIE_India~Zone004~TNRegion~Area001~Chennai001’,’ DOCCDE_doc002’)**	
REQ-CLIENT-004-09	**Digital asset Auto synch:** From the previous step, the doctor data on a combination with user profile would have got synched to the local sql lite database for both the logged in user and the chosen accompanist user. System should query the list of all these doctors from the local sql lite and prepare a distinct matrix of queries that will send a UNIQUE combination of Doctor Category Code + Doctor Specialty Code + User Hierarchy Code + User Role Code + Division Code (Optional). **Algorithm for the above:** **The following items needs to be cleared before every auto sync** 1. De-normalized query inputs table – table that contains COL1 through COL10 values 2. Tbl_DIGASSETS_MASTER - Clear all records whose “mode” field is not “offline”. We are not clearing “offline” records because offline records have a downloaded video file associated with it. 3. Tbl_DIGASSETS_RESULTS table – clear all records 4. Rest of the master table records including Tbl_DENORM_DIGASSETS_QUERY_INPUTS and Tbl_DIGASSETS_UNIQUE_PDTCODES The above query needs to be use the following generic values such that in future any new attributes such as Doctor Category Code etc can be removed or added without change in the android client. 5. From the configuration values set, get QUERY_PARAM_ALL_VIDEOS_VARIABLE. This should give a list of COL names that are separated by a “comma”. Split them and get each and every COL(X) value. 6. Go to Tbl_DENORM_DIGASSETS_QUERY_INPUTS. This must have been populated in the previous step and would contain every combination of doctor profile + user profile. The doctor profile + user profile combination is not unique because the same doctor profile and user profile will repeat. So get UNIQUE combination of Doctor Category Code + Doctor Specialty Code + User Hierarchy Code + User Role Code + Division Code (Optional). **Ex:** **DOCSPE_docspe001#DOCCAT_doccatgp001#USRROL_userroleRM#USRHIE_India~Zone004~TNRegion~Area001~Chennai001** The above is an example of one set of unique values for a given doctor row. When the system queries the next doctor + user profile, system should check	

for uniqueness of the second profile and then construct the query. This is to minimize number of same profile queries going to DAM server. All queries are a single array and treated as “AND clause” queries.

Finally a single query to DAM server will look like **FindDigitalAsset(“DOCSPE_docspe001#DOCCAT_doccatgp001#USRROL_userroleRM#USRHIE_India~Zone004~TNRegion~Area001~Chennai001”).** The actual API will be in the interface section. This query needs to be fired for every unique combination found in the earlier array creation.

7. Note that we have not sent the product code or doctor code and that is intentional, because we need to infer the product codes for the given combination of Doctor + User Profile.

8. Once the above query is fired to the DA database, DA database would return results based on the query.

9. System would have during installation time created a Full Text Search enabled virtual table with three columns – Refer - Tbl_DIGASSETS_RESULTS

10. DA API would return value of Digital assets and its attributes such as Name, Size, tag based metadata. The metadata is special and will be in the following format - **DOCSPE_docspe001# DOCSPE_docspe002# DOCSPE_docspe003#DOCCAT_supercore# DOCCAT_noncore# DOCCAT_core# USRROL_userroleRM#USRHIE_India~Zone004~TNRegion~Area001~Chennai001#PRDCDE_prd001#PRDCDE_prd002#OFFLINE_YES# DOCMKT_mkt000000123# LastTagUpdatedTimeStamp_1/2/2011# LastFileUpdatedTimeStamp_1/2/2011.** These values need to be stored in the local SQL Lite Database in the table - **Tbl_DIGASSETS_RESULTS.**

11. **Show videos matching to marketing code –**
    To allow the users to offline download spotlight videos (videos matching to marketing campaign that doctors are part of) the following needs to be done:
    - Query configuration table to get the column name for keyword “DOCMKT”
    - Use the column name to get the list of marketing codes that are unique in the table Tbl_DENORM_DIGASSETS_QUERY_INPUTS
    - Tbl_DIGASSETS_RESULTS by applying the query – Psuedo query – Select * from Tbl_DIGASSETS_RESULTS where MATCH Tags = “DOCMKT_<Each one of the marketing codes>”.
    - For every **unique** marketing code prepare and fire a query to the Digital Asset management solution Finally a single query to DAM server will look like **FindDigitalAsset(“MKTCOD_0000000124”).** The actual API will be in the interface section. This query needs to be fired for every marketing code found in the earlier array creation.

12. Algorithm to fill the table Tbl_DIGASSETS_RESULTS and Tbl_DIGASSETS_MASTER:
    This needs to be done for both the results sets that were done earlier (1) For Marketing code only – Point 11 and (2) General query

    FOR Tbl_DIGASSETS_MASTER

    - There is a master table called Tbl_DIGASSETS_MASTER.
    - ***VIJAY: TBD : DAM needs to give ONLINE URL AND OFFLINE URL***

    The below is a for each loop – meaning the Digital Asset could be mapped to many products but we need to create a combination of product code + DA code for EVERY product to which digital asset is mapped to. This is to ensure that when we search by product we can easily get to the offline or online URL

    - In the result set,
        a. Take the digital asset unique ID and name and create a record.
        b. In the above master table, all digital assets "mode" field in the SQL Lite database needs to be kept as "Online" and "Online URL" field needs to be updated to point to the online URL.
        c. get the product codes that each of the digital assets are mapped to. Easiest way is get all strings that are mapped as PRDCDE_XXXXX#PRDCDE_XXXXX#PRDCDE_XXXXXX. Store this value in the column "Product Code". Product codes are part of the overall metadata that is returned from digital assets side, so we need to pull product codes from the rest of the large string by using the prefix :PRDCDE_". Fill this for the product code in the master - Tbl_DIGASSETS_MASTER
        d. For every product code get the DA's LastFileUpdatedTimeStamp value and LastTagUpdatedTimeStamp and DA_FileUploadDateTime and fill them in the relevant columns.

    FOR Tbl_DIGASSETS_RESULTS

    - Get the product codes that each of the digital assets are mapped to. Easiest way is get all strings that are mapped as PRDCDE_XXXXX#PRDCDE_XXXXX#PRDCDE_XXXXXX. Store this value in the column "Product Code". Product codes are part of the overall metadata that is returned from digital assets side, so we need to pull product codes from the rest of the large string by using the prefix :PRDCDE_". Fill this for the product code in the master - Tbl_DIGASSETS_RESULTS

    - Store the remaining metadata (Without product codes) in the metadata column of the full text table - Tbl_DIGASSETS_RESULTS

13. **Finding unique Product Code for display**
    Store **every unique product code** in the previous step in to table - **Tbl_DIGASSETS_UNIQUE_PDTCODES.** During inserting to this table join with

Product master so that we can store product code and product name. This is what we will use to show in the left side of the eDetailing main screen for the user to choose and show product assets.

14. **Newer versions of offline content:**

Once Tbl_DIGASSETS_RESULTS have got filled we need find “Newer versions of offline content”. Since we did not clear “mode” = “offline” records earlier, we need to find newer versions of these assets if available.

14.1.1 How to use LastFileUpdatedTimeStamp: (Given from DAM API)

We will need to check against the Digital Asset if the already available LastUpdateTimeStamp is “LESSER” than the time stamp returned from API. If it is then

- ➢ System will delete the offline asset and store the name of the asset, time stamp in SQL Lite (old) and the new time stamp in a temporary string variable.
- ➢ Repeat this process for ALL the product and keep building this string variable.
- ➢ For all deleted records change the “mode” to online”. This is the way we can make the user aware of the fact that he needs to re-download the asset again.
- ➢ At the end of the operation inform the user of the offline assets that were deleted with an information that he needs to redownload the assets again.

Synchronize Tag related data

Asset_Tag_master is a persistent table that needs to be created as the SQL Lite end as a part of the installation scripts. Every time the user (Doctor / user) adds a new Tag in the form of a comment on to the system the Asset Tag Master gets locally populated. To ensure that user is prompted with pre-configured and tags entered by other users, a separate API call will be made that will download a number of tag text. These are to be stored to the SQL Lite system.

Additionally for every DigitalAsset ID that is getting downloaded system will provide a DigitalAsset marketing analytical information that will show the likes, dislikes, view count and star value such that they can be shown to the user.

