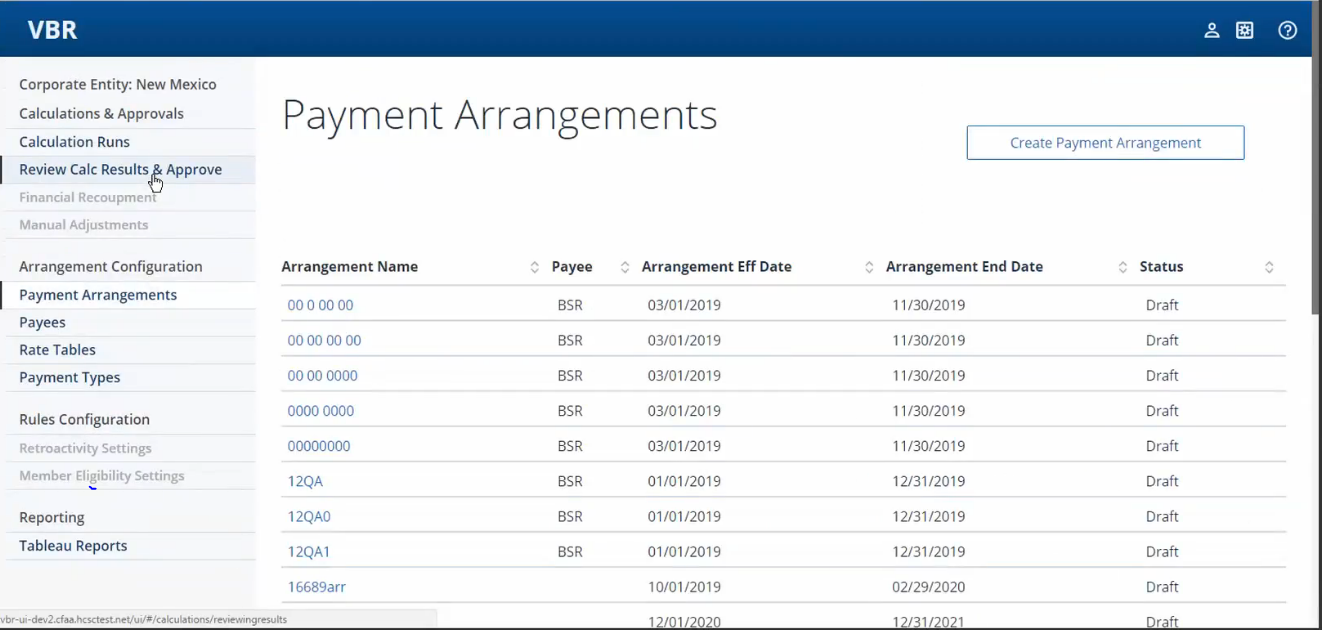
Payment Arrangements Screen



Once we click on any arrangement to open, before opening the given arrangement it will revalidate that particular arrangement.

In that process it will display the warning messages or the messages stating the reason due to which the given arrangement sets to invalid state.

These are possible values of Arrangement status:

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_EXPIRED*** = "EX";

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_FUTURE\_INVALID*** = "FI";

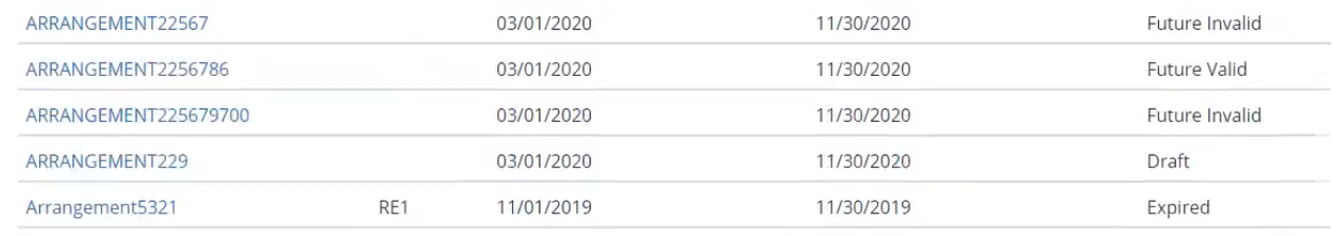
**public** **static** **final** String ***ARRANGEMENT\_STATUS\_DRAFT*** = "DF";

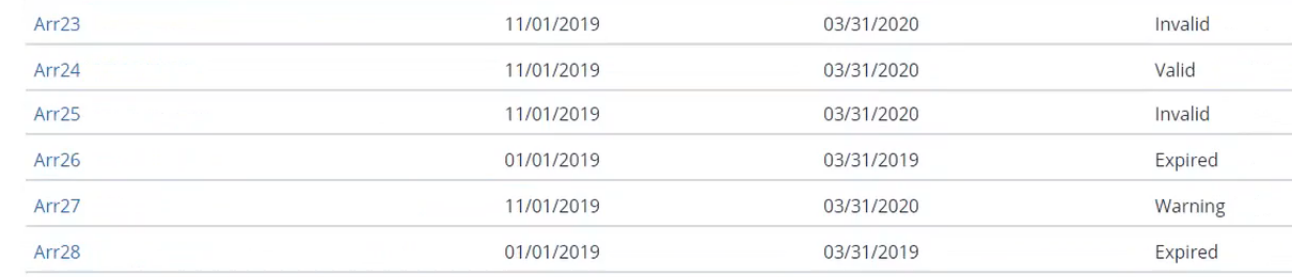
**public** **static** **final** String ***ARRANGEMENT\_STATUS\_INVALID*** = "IV";

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_VALID*** = "VA";

