**QueueProcessor Document**

1. **GQPQueueProcessor**

* (void) getDataFormKeyChainAndProcess -

Data is retrieved from KeyChain and saved to database with the help of the function saveDataToQueueDBFromKeyChain.Internet reachability is checked using reachabilityForInternetConnection function.If the internet is reachable,data is retrieved from database using getDataFromDBAndSendToSAPServer function and 7 timers are started by calling start7Timers function to process the data according to the timers set.

* (void) saveDataToQueueDBFromKeyChain -

Data from keychain is converted to array using arrayFromKeychain function.This array is sent to saveInDB function to save the retrieved data in database.

* (void)saveInDB:(NSMutableArray\*)objectArray -

For every item in the object array that has to be saved in database,primarily it is checked if the item already exists in the database with the help of checkAndDeleteRecordsAlreadyExistsDB function,if so,it is deleted.Details regarding the items are set by inserting a new object for the entity QueuedProcess, for every item to be saved.After all the items are saved,deleteItemsFromKeyChain function is called to delete the items from KeyChain.

* (void)checkAndDeleteRecordsAlreadyExistsDB:(NSString\*)entityName forRefID:(NSString\*)objectID andApplicationName:(NSString\*)appName-

For the entity entityName,the item with particular ReferenceId and ApplicationName is searched for,and if exists,it is deleted by calling deleteObject.

* (void)getDataFromDBAndSendToSAPServer -

The items of the entity QueuedProcess,with the status Completed or not Null are retrieved and stored in fetchedResultArray array.The variable processItemNumber is initialized to zero.processWebServiceRequestToSap function is called to initiate web service request to SAP.

* (void)processWebServiceRequestToSap -

For every item in the fetchedResultArray,the details of of the item is changed(updating status,incrementing ProcessCount and updating process start time) in the database,by calling updateProcessStartedDetailsInDBForObject function.Web service is called for the object with the help of initializeWebServiceCallForObject function in order to process the object.

* (void)updateProcessStartedDetailsInDBForObject:(QueuedProcess\*)object -

The particular object whose details have to be updated in the database is retrieved from the entity QueuedProcess.The object’s ProcessCount value,which indicates the number of processing attempts for the particular object is incremented.The status of the object is changed to Processing,as it will be sent for process by calling the web service.Object’s processStartTime is initialized with the time when it started the processing.

* (void)initializeWebServiceCallForObject:(QueuedProcess\*)object -

In order to process the particular object,the SOAP web service is called and its response is checked for the following possibilities.

* + - * + If the response message is equal to “E”,i.e. Error,updateProcessStatusInDBForObjectWithReferenceID function is called to update the object’s details to the current status; and updateErrorTableForReferenceId function is called in order to update the error table by including the current object.
        + If the response message is equal to “S”,i.e. Success,updateProcessStatusInDBForObjectWithReferenceID function is called to update the object’s details to the current status.
        + If the response has an errorResponseMessage, updateProcessStatusInDBForObjectWithReferenceID function is called to update the object’s details to the current status; and updateErrorTableForReferenceId function is called in order to update the error table by including the current object.

The function processWebServiceRequestToSap is called to update the object’s details and to start the web service for the object.

* (void)updateProcessStatusInDBForObjectWithReferenceID:(NSString\*)refId withStatus:(NSString\*)status -

The object with Reference ID refId passed as the parameter is retrieved having the entity QueuedProcess.The retrieved object’s status is updated with the status value passed as the parameter.The object’s endTime field is initialized with the current time.

* (void) updateErrorTableForReferenceId:(NSString\*)refId andErrorObject:(NSMutableArray\*)responseObject andErrorMessage:(NSString\*)errorMessage -

The object with Reference ID refId passed as the parameter is retrieved having the entity QueuedProcess.The object’s processCount value is retrieved.If its value is greater than 7,insertNewErrorRecordInDbWithRefID function is called to add this object as a new error record.

* (void)insertNewErrorRecordInDbWithRefID:(NSString\*)refID andCoredatObj:(QueuedProcess\*)coreDataObj errorType:(NSString\*)errorType andEroorDesc:(NSString\*)errorDesc -

A new error record for the particular object is added to the database using this function.But before doing so,it is checked if the object with the Reference ID already exists by calling the function checkAndDeleteRecordsAlreadyExistsDB,if exists,the object is deleted.Then a new object for the entity ErrorLog is created and all the properties of it are assigned with values,hence creating an error record for the object.

* (void) start7Timers -

The objects are retrieved for the entity QueuedProcess with processCount = 1,and these objects are stored in the array processCountOneArray.If the array is not empty,a timer is set with 1 minute interval,and when the timer fires,processDataInQueueTableToSapWebServer function is called to process the objects in the array.

This process is repeated 6 times as 7 timers are set altogether.The 7 timers’s interval are 1 minute,10 minutes,1 hour,4 hours,1 day and 7 days.

* (void)processDataInQueueTableToSapWebServer:(NSTimer\*)timer -

When a particular timer is fired,the objects with the respective timerInfo are retrieved and saved in tempArray.These objects are passed to updateProcessStartedDetailsInDBForObject function to update the objects’s process started details;and also sent to initializeWebServiceCallForObject function to initialize a web service call for the object in order to process the object.

1. **GQPViewController**

The view part of the Queue Processor app is taken care in this file.

* (void)viewDidLoad -

When the view is loaded,this function is triggered and the title for the screen is set as Queue Processor.Internet reachability is checked for,when changed,reachabilityDidChange function is called.Function initializeVariables is called.

* (void)viewDidAppear:(BOOL)animated -

When the view appears,this function is triggered.It calls the saveDataToQueueDBFromKeyChain function from GQPQueueProcessor file,to save data onto the database from the KeyChain.The getDataFromDBAndFillDatasource function is called to retrieve data from the database.

* (void)initializeVariables -

This function in-turn calls the getDataFromDBAndFillDatasource function is called to retrieve data from the database.

* (void)reachabilityDidChange:(NSNotification \*)notification -

When the internet reachability is changed,this function is triggered.While internet is reachable,getDataFormKeyChainAndProcess function from GQPQueueProcessor file is called to process the data in the keychain by calling the web service.

* (void)getDataFromDBAndFillDatasource -

The objects are retrieved from the database of the entity QueuedProcess,and stored in the array dataSourceArray.After which,the table displayed in the view is reloaded by calling reloadData to display the queued item details.