The down sync values need to tbe stored in **Tbl_DA_Analytical_SyncData** and must be displayed in the UI based on the Digital Asset ID.

Sync Down Product images:

Call the API from HD to sync down product images and store them locally in an accessible folder. No processing necessary except de-serializing and making the images as physical image.

HD-ED-CLIENT-004.a View Assets and marking for Offline download

Purpose: This use case maps to the screen where we show all unique product codes from table Tbl_DIGASSETS_UNIQUE_PDTCODES in the left side of the screen.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-004.a-01	**Show videos matching to the chosen product code** This use case maps to the screen where we show all unique product codes from table Tbl_DIGASSETS_UNIQUE_PDTCODES in the left side of the screen. The first product code will be default selected and the relevant assets will be picked from the offline table - Tbl_DIGASSETS_RESULTS by applying the query – Psuedo query – Select * from Tbl_DIGASSETS_RESULTS where MATCH ProductCode = "chosen product code".	
REQ-CLIENT-004.a – 01	**Show videos matching FOR EVERY MARKETING CODE** To allow the users to offline download spotlight videos (videos matching to marketing campaign that doctors are part of) the following needs to be done: 1. Query configuration table to get the column name for keyword "DOCMKT" 2. Use the column name to get the list of marketing codes that are unique in the table Tbl_DENORM_DIGASSETS_QUERY_INPUTS 3. Tbl_DIGASSETS_RESULTS by applying the query – Psuedo query – Select * from Tbl_DIGASSETS_RESULTS where MATCH Tags = "DOCMKT_<Each one of the marketing codes>".	
REQ-CLIENT-004.a - 02	**Choose and Select Digital assets to populate / offline download** This section is common for marketing section videos or for the selected product code. Once the digital assets and populated values to table - Tbl_DIGASSETS_RESULTS, we need to get a "distinct" list of ALL products that has got populated in the field "ProductCode" on all the rows. The current format will be PRDCDE~xxxx~xxx on all rows. Prepare an array of all the distinct products and show them in the view digital assets / download digital assets page such that user can choose a product and see the downloaded digital assets that were auto synched. The user will also have an opportunity to mark items for offline download. On choosing the product, a query to the table - **Tbl_DIGASSETS_RESULTS** would be made that would bring all the digital assets mapped to the product purely by using the "product code"	

	as the only key. Since the table Tbl_DIGASSETS_RESULTS is a Full Text Search enabled table, the MATCH clause can be used to get all rows that match to multiple digital assets code. Once digital assets code have been obtained, get the digital asset details from Tbl_DIGASSETS_MASTER ta ble. Once the list of digital assets have been shown to the client (android) the system will show these digital assets in a grid with a check box like option next to each of them enabling the user to download. The check box will stay disabled for assets that have the "online" only attribute set to "true thus disallowing a user from selecting for download. This can be found by **OFFLINE_YES** attribute. The user can download one or more digital assets and schedule them in a QUEUE for download. The user is now free to move to the next product and choose another set of digital assets to be downloaded. The following are the summary of actions: 1. When the system brings the list of digital assets to display in the grid, SQL Lite must be populated with all of the digital asset information with "Mode" as "Online" and update the "OnlineURL" field with the url returned by the digital asset. This is in table Tbl_DIGASSETS_MASTER. 2. When the user selects a digital asset to be locally downloaded then after the download system must update the "Mode" field to be "Offline" and update the "OfflineUrl" to be the local storage URL. System must **allow multiple downloads** by scheduling the download assets in a queue. User must not be restricted to selecting single download at a time. A progress bar indicating the download completion must be shown to the user at all points of time. Thus downloads must be asynchronous and must be initiated in a separate thread(s).	

HD-ED-CLIENT-005 View TP for detailing

Purpose: The assumption for this use case is that, user will select a Travel plan from the travel plan screen to view the list of doctors per the plan. This scenario aides that.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-	System will query its local database and retrieve the TP details that are	Local

005-01	available. The screen should all the details of the TP per the interface in the interface column.	
REQ-CLIENT-005-02	When the user clicks on a TP, system will show the doctors that are assigned to the TP. The relationship between TP and doctors are established as follows: When the data sync happened, the system has pulled the TP details which has the doctor code and MDL number attached with it. The system also has synched the list of doctors who are mapped to the logged on user / accompanist per TP - with the code and MDL number. Now the system can establish a relational link between these two information to retrieve the list of doctors / their details and Travel plan details.	
REQ-CLIENT-005-03	While showing list of doctors, the following data needs to be shown as a part of doctor details Doctor Name Doctor MDL No Etc per the interface	
REQ-CLIENT-005-04	**360 degree history pane:** The system should find if the user is online with internet connectivity, either with a 2G/ 3G / WiFi. If connectivity can be established, system will get the 360 degree (history) information about the doctor and display in a control. For details see "View Doctor 360 Details" scenario. If the system is offline then the call to get doctor 360 will not be made. Instead the system will display an error message that the "Cannot show 360 degree information. User is not online"	

HD-ED-CLIENT-006 View Doctor 360 Details

Purpose: To view 360 degree history details about a doctor. Includes last visited dates, samples – non-samples provided details etc

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-006-01	Doctor 360 degree is history information and this is a separate screen that shows historical information about a doctor's visit, The output of the HD API will be set of key value pairs and they need to be shown on the screen. This is a simple screen with no clickable links in the page.	

HD-ED-CLIENT-007 Search and Select doctor for detailing

Purpose: This scenario is when the user chooses to take the doctor route for detailing action. This shows a list of doctors that has been synched down from the HD source system. When the user choose a doctor, the system should display the list of products (product codes) that are mapped to the doctor

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-007-01	System has to query from the local table the list of accompanists that was either chosen as a part of TP or chosen by user during data sync. The system will show the accompanist names in the control for the user to choose an accompanist to see the doctors.	
REQ-CLIENT-007-02	Query the local database to get list of doctors who have been synched in to the system. This querying is with the context of doctors that belong to the chosen accompanist's region code in the previous step. Display the list of doctors in a grid like format so that the user can choose a doctor for detailing.	
REQ-CLIENT-007-03	If the user chose to search for doctors that are mapped to him, system needs to search for doctors whose mapping region code belongs to the logged on user's region code.	
REQ-CLIENT-007-04	Query the local database to get details of the list of products that are mapped to the selected doctor. The system will pass the doctor Id + MDL number and get the list of all products that are mapped to the selected doctor.	REQ-CLIENT-007-03
REQ-CLIENT-007-05	If no products are mapped to the chosen doctor, the system should inform the user that "The chosen doctor does not have any products mapped. Please contact the administrator" and remain on the same screen.	REQ-CLIENT-007-04

HD-ED-CLIENT-008 Calendar

Purpose: Shows the calendar view of the DCR for the entire month period. This calendar is just a view but does not have a calendar type functionality. It is a read only view of the user's DCR summary for the whole month.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-008-01	List of DCR summary for the calendar. Shows a calendar monthly view and against every date prints text that is passed back by the HD system specifically for calendar / day combination. Other than displaying the text the calendar control or view does not link to anything else. The data for this is already made available in the local SQL Lite database during the data sync stage.	

HD-ED-CLIENT-008 Digital Asset Render

Purpose: This scenario is invoked when the MR has chosen a TP / Doctor (No TP for MR who do not have TP) / Product and reached the detailing screen. Thus digital asset page is reached with a context of Rep (or) Manager (Role ID) / Territory Hierarchy / Division Code (Optional) / Doctor ID / Doctor Category / Doctor Specialty Code / Marketing Campaign Code / Product Code.

On load of the detailing screen the system has to query the digital assets using the combinations of codes and data available to show the digital assets in the user interface.