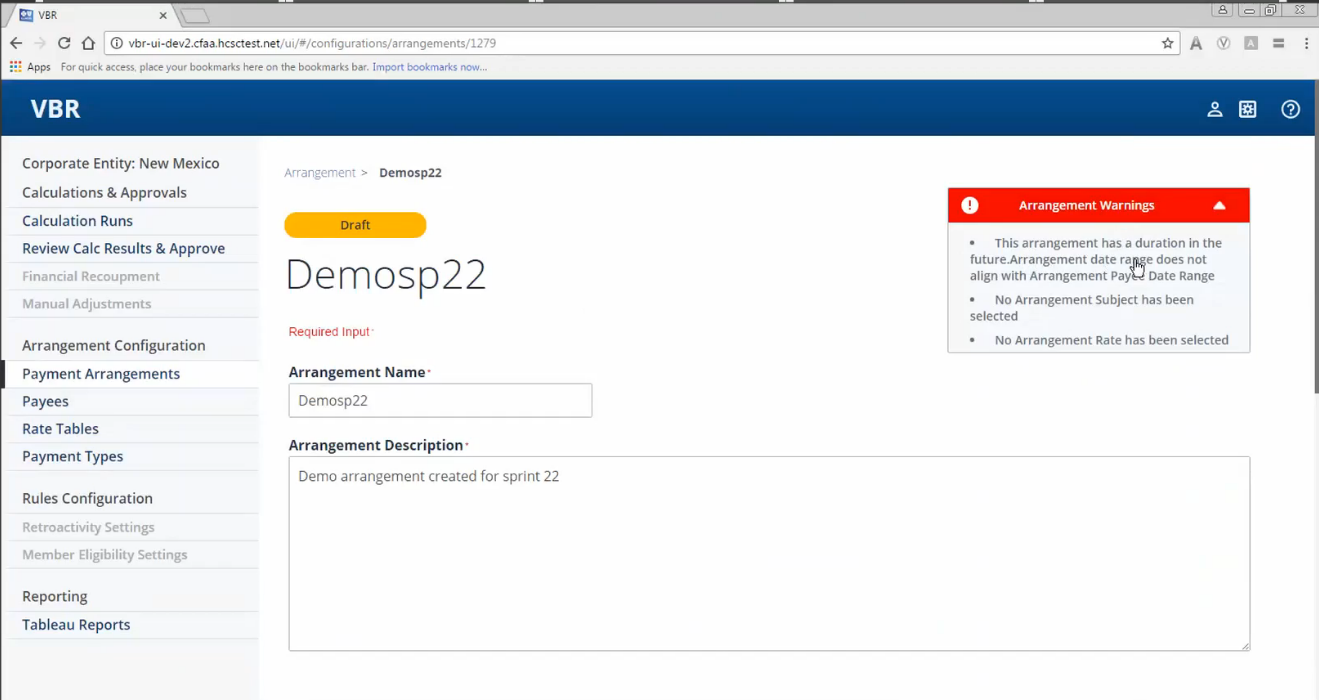
**public** **static** **final** String ***ARRANGEMENT\_STATUS\_WARNING*** = "WA";

