

PY CRM

Business Requirements

1. Retailer should be able to capture his customers details as following
 - a. Name (100 char)
 - b. Phone number
 - c. Age (Radio button for age range)
 - d. Date of Birth (calendar)
 - e. Anniversary date
 - f. Ethnicity (drop down menu)
 - g. Gender (radio button)
 - h. Email Id (Email id check should be enabled)
 - i. address (256 char)
 - j. Referred by (Phone number)
 - k. Remark (256 Char)
 - l. Set transaction message
 - m. Choose sender id
2. It should auto start
3. It should be placed in right side corner
4. Close button should be configurable
5. It should be downloadable from website
6. It should be auto installable
7. If uninstalled, it should ask for feedback
8. It should be able to upgrade
9. It should be able to send notification
10. All field should be configurable
11. User can download from website
12. It is free basic version
13. Getpy.biz and SMS signup should happen from same email id to transfer the SMS balance

Technical Requirements

1. Website should support Azure2C signup
2. It should support forgot passwd
3. It should support change passwd
4. It should send package details to download the executable with license file built in
5. Notification can be pushed to application

Security Requirements

1. Merchant-id should not be seen
2. File should not be pushed without authentication

Work Flow Requirements

1. Retailer should be able to sign up the page with his email id

2. Once signed in and ask for Executable, email will be sent with package detailer and installation instruction
3. Video link will be also part of email
4. Video link will have, installation details and setting details
5. If Retailer un-install the app, Feedback form should be pop-up and it should be collected

Screen Shot Requirements

1. It should be very sleek
2. It should use standard colours and fonts of power BI dashboard
3. It can't be closed

Settings Requirements

1. Each field should be configurable and Screen should show only configured fields
2. Field can be marked as mandatory or optional. This should also be configurable using settings tab

DASHBOARD Requirement

1. Retailer should be able see Customer details captured using PY CRM
2. Retailer should be able to upload Customer details
3. Retailer should be able to edit customer details
4. Dashboard should work with same login /passwd as "getpy.biz" sign up page
5. Campaign feature should be enabled
6. SMS can be bought from sms.getpy.biz using one more time sign-up
7. SMS API Key and Sender ID should be configurable from Dashboard
8. Dashboard should provide help on how to configure API Key and Sender

Marketing Requirements

1. Bi-weekly email to get in touch with customer
2. Social media write
3. Social media ads
4. Send email campaign to retailer on download
5. Track download
6. Track email opened
7. Track dashboard usage and data collected

Testing Requirements

1. It should work when internet is not there
2. Setting option should be tested for various configuration
3. Un-install should ask for feedback form and it should come to our server
4. Next version should be seen
5. Uploaded version should be installed

Security Requirements

Work-Flow

- Fill the form and hit submit button at www.getpy.biz
- Thank you message by mentioning email to download link. Download file link generation

Make ZIP file

- Zip PYCRM, PY uploader, PYLICENSE
- Copy zipped file to storage account
- Rename file name with generated key

Generate License File

License File will have following JSON Object

- Merchant id (Get Max Value of Merchant id))
- Generate SAS Token Using Rest API : Generate SAS Token (<http://sudiptachaudhari.com/azure-sas/>)
- Encrypt License File

Generate SAS Token Using Rest API design

- **Function: string pycrmProcessSASTokenSteps(int merchant_id)**
This is main function which will create container for given merchant id. Nomenclature of container will be "pycrm-" + merchant_id. For example for merchant id 24, container name will be pycrm-24. This function will call following functions

```
String pycrmProcessSASTokenSteps( int merchant_id)
{
    String containerName = pycrmGetContainerName( merchant_id);

    pycrmQueueCreateContainer( string containerName);
    pycrmBlobCreateContainer ( string containerName);
    string sasToken = pycrmGenerateSASToken ( string containerName);
    pycrmStoreSASToken ( merchantid ,containerName, sasToken);
    return sasToken;
}
```

- **Function : string pycrmGetContainerName (int merchantName)**
{
 string ContainerName = "pycrm-" + merchantId;

```

try
{
    return ContainerName;
}
catch (Exception e)
{
    return "-1";
}
}

```

- **Function : pycrmQueueCreateContainer(string containerName)**

```

pycrmQueueCreateContainer ( string containerName)
{
    try
    {
        CloudStorageAccount storageAccount =
        CloudStorageAccount.Parse("DefaultEndpointsProtocol=https;AccountName=Accoun
tName;AccountKey=Key");
        // Create the queue client
        CloudQueueClient queueClient = storageAccount.CreateCloudQueueClient();
        // Retrieve a reference to a queue
        queueContainer = queueClient.GetQueueReference(CONTAINER_NAME);
        // Create the queue if it doesn't already exist
        queueContainer.CreateIfNotExists();
    }
    catch (System.Exception e)
    {
        Log(" Queue Not able to connect to Azure");
    }
}

```

- **Function : pycrmBlobCreateContainer (string containerName)**

```

pycrmBlobCreateContainer ( string containerName)
{
    try {
        CloudStorageAccount storageAccount =
        CloudStorageAccount.Parse("DefaultEndpointsProtocol=https;AccountName=AccountName;
AccountKey=Key");
    }
}

```

```

        // Create the blob client.

        CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();

        // Retrieve a reference to a container.

        container = blobClient.GetContainerReference(containerName);

        // Create the container if it doesn't already exist.

        container.CreateIfNotExists();

        container.SetPermissions(new BlobContainerPermissions { PublicAccess =
BlobContainerPublicAccessType.Blob });

    }

    catch (System.Exception e)

    {

        Log(" Blob Not able to connect to Azure");

    }

}

```

- **Function : string pycrmGenerateSAS token (string containerName)**
{
 //Generate SAS token for 5 years time line
 // should have create permission on blob and queue

}
- **Function : pycrmStoreSAS token (int merchantid , string containerName, string sasToken)**
{

 // Save merchant id, containername, and sas token tuple in DB

}

Generate Email Content

- Captured user information has to be stored in database with following field. Each user will be identified using user id

- Name
- Email
- Phone number
- Sector (drop down menu)
- Address (Country/State/City) Dynamic
- Captured Lat/Lng
- Captured system
- Date (Automatic)
- Send Email with following information
 - Zipped File which will have PYCRM.EXE, PYSERVER.EXE, License files for PYCRM and PYSERVER
 - Video for Setup
 - Document for Setup
- Follow up Email Contents
Follow-up email content based on user activity ??????

Installing Package

- Unzip the package in some folder
- Check for dependencies ??? How to check this
- Detect if system is 32 bit or 64 bit (DO we need this ???)
- Make sure licence file is copied properly for PYCRM and PYSERVER

Deleting Zip File

- Webjob to delete Zip File after 7 days of creation

PYSERVER Modification

1. De-crypt licence file to get merchant id
2. While sending the file If key does not match with expected value delete the file and don't send it.
3. send mac address as part of log to detect if it getting used by two computers, locking can be done later for storage account
4. send log file during start-up for record keeping

Upgrade and Notification

- Build service to send Notification
- Build service to upgrade the software

Revoking SAS Token

In unforeseen condition SAS Token will be revoked to enable this functionality we will maintain one table which maps with SAS Token and Merchant-id