The way to link product codes to be shown in the Digital asset render screen is as follows:

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-008-01	**Choose Products to show in Digital Asset Render screen:** Take the chosen doctor's profile values – Doctor Category, Doctor Speciality and create a LIKE clause for SQL Lite to executed against FULL TEXT search table - Tbl_DIGASSETS_RESULTS. The query would be like Select * from Tbl_DIGASSETS_RESULTS table where DAMetaData LIKE '%<Doctor Category>%' AND DAMetaData LIKE '%<Doctor Speciality>%' The digtal asset render screen shows the following against every product (left side listing). 1. **Product Name (Join to product master)** 2. **Product Speciality (Join to product master)** 3. **Product Category (Join to product master)** 4. **Product Brand ((Join to brand master)** 5. **Product Image (HardCode this string – Your images folder path / PDTCAT_ProductCategoryCode.jpg)** **(Notice that images would have been synched down as a part of the Auto Sync activity)**	
REQ-CLIENT-008-02	**Render Assets in UI:** **When rendering Digital assets (video / document) links on the screen the following needs to be taken care of** 1. If the digital asset type is "Online", the "offline download" icon must not be provided. 2. If the digital asset type is "offline" playable then two checks needs to be done. Check if the digital asset file for the digital asset unique ID is already available offline in the videos folder. a. If it is, then, an offline "play" icon needs to be provided. b. If not, then, an "offline" download icon needs to be provided so that users can optionally download the	

	digital asset. Near to that a regular "play" needs to be provided so that users can stream the movies.	
REQ-CLIENT-008-03	**Fill Spotlight section:** In the user interface, to fill the spotlight section, a API call needs to be made to the Digital Asset management interface with the relevant parameters **Marketing campaign code** If Digital asset links are available in the return of this API call, those links need to be populated either in the video pane or in the office document pane based on the extension.	Estimated download size of the video to be provided back in API
REQ-CLIENT-007-04	**Fill Detailing Item section:** In the user interface, to fill the detailing section, an API call needs to be made to the digital asset management interface. Params: **And** clause of • Doctor Category • Doctor Specialty • Territory hierarchy • Product Code • Division Code (Optional) • User Type (Role) If Digital asset links are available in the return of this API call, those links need to be populated either in the video pane or in the office document pane based on the extension. On a second call to the same web service, call with the following parameters, List of product codes that are mapped to the doctor chosen If Digital asset links are available in the return of this API call, those links need to be populated either in the video pane or in the office document pane based on the extension. This list is in addition to the first list of links that were fetched in the first API call.	

HD-ED-CLIENT-008 Digital Asset Streaming view

Purpose: When the user clicks on the video control that has already been provided with a streaming URL, the video will start playing.

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-	When the user clicks on a video link on the video pane the system	

008-01	needs to check 1) If the Video URL is point to an online streaming URL, if it is then the system should check if the user is online, if the user does not have internet connectivity, then the system should alert the user that "Video cannot be played as internet connectivity is not available". 2) If the Video URL is point to an online streaming URL, if it is then the system should check if the user is online, if the user has internet connectivity, then the system should pass the video URL that is available in the local SQL Lite database and pass that to the Video control. The video control will then play the video from the streaming control.	
REQ-CLIENT-008-02	When the user clicks on a Microsoft office document link on the document pane the system needs to check 1) If the document URL is point to an online URL, if it is then the system should check if the user is online, if the user does not have internet connectivity, then the system should alert the user that "Document cannot be shown as internet connectivity is not available" else document must be shown. 2) If the document URL is point to an offline URL system must open the document using the relevant application for office documents and pdf.	
REQ-CLIENT-008-03	**Fill billing for play** **IF DIGITAL ASSET IS PLAYED FROM OFFLINE STORAGE (SD CARD)** **Fill billing for offline download** After every successful offline download of a video, system must create a record in table - tbl_DA_Itemized_Billing Mark "**Offline_Click**" field as "1" against a DA ID with all other contextual information including DATE TIME of offline download **IF DIGITAL ASSET IS PLAYED FROM ONLINE SERVER (STREAMING)** After every successful offline download of a video, system must create a record in table - tbl_DA_Itemized_Billing Mark "**Online_Play**" field as "1" against a DA ID with all other contextual information including DATE TIME of online play	

HD-ED-CLIENT-009 Digital Assets offline download and save

Purpose: Ability for a user to download any digital asset to local android device such that the local copy of the video file is used

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-009-01	**VIDEO FILES:** When the user clicks the "offline download" button, even before	

	saving the video the system must check the current space available in the system and check the expected download size. The expected download size would have been made available as a property of the video / office document link that was got from the Digital Asset provider. If the download size exceeds the available space, the system must prompt the user that "Space is unavailable, please create some storage space and then retry the operation".	
REQ-CLIENT-009-02	**VIDEO FILES:** Assuming the storage space is made available, When the user clicks the "offline download" button, the system requests the Digital service API with "DOWNLOAD_BITRATE" constant to get the offline format of the video. This video will be downloaded to the local sdcard / storage folder and In the SQL Lite "offline URL" needs to be altered pointing the local storage card .	
REQ-CLIENT-009-03	**Fill billing for offline download** After every successful offline download of a video, system must create a record in table - tbl_DA_Itemized_Billing Mark "**Offline**" field as "1" against a DA ID with all other contextual information including DATE TIME of offline download	

HD-ED-CLIENT-010 Create Digital Asset Analytical Data

Purpose: This is an automatic action by the system to capture user's interaction with digital assets. This action is on a digital asset such as "Video" / "Audio" that is being played offline from a SD card. The scenario is triggered when the

a. User signals end of presentation either by way of stopping the video using the "stop" button or
b. When the user shifts to some other video by clicking somewhere else
c. Clicks the back screen to some other screen

Requirement ID	Requirement Description	Technology Interface from HD
REQ-CLIENT-010-001	**During play of a video / audio file** – The system must record the context of the play action. The context includes – UserID, UserURL, Digital Asset ID, Doctor ID, Doctor MDL number, Region code of the logged in user, offline or online, play start time of the video, play end time of the video, total play time, date time of the system. **If the system is offline:** If the system is offline then the context data needs to be stored offline in the local SQL Lite database. **If the system is online:** If the system is online during the end of video refer -	

	HD-ED-CLIENT-012 scenario for more information	
REQ-CLIENT-010-002	During view of a Microsoft office document – When the user clicks on a Microsoft Office document, the system will record the following information - UserID, UserURL, Digital Asset ID, Doctor ID, Doctor MDL number, Region code of the logged in user, "online", date time of the system	
REQ-CLIENT-010-003	Added tables for storing marketing analytics - DA_Usage_Data is the table where this data needs to be stored	See table design

HD-ED-CLIENT-012 Sync Analytics data to HiDoctor

Purpose: This action can be initiated by the user when he clicks on "Sync Data Manually" button or automatically when the system detects that the system is connected to the internet.

Requirement ID	Requirement Description	Technology Interface from HD
	Call the HD API with the following parameters Digital Asset ID Digital Asset Action = "Offline" Date Time Viewed UserID of the logged in user Once or more records are synched the user needs to have a message indicating that data was synched	
	More information on this analytics and other in the API	

HD-ED-CLIENT-013 Sync DCR Data to HD

Purpose: This action can be initiated by the user when he clicks on "Sync Data Manually" button or automatically when the system detects that the system is connected to the internet.

Requirement ID	Requirement Description	Technology Interface from HD
	On launch of the application at any time if the system finds that the user is online (has internet connectivity) system must asynchronously poll any un-synched data in the DCR or marketing analytics tables and prompt the user if he wishes to sync data to the master systems (HD and Digital assets screen). If the user says "Yes" then the system must sync the DCR data and marketing analytics data to respective systems and clear SQL Lite tables. At the end of the sync users must be told that data has been successfully synched.	

HD-ED-CLIENT-014 Capture User feedback (Digital asset screen)

Requirement ID	Requirement Description	Technology Interface from HD
	The bunch of controls that are shown under the Digital asset video screen namely 1. Current likes of the digital asset 2. Current dislikes of the digital asset 3. Total view of the digital asset 4. Ability for a doctor / user to Like / Dislike a video 5. Ability for a doctor / user to be able add remarks in the form of tags are all governed by a single configuration variable – USER_CAN_ADD_ _TAGS. **This variable is default set to "N" that means this entire control is NOT visible. If the variable is set to "Y" then the control needs to be shown.** **Capture Analytics** The users feedback on star rating / like / dislike and remarks need to be stored in a table called - **Tbl_DA_Tag_Analytics.** This needs to be capture for every interaction, that means that user cannot EDIT what has been put in, for ex: If the user rates 5 star + like first and then subsequently 2 star + dislike, system will capture **2 interactions and record in the tbl_da_tag_analytics** table and send the data for upsync. **Capture tags created by user** 1. Create a table called DA_Tag_Master – 2. This table needs to be used in the screen where the user types his remarks (tags) / star rating / Like / dislike. 3. The remarks column should auto - suggest applicable tags based on the first 2 - 3 letters that the user types in. Ex: If the user types - #I L - then we should show #I Like It etc. If the user chooses to type his own tags instead of selecting one thats displayed, the system has to two either of the following on click of "Apply Tags" >> trim and check if the SAME tag exists (User could have typed #I like video (notice lower caps for l and v). In this case no need to insert in to tag master >> if the tag does not exist, then the system will INSERT the tag in Tag master	

	Thus, tag master will serve as a local cache of tag data to be popped up during user remarks step.	