Different Status values shown on the Payment Arrangements screen

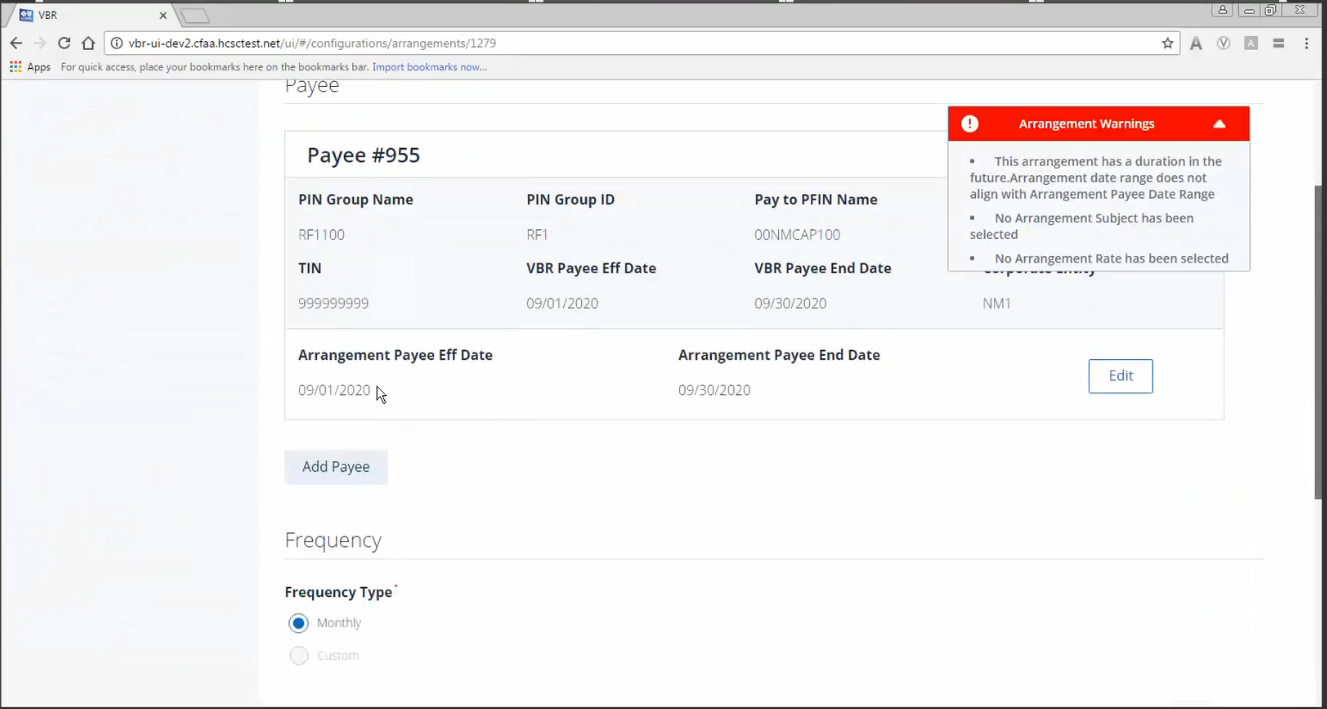




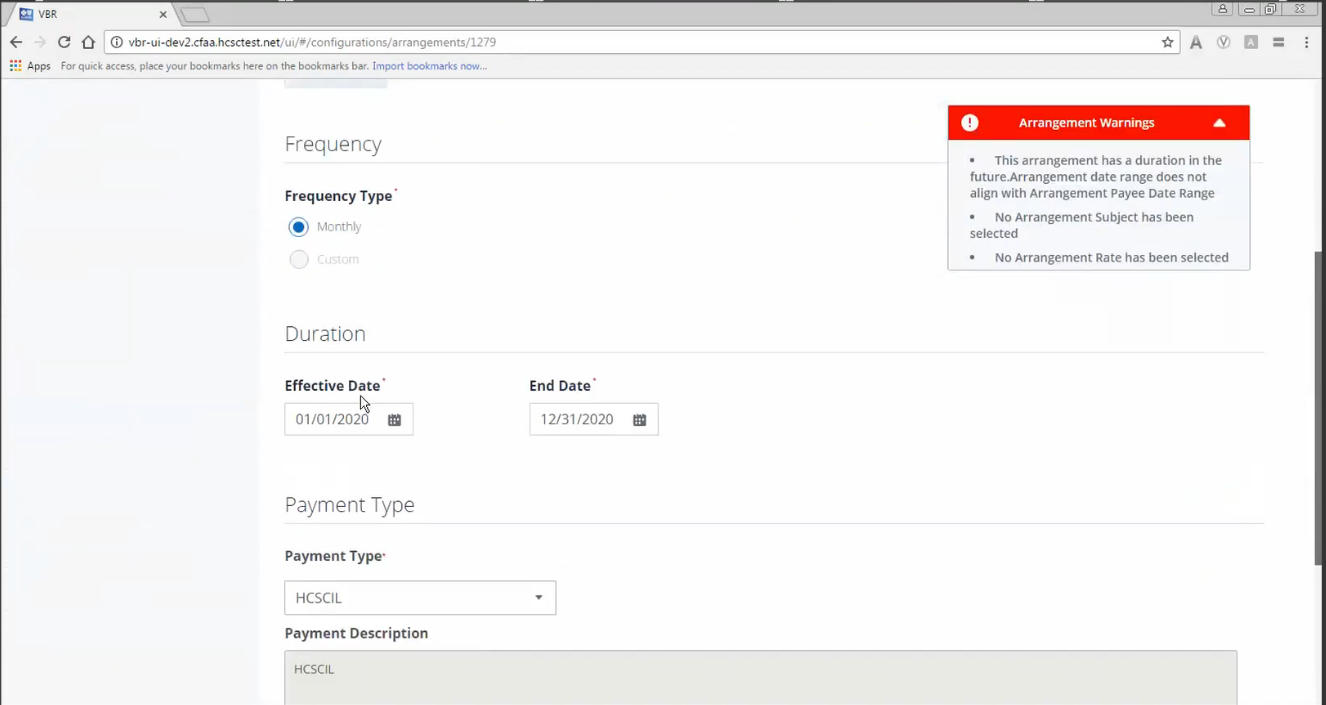
Payment Arrangement Screen



Payment Arrangement Screen (cont…)



Payment Arrangement Screen (cont…)



So these 3 entities:

1) Payment type

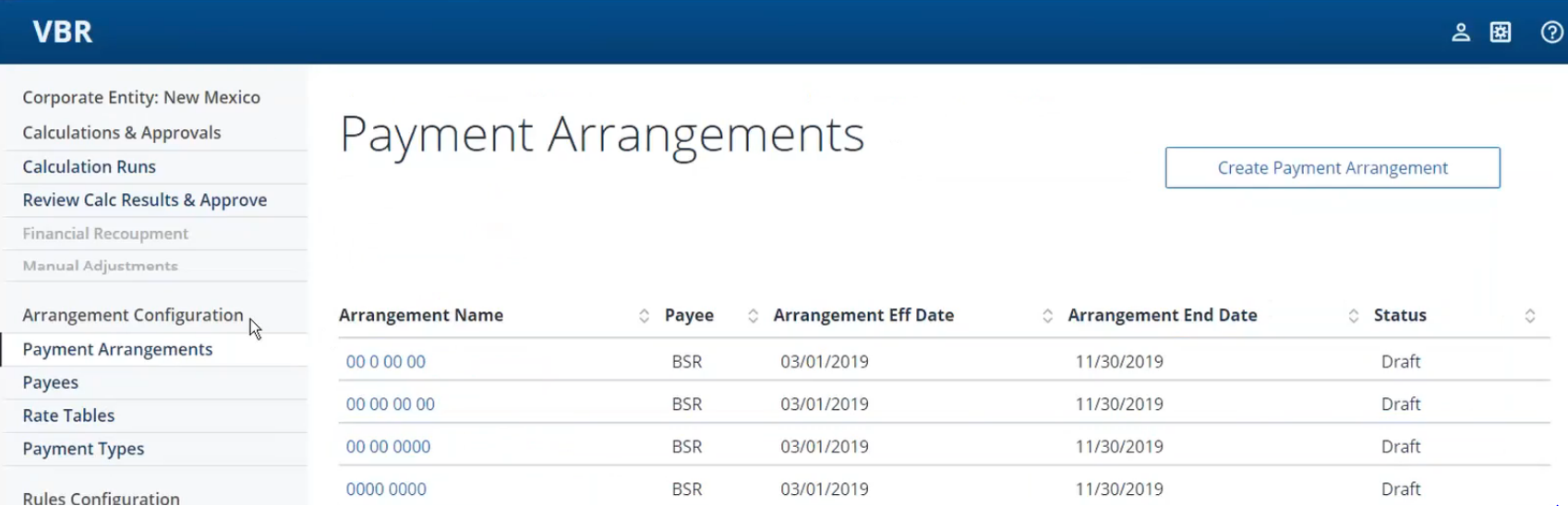
2) Payee

3) Rate table

They have to be created beforehand, these entities/items are then used later for setting up the arrangement.

