Program will require a login. That will include username, password and encryption key. The encryption key is used to encrypt and decrypt all information in the database. The user will chose that the first time they open the program and it can't be changed.

That encryption key won't be stored anywhere. That's why they enter it each time.

Once open, the user will have a main screen with a few options. First option is to "Get current charges". That will run the query I give you later and pull the current customer charges into a table in this program and display those in a listview.

Only one user for the program. When it opens the first time, it should sayd "Choose username" Choose password" and "Enter Encryption Key". You should then use that encryption key to store the username and password encrypted in the database and everything else that's stored.

Me or whoever else opens the program the first time. I am going to be putting this on 4 computers because there are currently 4 people that pull charges from other programs and need to process them.

yes. They'll each be running on their own database. So actually we should a setup form where we choose the local database for this program and a second where we choose the remote database to pull charges and a text box where we store the query.

On main screen after you get current charges, we need to display that with customerid, name, amount due and a checkbox or button on each one to "Approve" the charges. This is so the person running the program can confirm amounts due before they automatically charge the customer.

user would need to change it as they are manually confirming the amounts due.

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

I forgot. There should be a List Customers option on the main screen as well that brings up a list of customers. On that there should be a Add Customer button and edit customer button. This is where the user will add people in.

The fields for this would be customerid, firstname, lastname, street1, street2, city, state, zip, country, phone.

As well as payment fields for card\_type, card\_number, exp\_month, Exp\_year, ccv\_code.

ok. Back to charges list. After the user approves the charges. they should be able to hit a button to "process all charges". This will then go one by one and submit the amount due to Authorize.net and store the result in the table.

It would be good if the user could see this "Batch" of charges in the list as Green highligh or red highlight if they are approved/denied in addition to it saying approved or denied.

Also in that list of get charges, if a customer's payment expiration is expired, it should show it somehow so they can update the information before running the "process payments" function.

ok. Now once the charges go through, if there are any that are still pending because they were denied, the user can then go back and update the card information on the customer's record and re-run the process to collect the charges. It should then only charge the ones that have not been completed yet.

Then once that Batch is complete there should be a way for the user to "Close Batch". This will remove it from the screen so that it's ready for the next batch to be pulled next month.

yes. When you pull the current charges from the existing system using my query, it creates a "batch" in this program. That's the current list of charges that need to be paid.

To make it easier to view for the user, when you click "Get current charges" perhaps it could pop up with a box that says "Enter name for this batch" then the user can call it something like "December 2018" so that they can run a print out a report that shows Batch Name , Amount Charged, etc.

The last thing would be to just print a couple reports. Report one would be "Print Batch". This will print out the list of customers, amounts and approval for the batch you just ran.

The 2nd would be to veiw previous batches so you could list the batches (we just described above) and select one of those to print.

3rd would be a customer report. That would be customerid, name, card type, last 4 digits of card, expiration date.

That should be it. there may be mods needed in the future but we can add to new project if need be.

Bottom line is, we need to collect amounts due from that query and charge all the cards we store on file. And then be able to show a print out of that or be able to view past charges to see that they were approved.

You can use Right Clicks on listviews to make it easier to edit items or approve, etc. Whatever you thin will look and work best.

ok. So you will connect this program to a local sql server database on the hosting server. That's for the program itself. The charges are being pulled from another server in the network that's live in another program. My thought was to let the user choose the database locations. Exmple:

when they open for first time, it should bring up form to connect to database for program. User enters name or ip of the sql server, database name, username and password. The second connection would be to the live database that has the charges on it. The user should be able to enter name/ip of that server, that database name, username and password to connect to it and the user can then also enter the query to pull the current charges from it. This way if that database format changes in any way, the user can update the query so that it's always bringing back the information to you in the same recordset format.

instead of hard coding the db connections into your program, just store it locally and have it updatable.

maybe a config file in the program directory. However, we should encrypt db password.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | |  | Here's the Authorize.net sandbox account information: https://sandbox.authorize.net/ |  | 8:15: AM AEST |  | |  | | | | | |
|  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | |  | user: bardentech2018 |  | 8:15: AM AEST |  | |  | | | | | |
|  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | |  | Password: Freelancer!23 |  | 8:15: AM AEST |  | |  | | | | | |
|  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | |  | API Login ID: 3Jx92DvP Transaction Key: 586NKzy239Z7tCLU Key: Simon |  | 8:16: AM AEST |  | |  | | | | | |
|  |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | |  | If you log into that site, it will have more examples for testing. We'll need to add a form on the app to enter authorize.net api login and transaction key so we can change it when we go live. |  | 8:16: AM AEST |  | |  | | | | | |

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

$200 released once IIS installed on my server and site configured so that main page is responding and connecting to the databsae you setup locally.

$300 more released when..

the basic functionality is working where buttons open the lists, for each of the items I gave you and customer add/edit is working.

$400 more when all functionality is working including sending test charges to authorize.net sandbox.

$100 remaining when everything fully working and complete.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

We have members that pay random amounts each month based on charges accrued. We would like to store their payment information encrypted (PCI compliant) and each month automatically charge all the cards on file the amount due. That amount will be collected from a SQL query and matched by a customerID field in this program.

There are a couple requirements:

1. Main screen: Get Current Charges Button

This will run the query to get all charges accrued last month. I will provide the connection and query that you call. Just pull it into a recordset and display for review to the user.

2. The list that's returned should show CustomerID, CustomerName, Amount Due and a checkbox to Approve to collect. It should also query the table you create to store the payment information. If a payment type is about to expire or has expired it should highlight that customer in red.

3. The main screen should have another button to view customers.

- you will store customerID (which matches our other program), first name, last name, address, city, state, zip, country, phone. You should also have a button to add payment onto the customer's file. If a payment type is about to expire or has expired it should highlight that customer in red.

4. The user should be able to double click on a customer in the list also to edit this information or click an Add Customer button to add a new customer.

5. There should also be a Collect Payment button on the main screen. This will then query the list of charges that were "Approved" earlier and create a batch for processing so that these charges are not attempted again. Once the batch is created and logged, it should automatically charge the credit card on file.

6. It should one at a time send a api query to authorize.net with the card information and amount. It should then take the return (approved or denied) and store it in the batch log. When it's approved, it should tag that item in the batch as "Completed". If one card in the batch is denied it should note that the batch is not complete. The user can then update the card on file for the customer and be able to press another button on the main screen that says complete unfinished batch.

You get the point. Bottom line is that each month we'll do a single query to pull all charges for our customers. It should import into a batch for processing. It will then run the batch against all the cards on file (one card per person). If approved, note that it's approved. If denied mark the return reason then mark the batch Incomplete and have a way to show the incomplete batch and the cards that were not completed.

The last thing is there should be a report to run to pull up a batch report showing all payments made and totals. A couple other helpful reports should also be created like a customer list, etc.

All information including names and card numbers should be stored encrypted in a sql server database. The encryption key should be requested at login. When a user logs in they should be asked for username, password and encryption key. You must use at least the following encryption level:

AES – 128 bits or higher

TDES/TDEA – triple-length keys

RSA – 2048 bits or higher

ECC – 224 bits or higher

DSA/D-H – 2048/224 bits or higher

Please ask any questions you have before bidding. No surprises once we begin. Need it completed in 2 weeks.

https://developer.authorize.net/api/reference/features/oauth.html