HD-ED-CLIENT-015 Sync down and Display Digital Asset Analytics history

Requirement ID	Requirement Description	Technology Interface from HD
	Create a table called DA_Analytics_History. Refer table section for table design 1. Call the HD API to get analytics history for all the digital assets that the company the logged on user belongs to. 2. The expected data are – Like Count / Dislike Count / Star rating average / Total Views 3. Clear the existing DA_Analytics_History table and re-insert the history information. 4. Use the history information to show analytical data in the relevant screen.	

HD-ED-CLIENT-016 Sync down and update new TAGS for local TAG Cache

Requirement ID	Requirement Description	Technology Interface from HD
	Clear the table DA_TAGS_Master 1. Call the HD API to get list of tags that are available in the master tags list 2. Insert the sent tag descriptions in to the da_tags_master table	

HD-ED-CLIENT-017 Sync Billing Data to HD

Purpose: This is to sync the billing data to the HD

Requirement ID	Requirement Description	Technology Interface from HD
	1. Sync up the data to the HD using exposed API **2.** Generate the string using **tbl_DA_Itemized_Billing (one by one Row with "^" Separation and pass it to the HD API)**	*HD-ED-CLIENT-API-023*

DB SCHEMA

1. tbl_User_Info

Column Name	Data Type	Allow Null?	Max Length
Company_Code	NVARCHAR	Yes	15
User_Name	NVARCHAR	Yes	30
Password	NVARCHAR	Yes	30
URL	NVARCHAR	Yes	50
User_Code (PK)	NVARCHAR	Yes	15
Region_Code	NVARCHAR	Yes	15
Region_Name	NVARCHAR	Yes	30
User_Type_Code	NVARCHAR	Yes	15
User_Type_Name	NVARCHAR	Yes	30
Region_Hierarchy	NVARCHAR	Yes	500
Last_Sync_Date	DateTime	Yes	

2. tbl_Accompanist

Column Name	Data Type	Allow Null?	Max Length
User_Name	NVARCHAR	NO	120
Region_Code	NVARCHAR	NO	15

3. tbl_User_Division

Column Name	Data Type	Allow Null?	Max Length
User_Code (PK) (Ref from tbl_User_Info)	NVARCHAR	No	15
Division_Code (PK)	NVARCHAR	No	15
Division_Name	NVARCHAR	No	30

4. tbl_Config_Settings

Column Name	Data Type	Allow Null?	Max Length
ALLOCATED_DB_SIZE_IN_GB	Float	No	
CAN_ADD_OWN_TAGS	Bit	No	
DOWNLOAD_BITRATE	Float	No	
STREAMING_BITRATE	Float	No	
SYNC_NO_OF_DAYS	Smallint	No	
DATE_SETTINGS	Date	NO	

5. Tbl_TP_Header

Column Name	Data Type	Allow Null?	Max Length
TP_Id (Pk)	Bigint	No	
Call_Objective	NVARCHAR	No	15
TP_Date	Date	No	
CP_Name	NVARCHAR	Yes	50
Work_Category_Name	NVARCHAR	No	50
Work_Place	NVARCHAR	YES	50

6. Tbl_TP_Accompanist

Column Name	Data Type	Allow Null?	Max Length
TP_Id(Ref from tbl_TP_header)	BIGINT		
Acc_Name	NVARCHAR		100
Acc_Region_Code	NVARCHAR		15

7. tbl_TP_Doctors

Column Name	Data Type	Allow Null?	Max Length
TP_Id (Ref from tbl_TP-Header)	Bigint	No	
TP_Doctor_Id (PK)	Bigint	No	
Doctor_Code	NVARCHAR	No	30
Doctor_Region_Code	NVARCHAR	No	30

8. tbl_TP_Products

Column Name	Data Type	Allow Null?	Max Length
TP_Doctor_Id (Ref From tbl_TP_Doctors)	Bigint	No	
Product_Code	NVARCHAR	No	15
Quantity	Smallint	Yes	

9. tbl_TP_SFC

Column Name	Data Type	Allow Null?	Max Length
TP_Id (Ref From tbl_TP_Header)	Bigint	No	
From_Place	NVARCHAR	NO	50
To_Place	NVARCHAR	NO	50

10. tbl_Speciality_Master

Column Name	Data Type	Allow Null?	Max Length
Speciality_Code (PK)	NVARCHAR	No	15
Speciallity_Name	NVARCHAR	No	30

11. tbl_Brand_Master

Column Name	Data Type	Allow Null?	Max Length
Brand_Code (PK)	NVARCHAR	No	15
Brand_Name	NVARCHAR	No	30

12. tbl_Product_Master

Column Name	Data Type	Allow Null?	Max Length
Product_Code (PK)	NVARCHAR	NO	15
Product_Name	NVARCHAR	NO	300
Product_Type_Name	NVARCHAR	NO	30
Brand_Code	NVARCHAR	NO	15
Speciality_Code	NVARCHAR	NO	15
Product_Category_Name	NVARCHAR	YES	30

13. tbl_Digital_Asset_Info

Column Name	Data Type	Allow Null?	Max Length
Product_Code	NVARCHAR	YES	15
Mode	NVARCHAR APPLICABLE VALUES (OFFLINE/ONLINE)	YES	15
Offline_URL	NVARCHAR	YES	250
Online_URL	NVARCHAR	YES	250

14. tbl_User_Product_Mapping

Column Name	Data Type	Allow Null?	Max Length
User_Code (PK)	NVARCHAR	NO	15
Product_Code (PK)	NVARCHAR	NO	15

15. tbl_Doctor_Category_Master

Column Name	Data Type	Allow Null?	Max Length
Category_Code (PK)	NVARCHAR	NO	15
Category_Name	NVARCHAR	NO	50

16. tbl_Customer_Master

Column Name	Data Type	Allow Null?	Max Length
Customer_Code (PK)	NVARCHAR	NO	30
Region_Code (PK)	NVARCHAR	NO	15
Customer_Name	NVARCHAR	NO	300
MDL	NVARCHAR	YES	30
Category_Code	NVARCHAR	YES	15
Speciality_Code	NVARCHAR	YES	15
Customer_Entity_Type	NVARCHAR	NO	10

17. tbl_MC_Doctors

Column Name	Data Type	Allow Null?	Max Length
MC_Code (PK)	NVARCHAR	No	15
Doctor_Code (PK)	NVARCHAR	No	30
Region_Code (PK)	NVARCHAR	No	15

18. tbl_DCR_Master

Column Name	Data Type	Allow Null?	Max Length
DCR_Date	DATE	NO	
Flag	NVARCHAR	NO	100
Status	NVARCHAR	NO	15

19. Tbl_DIGASSETS_MASTER

Column Name	Data Type	Allow Null?	Max Length
Product_Code	NVARCHAR	NO	50
DA_Code	NVARCHAR	NO	15
DA_FileUploadDateTime	DateTime	NO	15
DA_DownloadDateTime	DateTime	Yes	
Mode	NVarchar		
OnlineURL	NVarchar		
OffLineURL	NVarchar		
LastFileUpdatedTimeStamp	Datetime		
LastTagUpdatedTimeStamp	Datetime		

20. Tbl_DENORM_DIGASSETS_QUERY_INPUTS

Column Name	Data Type	Allow Null?	Max Length
COL1			
COL2			
COL3			
COL4			
COL5			
COL6			

COL7			
COL8			
COL10			

21. Tbl_DIGASSETS_RESULTS (FULL TEXT ENABLED VIRTUAL TABLE IN SQL LITE)

Column Name	**Data Type**	**Allow Null?**	**Max Length**
DACode (FK)			
Prd_Code	**Stored in ~ delimited format – like PRD_prd01~PRD_prd02~ PRD_prd03**		
DAMetaData	**Long Key_Value ~ delimited parameter list that has all the tags except product codes**		