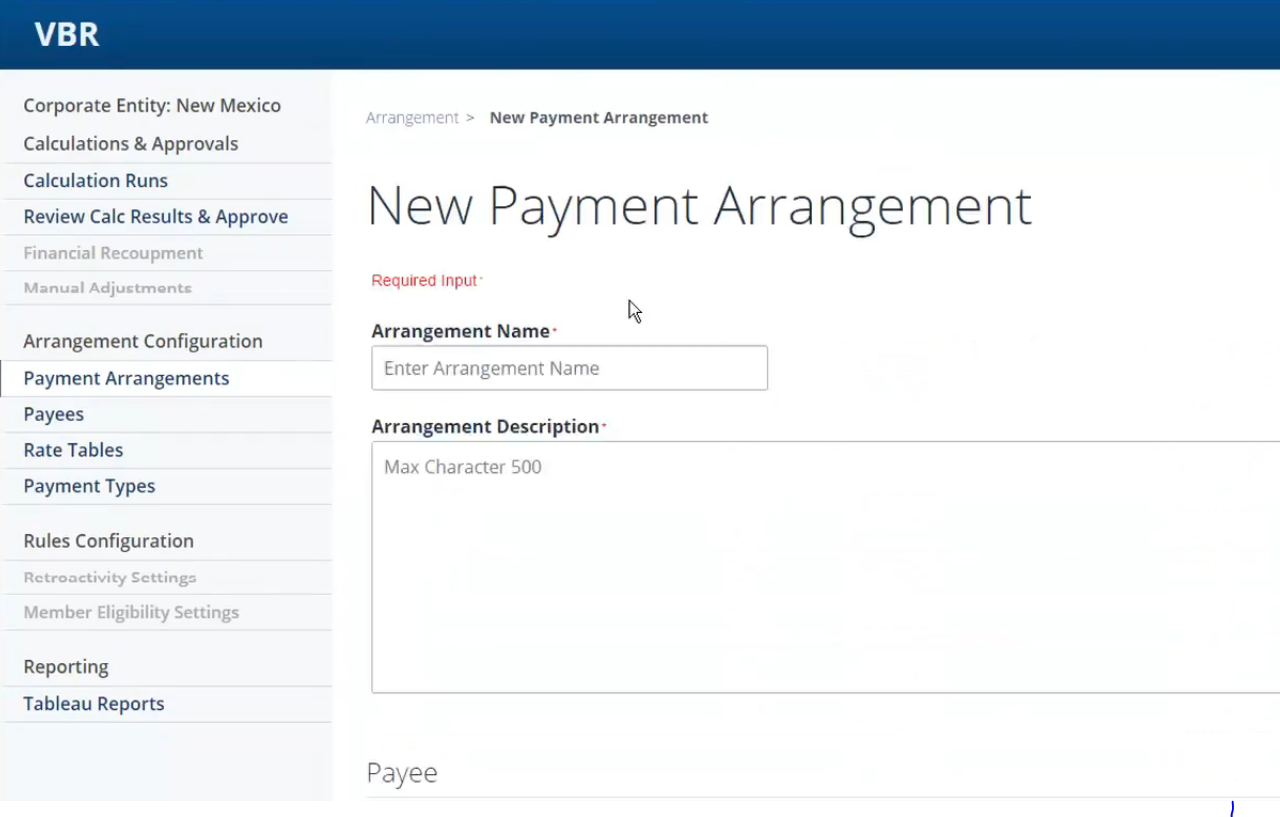
Without Payee and Rate table, we can still go ahead and save the arrangement in the Draft status. However for this still we need the 6 minimum fields to be entered: Name, Desc, Payment frequency, Effective Start& End Dates and Payment type.

Payment Arrangements Screen – Create Payment Arrangement button



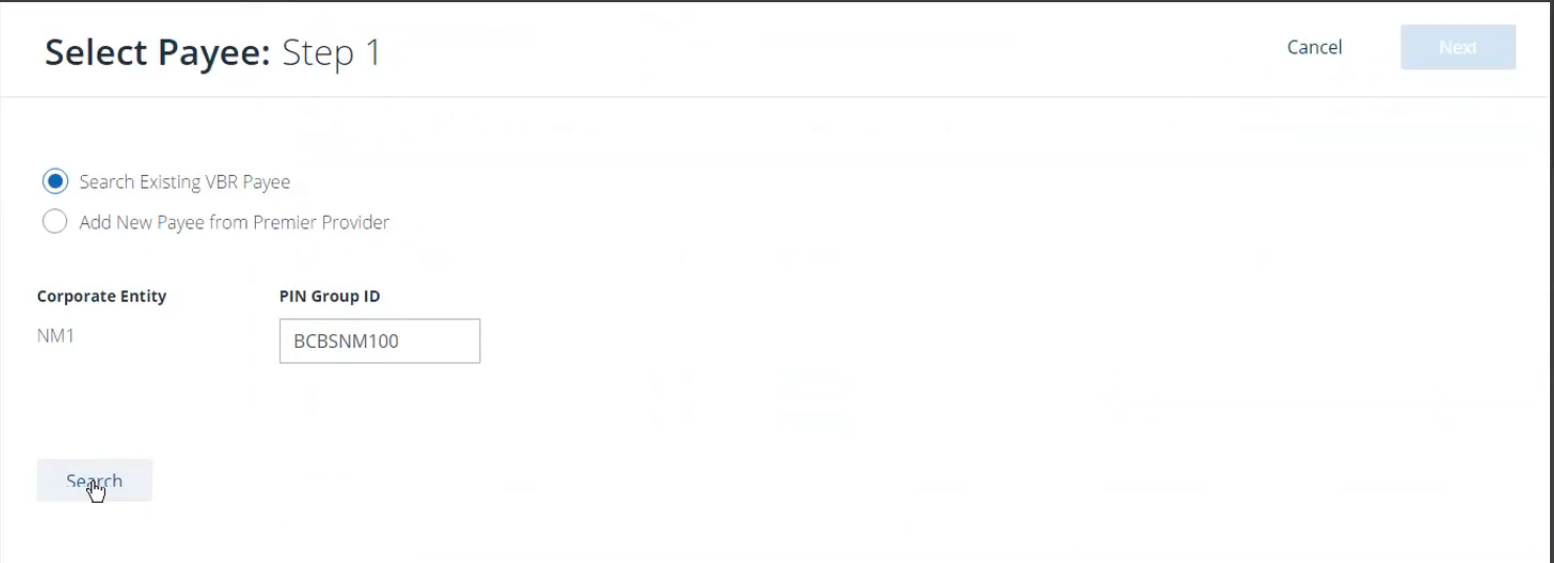
For creating a new arrangement, we click on “Create Payment Arrangement” and get the following screen

