|  |  |
| --- | --- |
| WYSIWYG | Synprod |
|  |  |
| 12/09/2017 | Project document |
|  | This document contains all the technical and decision of the Synprod project. |

Synprod

Project document

Table of content

Table des matières

[INTRODUCTION 1](#_Toc497993742)

[Objective 1](#_Toc497993743)

[Core Functions 2](#_Toc497993744)

[Description of Core Functions 2](#_Toc497993745)

[1. Client sign up 2](#_Toc497993746)

[2. Client Account 2](#_Toc497993747)

[2.1. Type of Client Account 2](#_Toc497993748)

[2.1.1. On premise 2](#_Toc497993749)

[2.1.2. Dolibarr clients online 2](#_Toc497993750)

[2.1.3. Online clients using software hosted by third party Vendors 3](#_Toc497993751)

[3. Payment Gateway 3](#_Toc497993752)

[4. Connector download 3](#_Toc497993753)

[4.1. Receive notification about packages going to expire. 3](#_Toc497993754)

[Technical requirement 3](#_Toc497993755)

[Use case for On premise clients 4](#_Toc497993756)

[Design 6](#_Toc497993757)

[Entity Relationship Diagram 6](#_Toc497993758)

[Entities 6](#_Toc497993759)

[Flow chart synprod 7](#_Toc497993760)

[The Connector 7](#_Toc497993761)

[Design of the Connector 9](#_Toc497993762)

[Creating the DSN for a user in VB 9](#_Toc497993763)

[Life cycle of the Connector 10](#_Toc497993764)

# INTRODUCTION

## Objective

The project’s aim is to allow the company to allow clients to use its connector by downloading it and using it to generate production reports on their premises. Therefore, by making the connector available on the internet, this project will lead to huge cost saving in terms of Transport, troubleshooting and on the other hand it will generate profit by being able to sell globally.

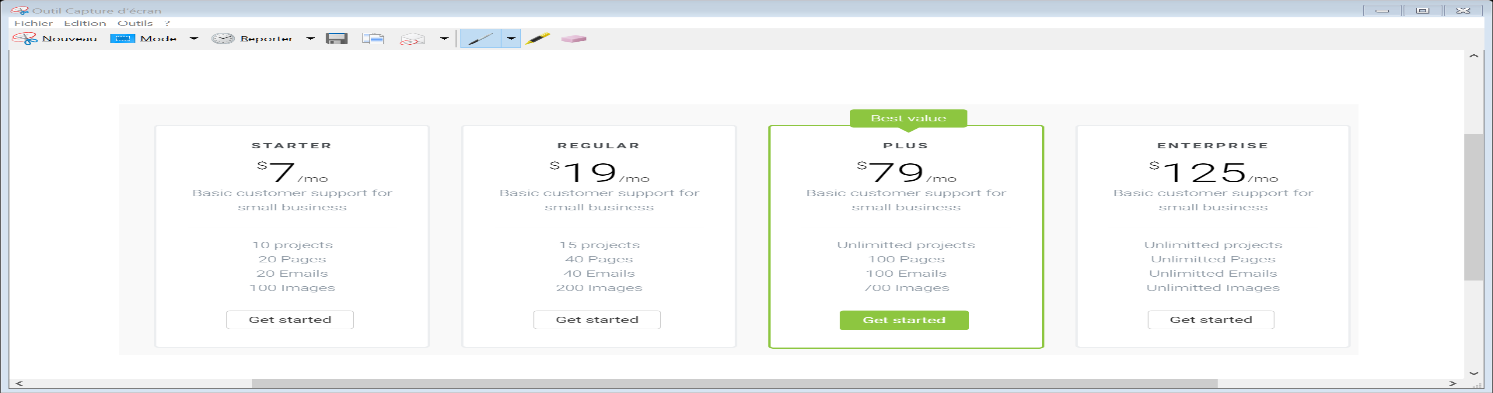
### Core Functions

1. Client sign up
2. Client Account
3. Payment Gateway
4. Connector download
5. Admin panel for our company

#### Description of Core Functions

##### Client sign up

When the Client will visit the website. He will be presented to various packages (services) as shown below



When he clicks on one of the packages he will be brought to a login page. If he has an account he will input his username and password else he will be force to sign up.

##### Client Account

The client will have a profile page where he will be able to :

1. Change his password
2. Change his details
3. Update his wallet information
4. See a history of the packages he purchased
5. Receive notification about packages going to expire.