Tbl_DIGASSETS_UNIQUE_PDTCODES

Column Name	**Data Type**	**Allow Null?**	**Max Length**
ProductCode			
ProductName			

Tbl_Selected_Accompanist

Column Name	**Data Type**	**Allow Null?**	**Max Length**
Acc_Name	NVARCHAR		100
Acc_Region_Code	NVARCHAR		15

Tbl_DCR_Doctor_Visit

Column Name	**Data Type**	**Allow Null?**	**Max Length**
Company_Code	NVARCHAR		30
User_Code	NVARCHAR		30
DCR_Actual_Date	DATETIME		
Doctor_Visit_Code(PK)	NVARCHAR	DOC00001_2013-12-31	100
DCR_Entered_Date	DATETIME		
Doctor_Code	NVARCHAR		30
Doctor_Region_Code	NVARCHAR		30
Doctor_Visit_Time	NVARCHAR		30
Remarks	NVARCHAR		500
Is_Accompanist_Doctor	BIT(0/1)		
Lattitude	NVARCHAR		30
Langitude	NVARCHAR		30

Tbl_DCR_Product_Details

Column Name	Data Type	Allow Null?	Max Length
Company_Code	NVARCHAR		30
Doctor_Visit_Code(FK)	NVARCHAR	DOC00001_2013-12-31	100
DCR_Product_Detail_Code(PK)	NVARCHAR	DOC00001_2013-12-31_PDC000001	100
Product_Code	NVARCHAR		30
Qty_Given	INT		
Is_Detailed	CHAR		1

Tbl_DCR_Chemist_Visit

Column Name	Data Type	Allow Null?	Max Length
Company_Code	NVARCHAR		30
Doctor_Visit_Code(FK)	NVARCHAR	DOC00001_2013-12-31	100
DCR_Chemist_Visit(PK)	NVARCHAR	DOC00001_2013-12-31_CMC000001_VAVA PHARMACY (or) DOC00001_2013-12-31_NULL_VAVA PHARMACY	100
Chemist_Code	NVARCHAR		50
Chemist_Name	NVARCHAR		50
POB	NUMERIC		9,2

Note: this table contains Chemist_Name because we allow the user to enter flexi chemist also, if the chemist code found in doctor master put the code in Chemist_Code for flexi chemist this column will be null.

Tbl_DCR_RCPA_Details

Column Name	Data Type	Allow Null?	Max Length
Company_Code	NVARCHAR		30
RCPA_Details_Code	NVARCHAR	DOC00001_2013-12-31_1	100
Doctor_Visit_Code(FK)	NVARCHAR	DOC00001_2013-12-31	100
DCR_Chemist_Visit(FK)	NVARCHAR	DOC00001_2013-12-31_CMC000001_VAVA PHARMACY	100

		(or) DOC00001_2013-12-31_NULL_VAVA PHARMACY	
Sale_Product_Code	NVARCHAR		30
Support_Qty	INT		
Competitor_Product_Name	NVARCHAR		50
Competitor_Product_Code	NVARCHAR		30

Note:

For sale product insert the row as

Sale_Product_Code = PRC00001

Support_Qty = 10

Competitor_Product_Name = NULL

Competitor_Product_Code = NULL

For Competitor (Pick the competitor from my own product)

Then ,

Sale_Product_Code = PRC00001

Support_Qty = 15

Competitor_Product_Name = NULL

Competitor_Product_Code = PRC00002

For Competitor (Pick the competitor flexi)

Then ,

Sale_Product_Code = PRC00001

Support_Qty = 15

Competitor_Product_Name = Atogla

Competitor_Product_Code = NULL

tbl_Sale_Product_Mapping

Column Name	**Data Type**	**Allow Null?**	**Max Length**
Sale_Product_Code	NVARCHAR	NO	15
Mapping_Product_Code	NVARCHAR	NO	15

tbl_DA_Itemized_Billing

Company_Code	**DA_Id**	**User_Code**	**User_Name**	**Region_Code**	**Region_Name**	**Division_Code**	**Division_Name**	**Date Time**	**Offline_Click**	**Downloaded**	**Online_Play**
COM000001	**1**	**USC00001**	**Senthil1234**	**REC000001**	**Trichy**	**DIV0000001**	**Div1**	**2012-01-01**	**1**	**1**	**7**
COM000001	**12**	**USC00002**	**Ram12**	**REC000001**	**Trichy**	**DIV0000001**	**Div1**	**2012-01-01**	**1**	**1**	**1**
COM000001	**13**	**USC00003**	**Ravi1**	**REC000023**	**Tnagar**	**DIV0000001**	**Div1**	**2012-01-01**	**1**	**1**	**1**
COM000001	**4567**	**USC00004**	**Raja**	**REC000045**	**Adayar**	**DIV0000001**	**Div1**	**2012-01-01**	**1**	**1**	**1**
COM000001	**8754**	**USC00005**	**Siva**	**REC000001**	**Trichy**	**DIV0000001**	**Div1**	**2012-01-01**	**1**	**1**	**1**
COM000001	**45**	**USC000006**	**Hari**	**REC000067**	**ECR**	**DIV0000001**	**Div1**	**2012-01-01**	**1**	**1**	**1**

Continuation of **tbl_DA_Itemized_Billing**

DCR_Actual_Date	**Product_Code**	**Product_Name**	**Doctor_Code**	**Doctor_Region_Code**	**Doctor_Code**	**Doctor_Region_Code**	LastFileUpdatedTimeStamp	LastTagUpdatedTimeStamp
<Calendar Date Time>								

Tbl_DA_Tag_Analytics :

Column Name	**Data Type**	**Allow Null?**	**Max Length**
DA_ID			
Doctor_Code			
Doctor_Region_Code			
User_Code			
Like			
Dislike			
Rating			
DateTime			
Tag_Description (Sample - #I Like Video~#Good Video)			

Tbl_DA_Tag_Master :

Column Name	**Data Type**	**Allow Null?**	**Max Length**
Tag_ID			
Tag_Description			
Tag_Used_Count			

DA_Analytics_History

Column Name	**Data Type**	**Allow Null?**	**Max Length**
DA ID			
TotalViewsCount			
TotalLikesCount			
TotalDislikesCount			
StarValue			

TECHNICAL SPECIFICATION

E-Detailing HiDoctor Android Version – API Document for client Application

HD-ED-CLIENT-001 Authentication & Authorization

HD-ED-CLIENT-API-001 Authentication & Authorization

Method Name:

CheckUserAuthentication

Input Parameters:

1. String correlationId (which is get from **StartSync API**)
2. string userName,
3. string password,
4. string subDomainName
5. out string result
   a. If the entered url is http://fdc.hidoctor.in then the subdomain is **fdc.hidoctor.in.** Hence the value of subDomainName parameter for this case will be fdc.hidoctor.in

Return Type and Format:

Boolean (true/false)

If the value is true, proceed to next step

If the value is false, get the out (result) and display the message to the user and stop the user

HD-ED-CLIENT-API-002 To Get the logged on user info

Method Name:

GetUserInfo

Input Parameters:

1. String correlationId (which is get from **StartSync API**)
2. string username (from tbl_User_Info),
3. string subDomainName (from tbl_User_Info),
   a. If the entered url is http://fdc.hidoctor.in then the subdomain is **fdc.hidoctor.in.** Hence the value of subDomainName parameter for this case will be fdc.hidoctor.in
4. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{"Company_Code":"COM00000011","User_Code":USC00000001,"Region_Code":"REC00000001","Region_Name":"Chennai 1","User_Type_Code":"UTC000000"1,"Region_ Hierarchy":" REC00000001~ REC00000002~ REC00000003~ REC00000004","User_Type_Name":"Active"}]}]}
```

The json string will have the following basic user info.

1. Company_Code
2. User_Name
3. Password

4. URL
5. User_Code
6. Region_Code
7. Region_Name
8. User_Type_Code
9. Region_Hierarchy
10. User_Type_Name

The Android client should store the above information in SQL lite along with username, password and url.

If the result is empty proceed further, else show message to the user

“Last_Sync_Date” he has to maintain after sync down the data successfully.

HD-ED-CLIENT-API-003 to Get Accompanist Details

Method Name:

GetAccompanistDetails

Input Parameters:

1. String correlationId (which is get from **StartSync API**)
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. string lastModifiedDate (from tbl_User_Info),
5. out string result

Return Type and Format:

Json String

{"Tables":[{"Rows":[{"User_Name":"Senthil001,TTM(LMELI)”,"Region_Code":"REC00001266”},{" User_Name":" nagarajapandianF0754,TTM(LM MADURAIT)”,"Region_Code":"REC00001266"}]}]}

The Json string will have the following details.

1. User_Name
2. Region_Code

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-004 to get the user division

Method Name:

GetUserDivision

Input Parameters:

1. String correlationId (which is get from **StartSync API**)
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. string lastModifiedDate (from tbl_User_Info),
5. out string result

Return Type and Format:

Json String