New Payment Arrangement screen

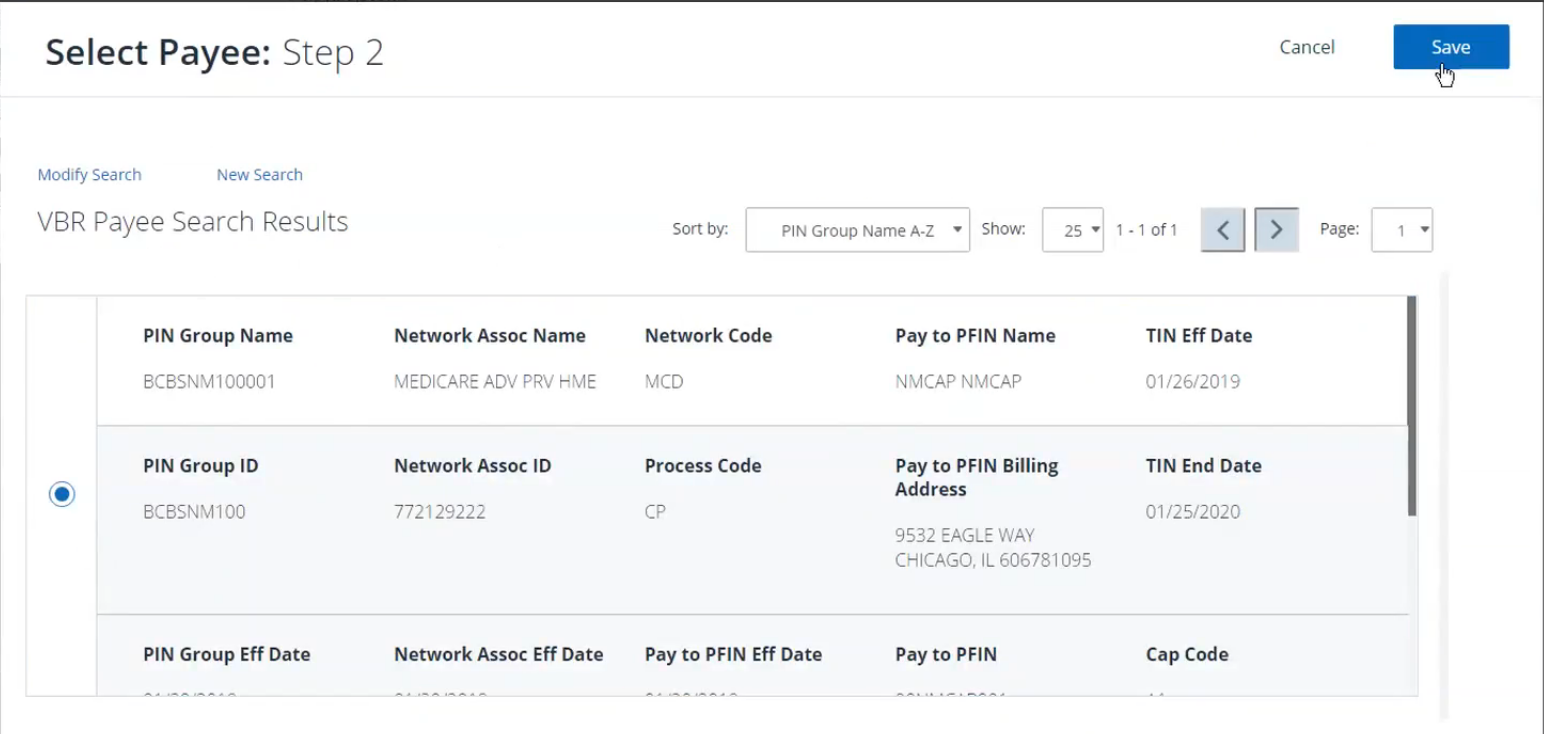


New Payment Arrangement screen (cont…)

