**iScreen - User Stories:**

**Requirement as provided in the iScreen\_iPhone\_wfs.pdf.**

**Note:**

* Considered and prepared the user Stories like we are going to store the collections data and meta data related to the video in server and client will request server and get the response related to the data and use it to show in the UI.
* Initially read all the data from server and store locally and then one certain period of time update the table or database stored locally.
* To initially start with and to make the UI up in one device we can plan to keep all the collections data and meta data of the video locally and in server can keep only the video files broken in small chunk and access it directly.

**Stories:**

1. When application is launched will display the menu screen with

Title : LIBRARY  
 Menu Items: Collections & TV ON Demand

1. When launch the initial menu screen will do the following in background
   1. Read the collections data stored in server - Contains the name of the show, thumbnail picture link, title, season#, episode#, date, network.
   2. Store the details obtained locally in a file / database.
   3. Read the various category of On-Demand TV. For each category get the title, thumbnail picture link, program name, season #, season related thumbnail image link, episode#, date, title, and network, index URL for the season video, index URL for the episode video. This is for all the on-demand video stored in server. Store the details locally as a file / database
   4. Create a thread and activate it periodically, to ping the server to get the collections data and On-Demand video data. Logic should be implemented to identify whether new update is available and take the same and update in the file / data base.
2. On press of TV On Demand menu
   1. Read the category locally stored.
   2. Display the list of category with navigation option (Button with >).
3. On press of navigation option in one of the category list.
   1. Read the program name stored locally
   2. Draw a image view and update with default image.
   3. Provide the navigation option.
   4. Form the list and display.
   5. Start loading the respective images by reading from server as provided in the link.
4. On press of the particular program navigation button
   1. Should read the program name, title, episode#, season# locally stored related to the program.
   2. Draw the image view and display default image.
   3. Provide Add option.
   4. Provide play option.
   5. Provide Add all option.
   6. Form the list and display.
   7. Start loading the respective images by reading from server as provided by link.
5. When the play option is pressed in the season.
   1. Pass the season index URL to the media player.
   2. Media player will parse and then will take the contents and play.
6. When Add option is pressed in the season.
   1. Store the program name, thumbnail picture link, title, season#, episode#, date, network.
   2. Display the collection screen.
7. Add all option is pressed in the season.
   1. Store the entire episode name and other details as provided in the above point.
   2. Display the collection screen.
8. Collection screen look.
   1. Will display the program name locally stored and in collections structure / data base
   2. Display the navigation option.
9. When navigation option is pressed in collection screen.
   1. Draw the image view.
   2. Display the title, episode#, season #, network, date
   3. Provide the play option.
   4. Read the image from the server and display it using the lazy loading principle.
10. When play option is pressed in the collection screen.
    1. Get the episode index URL.
    2. Provide to the Media player.
    3. Media player will take care of reading the content and display.
11. Maintaining 2 databases / structures.
    1. Collection database / structures.
    2. On-Demand TV database / structures.
12. Module to access the database / Structures.
    1. Module will be created to access the database/structures.
    2. From UI it will be GET option to access the data base.
    3. From Web logic it will SET option to access the data base.
    4. The databases / file will be opened initially when the application is launched.
    5. Allow one operation at a time in the database / file (either read or write).
13. Module to access the web.
    1. To get the collection data like program name, episode#, season#, date, network, picture link, URL.
    2. To download the picture in terms of collections as well as TV On-demand.
    3. To get the media data.
    4. To get the TV On-Demand data like program name, episode#, season#, date, network, picture link, Season URL, Episode URL.
14. Parser to parse the XML / JASON data what we are receiving.

**Query:**

1. When adding the program in collections - do we need to add as per the category?

2. What is the time, we can access the server to get the collections and TV On-Demand details and store it locally.