```
        {"Tables":[{"Rows":[{"User_Code":"USC00000001”,"Divison_Code":"DIV00000001”
,"Divison_Name":"DIV1”},{"User_Code":"USC00000001”,"Division_Code":"
DIV00000002","Divison_Name":"DIV2”}]}]}
```

The Json string will have the following details.

1. User_Code
2. Division_Code
3. Division_Name

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-005 to get config settings

Method Name:

GetConfiguration

Input Parameters:

1. String correlationId (which is get from **StartSync API**)
2. string companyCode (from tbl_User_Info),
3. string userCode
4. out result

Return Type and Format:

Json String

```
        {"Tables":[{"Rows":[{"Action":"ALLOCATED_DB_SIZE_IN_GB","Intent":"20"},{"Action":"EXTENDE
D_MEMORY_UPTO_IN_GB","Intent":"30"},{"Action":"DOWNLOAD_BITRATE","Intent":"300"},{"Action":"
STREAMING_BITRATE","Intent":"800"},{"Action":"SYNC_NO_OF_DAYS","Intent":"7"},{"Action":"DATE_SE
TTINGS","Intent":"dd/mm/yyyy"},{"Action":"TOTAL_FIELDS_TO_QUERY","Intent":"2"},{"Action":"QUERY_
PARAM_SPOTLIGHT","Intent":"COL3"},{"Action":"QUERY_PARAM_ALL_VIDEOS_VARIABLE","Intent":"COL
```

1,COL2,COL3,COL4,COL5,COL6,COL7,COL8"},{"Action":"QUERY_PARAM_ALL_VIDEOS_CONSTANT_TBLNAME","Intent":"TBL_USER_INFO"},{"Action":"TBL_NAME_SQLLITE_FTS_METADATA","Intent":"TBL_FTS_EDET"},{"Action":"EST_SIZE_BIT_RATE_STREAM","Intent":"1.1"},{"Action":"EST_SIZE_BIT_RATE_OFFLINE","Intent":"2.5"},{"Action":"COL1","Intent":"DOCSPE"},{"Action":"COL2","Intent":"DOCCAT"},{"Action":"COL3","Intent":"DOCMKT"},{"Action":"COL4","Intent":"USRROL"},{"Action":"COL5","Intent":"USRHIE"},{"Action":"COL6","Intent":"USRDIV"},{"Action":"COL7","Intent":"PDTCDE"},{"Action":"COL8","Intent":"DOCCDE"}]}]}

The Json string will have the following details.

1. ALLOCATED_DB_SIZE_IN_GB
2. CAN_ADD_OWN_TAGS
3. DOWNLOAD_BITRATE
4. STREAMING_BITRATE
5. SYNC_NO_OF_DAYS
6. DATE_SETTINGS

HD-ED-CLIENT-API-006 *to get TP header Information*

Method Name:

GetTPHeader

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. String userCode (from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

First Table Contains following:

{"Tables":[{"Rows":[{"TP_Id":"1234”," Call_Objective ":"FIELD” ," TP_Date ":"2012-01-01” ," CP_Name":"CPMNAME1” ," Work_Category_Name":"HQ” ," Work_Place":"Chennai 1” },

{"TP_Id":"5421”," Call_Objective ":"FIELD_RCPA” ," TP_Date ":"2012-01-02” ," CP_Name":"CPMNAME2” ," Work_Category_Name":"Ex-HQ” ," Work_Place":"Chennai 2” }]}]}

Second Table contains following Data

{"Tables":[{"Rows":[{"TP_Id":"1234”," Acc_Name ":"2012-01-01” ," Acc_Region_Code":"CPMNAME1” },

```
        {"Tables":[{"Rows":[{"TP_Id":"1234”," Acc_Name ":"2012-01-01” ,"
Acc_Region_Code":"CPMNAME1” },
```

The Json string will have the following details.

1. TP_Id
2. Call_Objective
3. TP_Date
4. CP_Name
5. Work_Category_Name
6. Work_Place
7. TP_Id
8. Acc_Name
9. Acc_Region_Code

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-007 To get TP doctor Info

Method Name:

GetTPDoctors

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. String userCode(from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

```
        {"Tables":[{"Rows":[{"TP_Id":"1234”," TP_Doctor_Id":"1” ," Doctor_Code":"DOC0000000025” ,"
Doctor_Region_Code":"REC00000001” },

         {"TP_Id":"5421”," TP_Doctor_Id":"2” ," Doctor_Code":" DOC0000000026” ,"
Doctor_Region_Code":" REC00000002” }]}]}
```

The Json string will have the following details.

1. TP_Id (Ref from tbl_TP-Header)
2. TP_Doctor_Id
3. Doctor_Code
4. Doctor_Region_Code

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-008 to get TP products Info

Method Name:

GetTPProducts

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. String userCode(from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" TP_Doctor_Id":"1”," Product_Code":"PDC0000001” ," Quantity ":"10”},
 {" TP_Doctor_Id":"1”," Product_Code":" PDC0000002” ," Quantity ":" 15”}]}]}
```

The Json string will have the following details.

1. TP_Doctor_Id (Ref From tbl_TP_Doctors)
2. Product_Code
3. Quantity

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-009 to get TP sfc Info

Method Name:

GetTPSFC

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. String userCode(from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" TP_Id":"1234”," From_Place":"Chennai” ," To_Place":"Trichy”},
 {" TP_Id":"5421”," From_Place ":" Trichy” ," To_Place":"Thuraiyur”}]}]}
```

The Json string will have the following details.

1. TP_Id (Ref From tbl_TP_Header)
2. From_Place
3. To_Place

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-010 to Get Speciality Details

Method Name:

GetSpecialityDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" Speciality_Code":"SPC00000001”," Speciallity_Name":"Specaility 1” },
 {" Speciality_Code":" SPC00000002”," Speciallity_Name":" Specaility 2” }]}]}
```

The Json string will have the following speciality details.

1. Speciality_Code
2. Speciallity_Name

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-011 to Get Brand Details

Method Name:

GetBrandDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" Brand_Code":"BRC00000001”," Brand_Name":"Brand 1” },

 {" Brand _Code":" BRC00000002”," Brand _Name":"Brand 2” }]}]}
```

The Json string will have the following brand details

1. Brand_Code
2. Brand_Name

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-012 To Get Product Details

Method Name:

GetProductDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode(from tbl_User_Info),
4. string lastModifiedDate (from tbl_User_Info),
5. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" Product_Code":"PDC00000001”," Product_Name":"Atogla lotion 50g” ,"
Product_Type_Name":"Sales”,”Brand_Code”:”BRC00000001”,”Speciality_Code”:”SPC00000001”,
”Product_Category_Name”:”CREAM”},

 {" Product_Code":" PDC00000002”," Product_Name":" Atogla lotion 100ml”,"
Product_Type_Name":"Sample”,”Brand_Code”:”BRC00000001”,”Speciality_Code”:”SPC00000001”,
”Product_Category_Name”:”Tablet”}]}]}
```

The Json string will have the following product details.