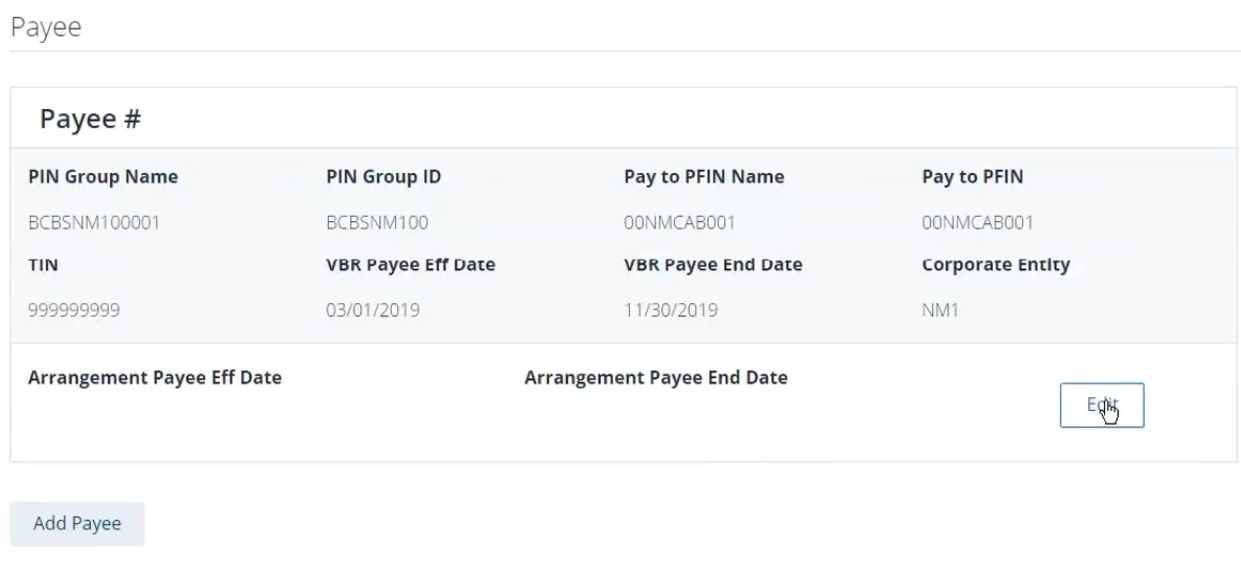
Click on “Add Payee” button



New Payment Arrangement screen (cont…) – adding Payee

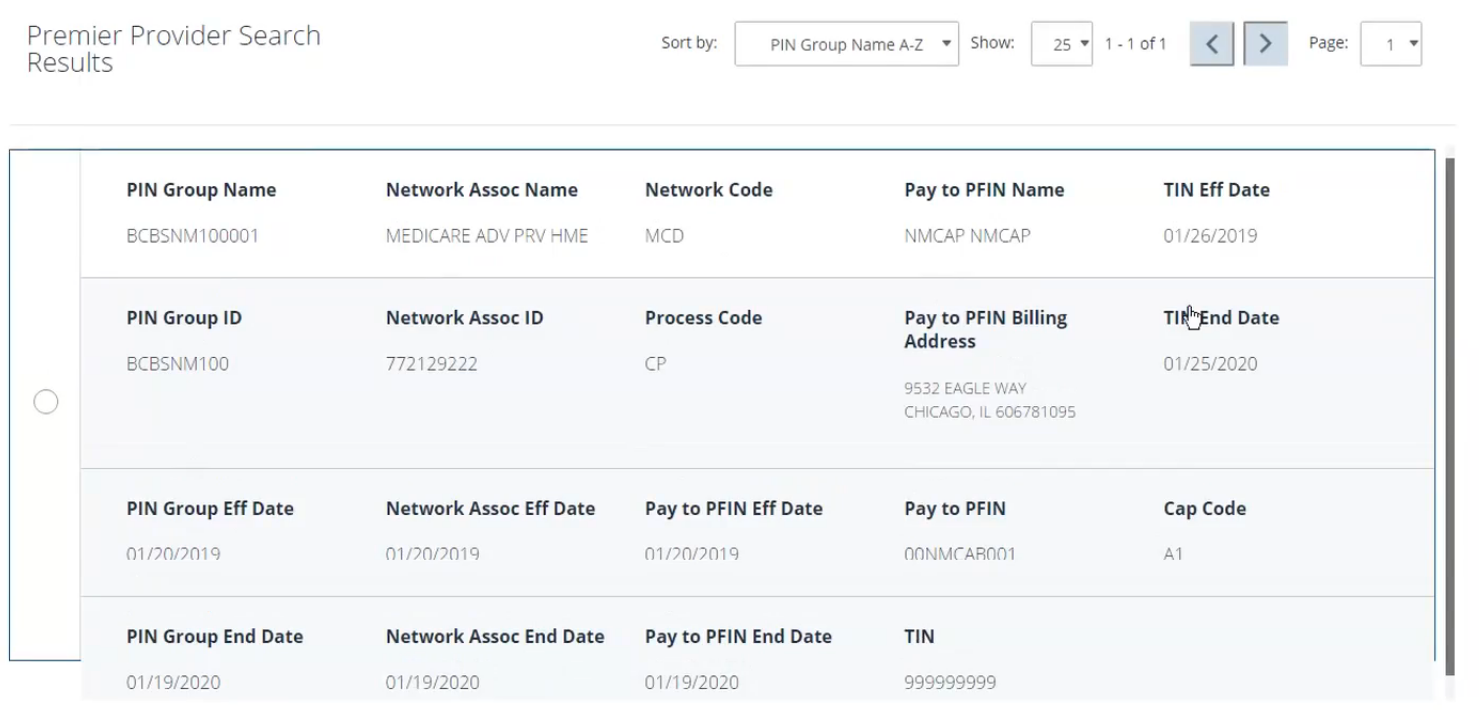


New Payment Arrangement screen (cont…) – Payee added



We click on “Edit” to add the Arrangement Payee Effective date range and we need to make sure that this arrangement payee date range has to be within the Vbr effective date range, otherwise no need to have arrangement for that payee.