##### Type of Client Account

We shall have 3 types of Clients

##### On premise

* 1. The Client will download the connector and will use it on his premise. No interaction with Synprod will occur. This type of client can be considered as a ‘basic’ client
  2. The Client will download the connector and will have access to our online service on Synprod. Each module online is accessible by activating the module ( paying for the module)

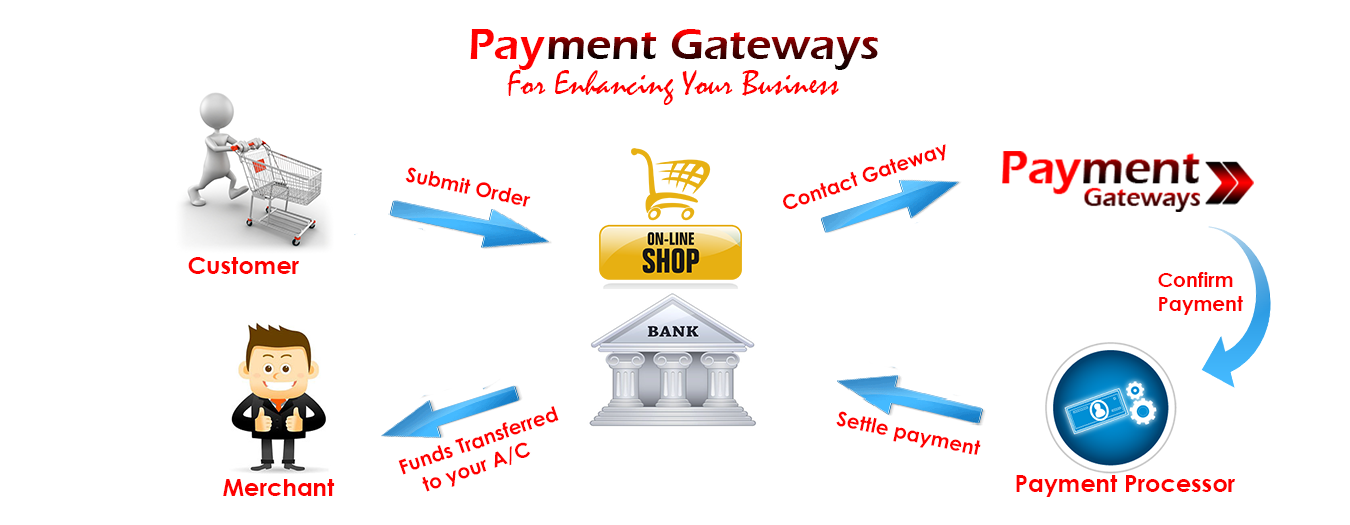
##### Dolibarr clients online

1. These clients will connect on our platform to use their dolibarr CMS

##### Online clients using software hosted by third party Vendors

##### Payment Gateway

Clients will be able to make purchases using several payment gateways like PayPal, Credit cart etc. The processes will look like below:



Synprod

##### Connector download

When the user payment is done he will get a unique link to download the connector. The issue might be that he shares the connector to his friend running a similar system as him. This will be a lost for us.

Update 08/11/2017

The connector will be a desktop agent which will synchronize the data from the client site to our server. Basically it will synchronize data from ( mysql, ms sql and other databases) to our servers.

##### Receive notification about packages going to expire.

Connectors are not life time, they will need to be renewed. There will be an expiry date inside the main engine of the controller.

# Technical requirement

The Synprod will be run on

Apache 2.4.23

Php 5.6.25

Mysql 5.7.14

Symfony 3.1

<https://github.com/wysiwyg-team/synprod/tree/master/web>

Front End

Bootstrap 3.0

Css 3

Jquery

Font awesome.

Version control

Git

Repository : <https://github.com/wysiwyg-team/synprod.git>

Dev url : <http://dev.syncprod.fr/>

Use case for On premise clients:

The on-premise clients will register and selects a package. The package normally will allow him to download a connector. This connector will perform its jobs at the client’s premise. No interaction what so ever with our systems.

Hence the main functions will be:

Login/ Logout

Buy a module

Client Account

Download a package. (will be able to download 3 times only)

# Design

In this section, we shall describe in detail the design of the various components

## Entity Relationship Diagram

Synprod

Client

Admin

Payment Gateway

Connector

Figure 1 system level Entity Diagram

## Entities

1. Client

There will be 3 types of clients as described in section

The client will perform the following actions

* 1. Login
  2. Logout
  3. Signup
  4. Manage his account
  5. Purchase connector
  6. Raise ticket for issues

1. Admin

The admin will perform the following actions

* 1. Login
  2. Logout
  3. Manage client
  4. Answer issues

1. Payment Gateway

The payment gateway entity will perform the following

* 1. Handle payment request
  2. Validate payment

1. Connector

The Connector entity will perform the following

* 1. Connect to the client database ( sage, cegid etc )
  2. Send credential to our server for authorization
  3. Transfer data from client to our server (daily, hourly.. ) This needs to be configurable on the client account on our site
  4. Connector stops working with our server if customer stop his subscription. Therefore the needs to implement a customer account.