1. Product_Code
2. Product_Name
3. Product_Type_Name
4. Brand_Code

5. Speciality_Code
6. Product_Category_Name

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-013 To Get User Product Mapping Details

Method Name:

GetUserProductDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode(from tbl_User_Info),
4. string lastModifiedDate (from tbl_User_Info),
5. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" User_Code":"USC00000001”," Product_Code":"PDC00000001” },
 {" User_Code":" USC00000002”," Product_Code”:" PDC00000002” }]}]}
```

The Json string will have the following details.

1. User_Code
2. Product_Code

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-014 to Get Doctor Category Details

Method Name:

GetDoctorCategoryDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode(from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

{"Tables":[{"Rows":[{" Category_Code":"CAT00000001”," Category_Name":"Core” },

{" Category_Code ":" CAT00000002”," Category_Name”:" Non-Core” }]}]}

The Json string will have the following doctor category details

1. Category_Code
2. Category_Name

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-015 to get Doctor Details

Method Name:

GetDoctorDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. string regionCodes, (EX. ‘REC000001’,’REC00002’,’REC000003’ (or) ‘REC000001’)
5. string lastModifiedDate,
6. out string result

Return Type and Format:

Json String

{"Tables":[{"Rows":[{" Doctor_Code ":"DOC00000001”," Region_Code ":"REC00000001” ," Doctor_Name":"Senthil”,” MDL”:”145”,” Category_Code”:”CAT00000001”,”Specaility_Code”:”SPC00000001” ,”Customer_Entity_Type”:”DOCTOR”},

{" Doctor_Code ":" DOC00000002”,"Region_Code":" REC00000002”," Doctor_Name ":"Sample”,” MDL”:”1245”,” Category_Code”:” CAT00000001”, ”,”Specaility_Code”:”SPC00000001” ,”Customer_Entity_Type”:”DOCTOR”}]}]}

The Json string will have the following doctor master details

1. Doctor_Code
2. Region_Code
3. Doctor_Name
4. MDL
5. Category_Code
6. Speciality_Code

7. Customer_Entity_Type

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-016 to get MC Doctor Details

Method Name:

GetMCDoctorDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. string regionCodes, (EX. ‘REC000001’,’REC00002’,’REC000003’ (or) ‘REC000001’)
5. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" MC_Code":"CAM0000001”," Doctor_Code":"DOC00000001” ," Region_Code":"REC0000002”},

{" MC_Code":" CAM0000002”," Doctor_Code":" DOC00000002”," Region_Code ":" REC0000003”}]}]}
```

The Json string will have the following doctor master details

1. MC_Code
2. Doctor_Code
3. Region_Code

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-017 to get DCR Details

Method Name:

GetDCRDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" DCR_Date":"2012-01-01”," Flag":"Filed” ," Status":"Drafted”},

 {" DCR_Date":" 2012-01-02”," Flag":" Filed_Rcpa”," Status":" Approved”}]}]}
```

The Json string will have the following doctor master details

1. DCR_Date
2. Flag
3. Status

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-018   to get Chemist Details

Method Name:

GetChemistDetails

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode(from tbl_User_Info),
4. string regionCodes, (EX. ‘REC000001’,’REC00002’,’REC000003’ (or) ‘REC000001’)
5. string lastModifiedDate,
6. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{" Doctor_Code ":"CHE00000001”," Region_Code ":"REC00000001” ,"
Doctor_Name":"Senthil”,” MDL”:”145”,”
Category_Code”:”CAT00000001”,”Specaility_Code”:”SPC00000001”
,”Customer_Entity_Type”:”CHEMIST”,”Last_Visited_Date”:”2012-01-01”},

 {" Doctor_Code ":" CHE00000002”,"Region_Code":" REC00000002”," Doctor_Name ":"Sample”,”
MDL”:”1245”,” Category_Code”:” CAT00000001”, ”,”Specaility_Code”:”SPC00000001”
,”Customer_Entity_Type”:” CHEMIST”}]}]}
```

The Json string will have the following doctor master details

1. Doctor_Code
2. Region_Code
3. Doctor_Name
4. MDL

5. Category_Code
6. Speciality_Code
7. Customer_Entity_Type

If the result is empty proceed further, else show message to the user

TECHNICAL SPECIFICATION – DIGITAL ASSET MANAGEMENT

HD-ED-DA-API-019 Download Digital Asset

Gets the list of digital asset that were uploaded / created for the company code against a give date range and filtering tags

Method Name:

DownloadDigitalAsset

Input Parameters:

1. string hexCode(used to uniquely identified swaas),
2. string companyCode(used to identified which company),
3. string DigitalAssetUniqueCode

Return Type and Format:

- memory stream ?????

HD-ED-DA-API-020 Find Digital Assets

Gets the list of digital asset that were uploaded / created for the company code against a give date range and filtering tags

Method Name:

FindDigitalAssets

Input Parameters:

1. string hexCode(used to uniquely identified swaas),
2. string companyCode(used to identified which company),
1. string tagsToFind (tags with separation)(DOCSPE_SPC0001#DOCSPE_SPC0002#DOCCAT_DOC001# DOCCAT_DOC002)

Return Type and Format:

Array of digital assets with the following values for each digital asset

- string DigitalAssetUniqueID
- string DigitalAssetName
- string DigitalAssetOnlineURL
- string DigitalAssetOfflineDownloadURL
- int SizeOfDigitalAsset

HD-ED-CLIENT-API-021 to get sale product mapping

Method Name:

GetSaleProductMapping

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode(from tbl_User_Info),
4. out string result

Return Type and Format:

Json String

```
{"Tables":[{"Rows":[{"Sale_Product_Code":"PRC1”,"Mapping_Product_Code":"PRC2”},
```

The Json string will have the following doctor master details

1. Sale_Product_Code
2. Mapping_Product_Code

HD-ED-CLIENT-API-022 Insert DCR

Method Name:

InsertDCR

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode(from tbl_User_Info)
4. string doctorVisitData(from Tbl_DCR_Doctor_Visit) (per doctor)
5. string productDetailsData(from Tbl_DCR_Product_Details)
6. string chemistVisitData(from Tbl_DCR_Chemist_Visit)
7. string rcpaDetailsData(from Tbl_DCR_RCPA_Details)
8. out string result

Return Type and Format:

Bool

If the result is false get the out string result value for error.

Example Data

doctorVisitData :

COM00001^USC000001^2012-01-01^ DOC00001_2013-12-31**^2012-01-01^DOC0000001^REC00000001^12:30 AM^Good Doctor ^1^14.5221122^45.232614**

This string generation order should be like this, which is derived from tbl_DCR_Doctor_Visit rows (one by one)

This will be only one row

productDetailsData:

COM00001^ DOC00001_2013-12-31**^** DOC00001_2013-12-31_PDC000001**^PRC0000001^10^Y#**
COM00001^ DOC00001_2013-12-31**^** DOC00001_2013-12-31_PDC000002**^PRC0000002^9^N#**
COM00001^ DOC00001_2013-12-31**^** DOC00001_2013-12-31_PDC000003**^PRC0000003^15^N**

This string generation order should be like this, which is derived from tbl_DCR_Product_Details

This string contains n number of rows which is depends on the product which is entered in DCR screen

Column separation : ^

Row separation: #

chemistVisitData:

COM00001^ DOC00001_2013-12-31**^** DOC00001_2013-12-31_CMC000001_VAVA PHARMACY**^CHC00000001 ^vava pharmacy^10**

#COM00001^ DOC00001_2013-12-31**^** DOC00001_2013-12-31_NULL_VAVA PHARMACY**^CHC00000001^vava pharmacy^15**

# COM00001^ DOC00001_2013-12-31**^** DOC00001_2013-12-31_NULL_VAVA PHARMACY **^CHC00000001 ^vava pharmacy^20**

This string generation order should be like this, which is derived from tbl_DCR_Chemist_Visit

This string contains n number of rows which is depends on the chemist met, which is entered in DCR screen

Column separation : ^

Row separation: #

rcpaDetailsData:

COM000001^ DOC00001_2013-12-31_1^DOC00001_2013-12-31**^**
DOC00001_2013-12-31_CMC000001_VAVA

PHARMACY^PRC000001^10^ ^ #

COM000001^ DOC00001_2013-12-31_2^ DOC00001_2013-12-31^ DOC00001_2013-12-31_CMC000001_VAVA PHARMACY^PRC000001^10^ ^PRC00000002#

COM000001^ DOC00001_2013-12-31_3^ DOC00001_2013-12-31^ DOC00001_2013-12-31_CMC000001_VAVA PHARMACY^PRC000001^10^Atogla^

This string generation order should be like this, which is derived from

tbl_DCR_RCPA_Details

This string contains n number of rows which is depends on the rcpa details, which is entered in DCR screen