[So this is the correlation worth to be mentioned: When a provider is added as payee in the VBR system, it is added from the Provider Source System(Premier Provider) through Provider API using the search mechanism of PIN Groupid. This Provider has its own dates ranges as shown below in one of the search results:



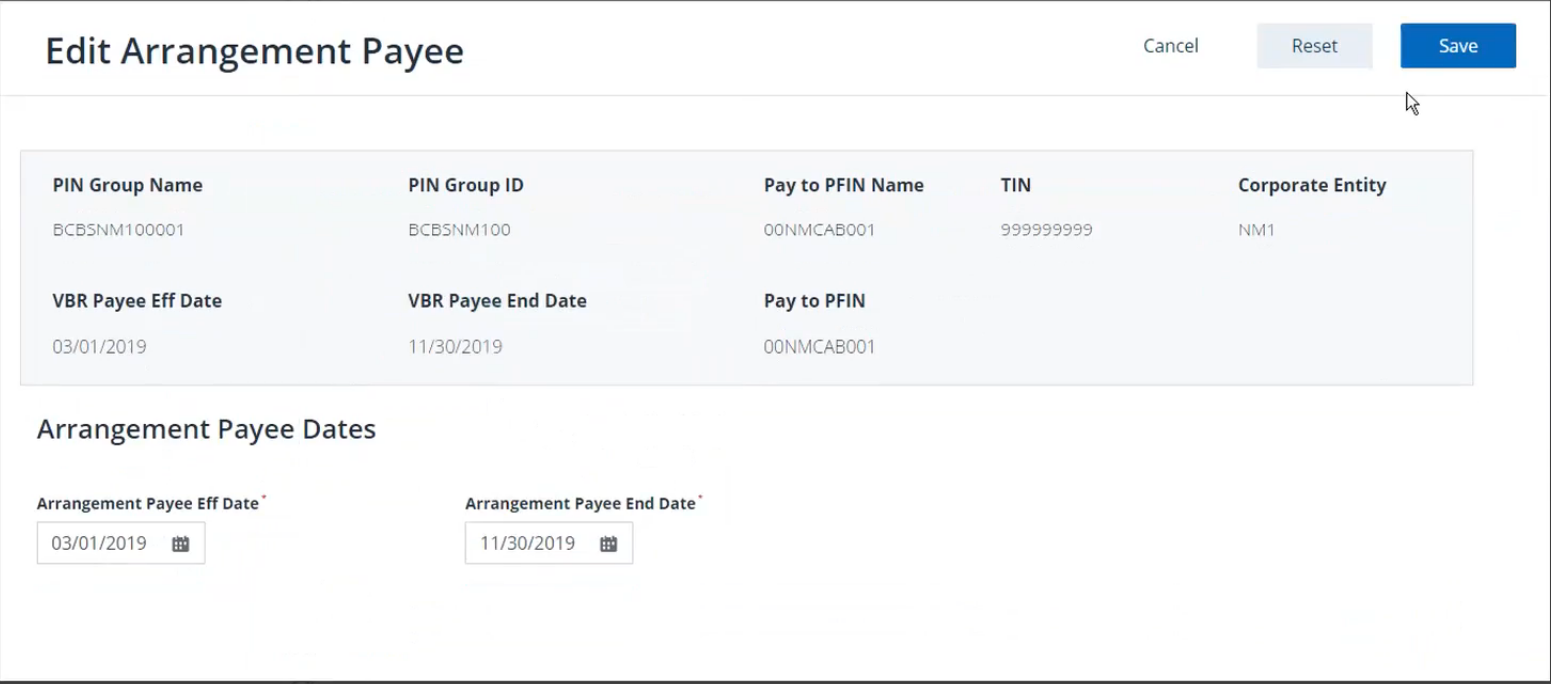
We see Pin Group Eff. date range, Network Assoc Eff. date range and Pay to PFin Eff. date range.

Now once we select this and proceed to add this in our Vbr system, then we have to add the Vbr Eff. date range, which implies the period for which the payee has to be active in the Vbr system.

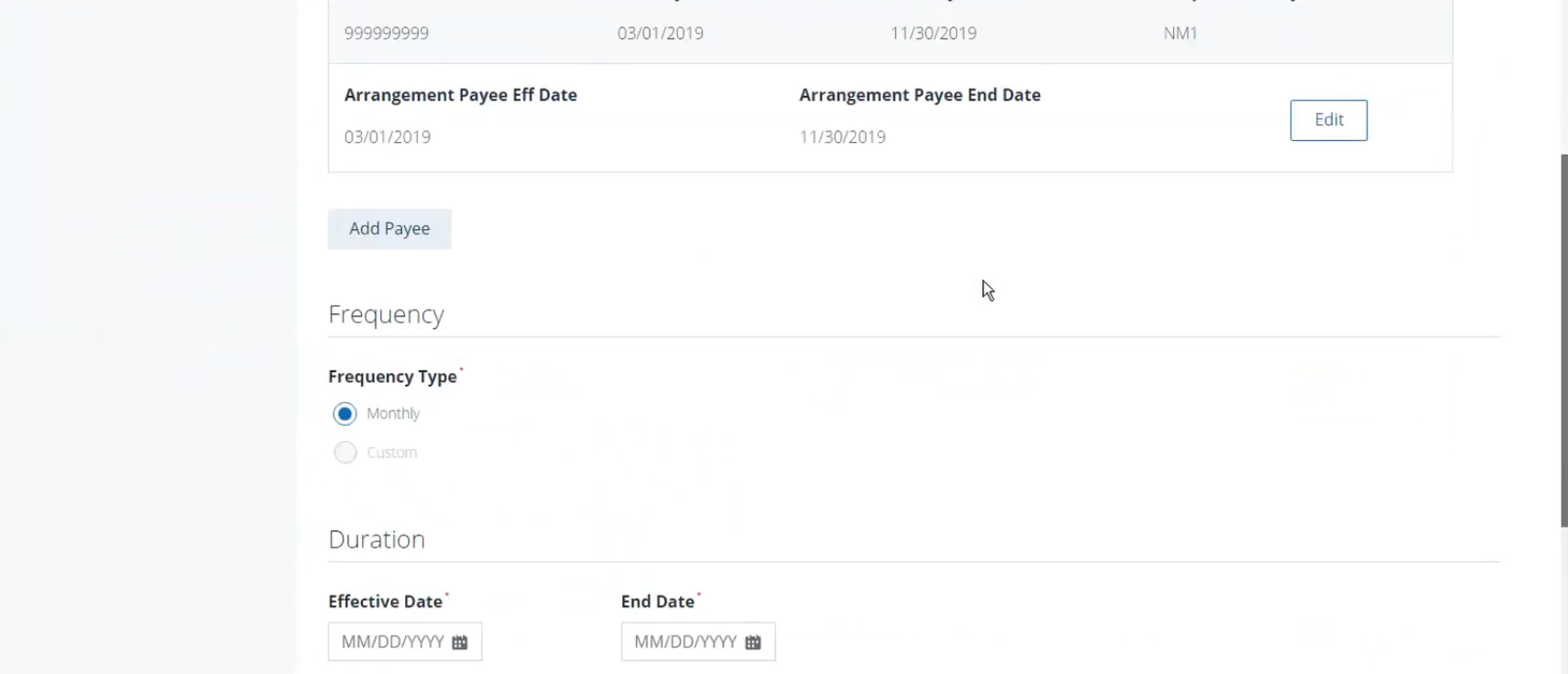
We don’t want to set the dates which are outside of our provider API. We have to make sure the Vbr payee date range is within that of the provider, otherwise what it would mean that we are giving the payee an active status in our system even if it is not active in Provider API.