## Flow chart synprod

Client

Login

Buy connector+ package

Connector

Sign up

Not registered

Payment Gateway

# The Connector

Synprod consists of a Connector (desktop Agent ) through which all communication between client and Synprod application will occur. Since we cannot have access to the client database directly we are obliged to pass through a connector.

Connector workflow Diagram

Figure 2 WorkFlow of the connector

Client site

Synprod Application

Sql server

Connector

Synprod database

Workflow descriptions

When a client purchases a package, he will get the connector alongside of the package. This connector will be for a particular task only depending on the package chosen. Before he downloads the Connector, we will have to build it with the configurations of the client.

The client will have to enter the following on this customer account before the package is downloaded.

1. Database name
2. Database path
3. Username
4. Port number

Then the application will “bake” the connector with the predefine values together with a unique Id for reference. The clients can change those settings in a configuration page on the connector.

The client will then press a button “connect to Database” to execute the connection with his Database.

The connector will then enable the synchronize button. This will then transfer data to Synprod Api application. [ for on premise premium clients only]

For on premise clients, the connector will only transfer data to our on premise package for internal use.

For on premise premium clients, the connector will authenticate the user before any service is executed with our Api authentication service. Once authenticated, then it will be able to transfer data and execute tasks.

The Connector is will be packaged into an executable for deployment on windows platform.

## Design of the Connector

ODBC

Open Database Connectivity (ODBC) is a standard application programming interface (API) for accessing database management systems (DBMS). The designers of ODBC aimed to make it independent of database systems and operating systems. An application written using ODBC can be ported to other platforms, both on the client and server side, with few changes to the data access code.

(source: wikipedia.org)

Some of the applications Synprod will sell are odbc compatible, therefore it is important that on premise packages has a DSN ( source name present ) available on the client premise. Therefore the connector needs to allow the user to set up the source name

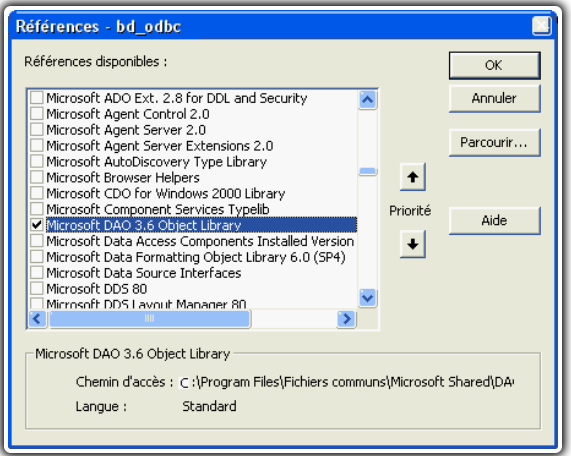
This can be set up as a form where the user needs to enter

1. The database name
2. The driver of the odbc
3. Database Username,
4. Database Password
5. Attributes ( this varies according to the driver of the ODBC )

## Creating the DSN for a user in VB

In visual basic the DSN is created as follows:

First include the library **DAO**



For sql server DSN the code in visual basic is as follows:

Sub RegDB\_sqlsvr()

Dim strDSN As **String**, strAttr As **String**, strODBCDrv As **String**

' Nom de la source de données ODBC

strDSN = "Essai SQLSVR"

' Nom du pilote ODBC

strODBCDrv = "SQL Server"

' Attributs

' UID et PWD ne sont pas acceptés par le pilote SQL Server

strAttr = "SERVER=LZ2" & vbCr & "DATABASE=EMM"

' Si Serveur SQL avec authentification Windows,

' activer la ligne ci-dessous :

'strAttr = strAttr & vbCr & "Trusted\_Connection=Yes"

DBEngine.RegisterDatabase strDSN, strODBCDrv, True, strAttr

End Sub

There are also ways of creating a DSN source in vb for system instead of the current user.

Since we can create the DSN in the connector(Agent), this means that for on premise packages will work provided that the client has the drivers install on this machine.

## Life cycle of the Connector

1. Before packaging a unique id is inserted in the config section of the Connector.
   1. This unique id is saved in the client Table on our Synprod application
2. The connector is packed along with the on premise application
3. The client runs the executable
   1. He is presented with a form where he enters the details (see section ODBC aboved)
   2. The dsn is created and the on premise application can get its data