Column separation : ^

Row separation: #

Note: string should be in above format if there is any null (or) empty data , leave it as empty , but "^" count should be unique

Company_Code	User_Code	DCR_Actual_Date	Doctor_Visit_Code	DCR_Entered_Date	Doctor_Code	Doctor_Region_Code	Doct
COM00000068	USC000001	01/01/2012 00:00	DOC00001_2013-12-31	00:00.0	DOC0000001	REC00000001	

Company_Code	Doctor_Visit_Code	DCR_Product_Detail_Code	Product_Code	Qty_Given	Is_Detailed
COM00000068	DOC00001_2013-12-31	DOC00001_2013-12-31_PDC000002	PRC0000002	9	N
COM00000068	DOC00001_2013-12-31	DOC00001_2013-12-31_PDC000001	PRC0000001	10	Y
COM00000068	DOC00001_2013-12-31	DOC00001_2013-12-31_PDC000003	PRC0000003	15	N

Company_Code	Doctor_Visit_Code	DCR_Chemist_Visit	Chemist_Code	Chemist_Name	P
COM00000068	DOC00001_2013-12-31	DOC00001_2013-12-31_CMC000001_VAVA PHARMACY	CMC000001		
COM00000068	DOC00001_2013-12-31	DOC00001_2013-12-31_NULL_VAVA	CHC00000001		
COM00000068	DOC00001_2013-12-31	DOC00001_2013-12-31_NULL_VAVA PHARMACY	CHC00000001		

Company_Code	RCPA_Detail_Code	Doctor_Visit_Code	DCR_Chemist_Visit	Sale_Product_Code	S
COM00000068	DOC00001_2013-12-31_1	DOC00001_2013-12-31	DOC00001_2013-12-31_CMC000001_VAVA PHARMACY	PRC000001	
COM00000068	DOC00001_2013-12-31_2	DOC00001_2013-12-31	DOC00001_2013-12-31_CMC000001_VAVA PHARMACY	PRC000001	
COM00000068	DOC00001_2013-12-31_3	DOC00001_2013-12-31	DOC00001_2013-12-31_CMC000001_VAVA PHARMACY	PRC000001	

HD-ED-CLIENT-API-023 Insert DA Itemized Billing

Method Name:

InsertDAItemizedBilling

Input Parameters:

1. String correlationId (the id which is given from **StartSync API)**
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. string daItemizedDetails(from *tbl_DA_Itemized_Billing*) (per DA)
5. out string result

Return Type and Format:

Bool

If the result is false get the out string result value for error.

Example Data:

COM000001^1^USC000001^Senthil1234^REC000001^Trichy^DIV0000001^Div1^01/01/2012^1^1^7^

With extra parameters

This string derived from ***tbl_DA_Itemized_Billing***

One row at a time

HD-ED-CLIENT-API-024 Start Sync

This API Need to be call first when Down Sync/Up sync

Method Name:

StartSync

Input Parameters:

1. string companyCode (from tbl_User_Info),
2. string userCode (from tbl_User_Info),
3. out string result

Return Type and Format:

String

Sample string : ***43f91643-294e-4175-8867-9b5d0c5a9029***

If the result is empty proceed further, else show message to the user

HD-ED-CLIENT-API-025 End Sync

This API Need to be at the end of when Down Sync/Up sync

Method Name:

EndSync

Input Parameters:

1. string correlationId(the id which is given from **StartSync API**)
2. string companyCode (from tbl_User_Info),
3. string userCode (from tbl_User_Info),
4. out string result

Return Type and Format:

Bool.

If the return type is false then read the out string.

HD-ED-CLIENT-API-028 Get Doctor 360

Method Name:

GetDoctor360

Input Parameters:

1. string correlationId(the id which is given from **StartSync API**)
2. string companyCode (from tbl_User_Info),
3. string doctorCode
4. string regionCode(from tbl_User_Info)
5. string userCode (from tbl_User_Info),
6. out string result

Return Type and Format:

string

If the return type is false then read the out string.

Sample String

```
*{"Tables":[{"Rows":[{"Customer_Name":"A
NAHAR","MDL_Number":"00000055","Category_Name":"Non
Core","Speciality_Name":"Ortho","DOB":"01 Jan
1900"}]},{"Rows":[]},{"Rows":[{"DCR_Actual_Date":"03/14/2013"},{"DCR_Actual_Date":"03/02/2013"},{"
DCR_Actual_Date":"02/07/2013"}]},{"Rows":[{"Product_Name":"Bilactam XL
Tablet","Quantity_Provided":0,"DCR_Date":"07/02/2013"},{"Product_Name":"Pentastar D
Capsules","Quantity_Provided":9,"DCR_Date":"07/02/2013"},{"Product_Name":"Bilactam XL
Tablet","Quantity_Provided":5,"DCR_Date":"02/03/2013"},{"Product_Name":"Pentastar
Tablets","Quantity_Provided":0,"DCR_Date":"02/03/2013"},{"Product_Name":"Bilactam XL
Tablet","Quantity_Provided":0,"DCR_Date":"14/03/2013"},{"Product_Name":"Pentastar
Tablets","Quantity_Provided":9,"DCR_Date":"14/03/2013"}]},{"Rows":[]},{"Rows":[{"Chemists_Name":"
NAHAR","DCR_Date":"07/02/2013","PO_Amount":0.00},{"Chemists_Name":"NAHAR","DCR_Date":"02/0
3/2013","PO_Amount":0.00},{"Chemists_Name":"NAHAR","DCR_Date":"14/03/2013","PO_Amount":0.0
0}]},{"Rows":[]},{"Rows":[]},{"Rows":[{"Date":"14/03/2013","Remarks_By_User":null},{"Date":"02/03/20
13","Remarks_By_User":null},{"Date":"07/02/2013","Remarks_By_User":null}]}]}*
```

This JSON string contains 8 tables:

Customer_Name	**MDL_Number**	**Category_Name**	**Speciality_Name**	**DOB**
A NAHAR	55	Non Core	Ortho	36526

Campaign_Name
Test Campaign1

DCR_Actual_Date
03/14/2013
03/02/2013
02/07/2013

Product_Name	**Quantity_Provided**	**DCR_Date**
Bilactam XL Tablet	0	07/02/2013
Pentastar D Capsules	9	07/02/2013
Bilactam XL Tablet	5	02/03/2013
Pentastar Tablets	0	02/03/2013
Bilactam XL Tablet	0	14/03/2013
Pentastar Tablets	9	14/03/2013

Product_Name	**Quantity_Provided**	**DCR_Date**
Atogla	11	01/01/2013

Chemists_Name	**DCR_Date**	**PO_Amount**
NAHAR	07/02/2013	0
NAHAR	02/03/2013	0
NAHAR	14/03/2013	0

Product_Name	**Product_Code**	**MyQty**	**Competitor_Product_Name**	**Comp_Qty**
Atogla	PRC000001	15	BANATAN	15

Product_Name	**Support_Quantity**	**Potential_Quantity**	**Date**
Atogla	15	15	01/01/2013

Date	**Remarks_By_User**
14/03/2013	NULL
02/03/2013	NULL
07/02/2013	NULL

Based on this need to generate the page design (this call when the user is online only)

Batch API Calling

Batch Number	**API name**
Batch 1	HD-ED-CLIENT-API-017,
Batch 2	HD-ED-CLIENT-API-006, HD-ED-CLIENT-API-007, HD-ED-CLIENT-API-008, HD-ED-CLIENT-API-009
Batch 3	HD-ED-CLIENT-API-005, HD-ED-CLIENT-API-004, HD-ED-CLIENT-API-003, HD-ED-CLIENT-API-010, HD-ED-CLIENT-API-011, HD-ED-CLIENT-API-012, HD-ED-CLIENT-API-013, HD-ED-CLIENT-API-014, *HD-ED-CLIENT-API-021*
Batch 4	HD-ED-CLIENT-API-015, HD-ED-CLIENT-API-016, HD-ED-CLIENT-API-018
Batch 5	*HD-ED-DA-API-020 Find Digital Assets* *HD-ED-DA-API-019 Download Digital Asset*