And then this payee is linked with an arrangement, it means the time period for which this payee entity will be part of the given arrangement. And so the arrangement payee date has to be within the Vbr payee date.]

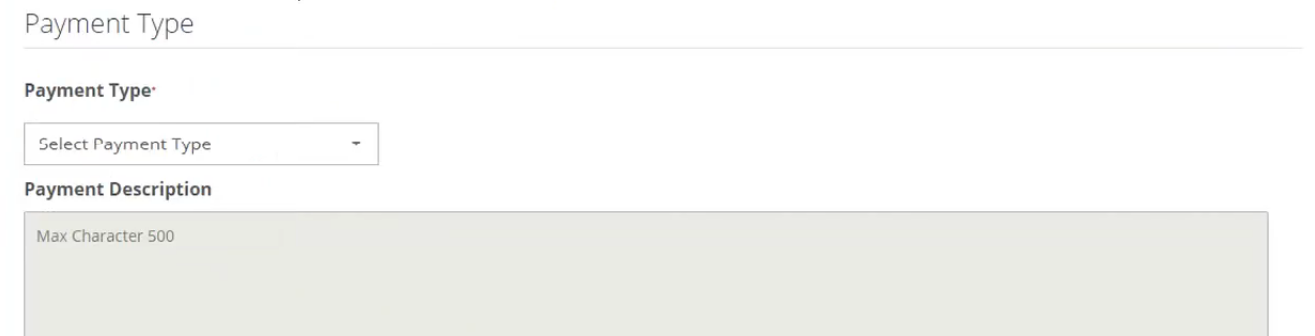
New Payment Arrangement screen (cont…) – Payee edited in terms of Effective dates and then saved



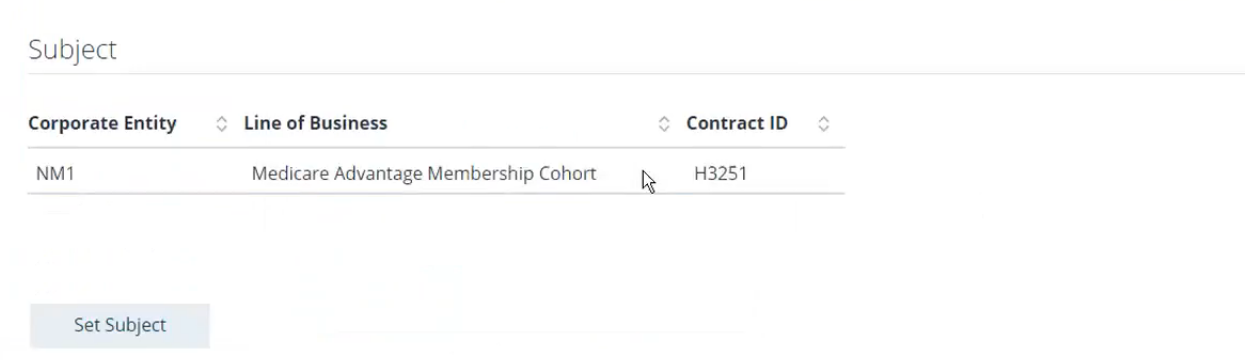
New Payment Arrangement screen (cont…) – add Frequency Type and Arrangement duration



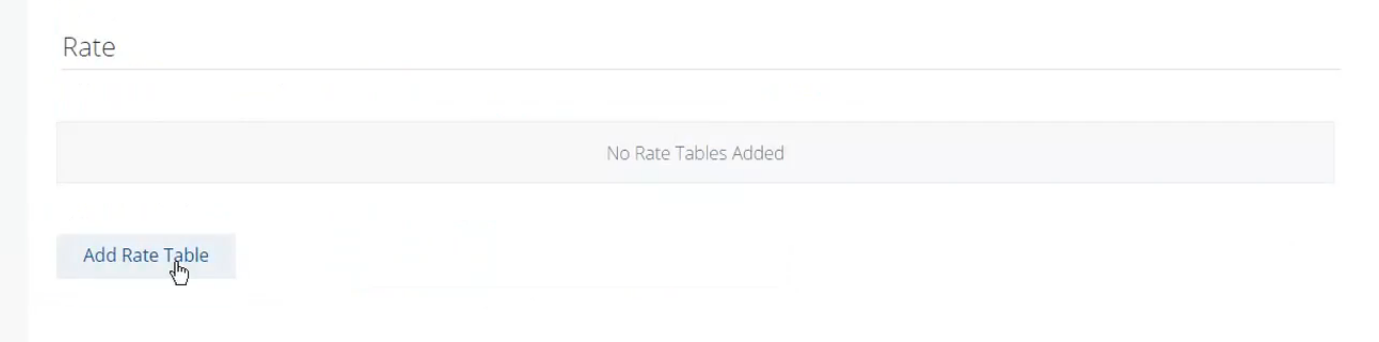
New Payment Arrangement screen (cont…) – add Payment Type and Description



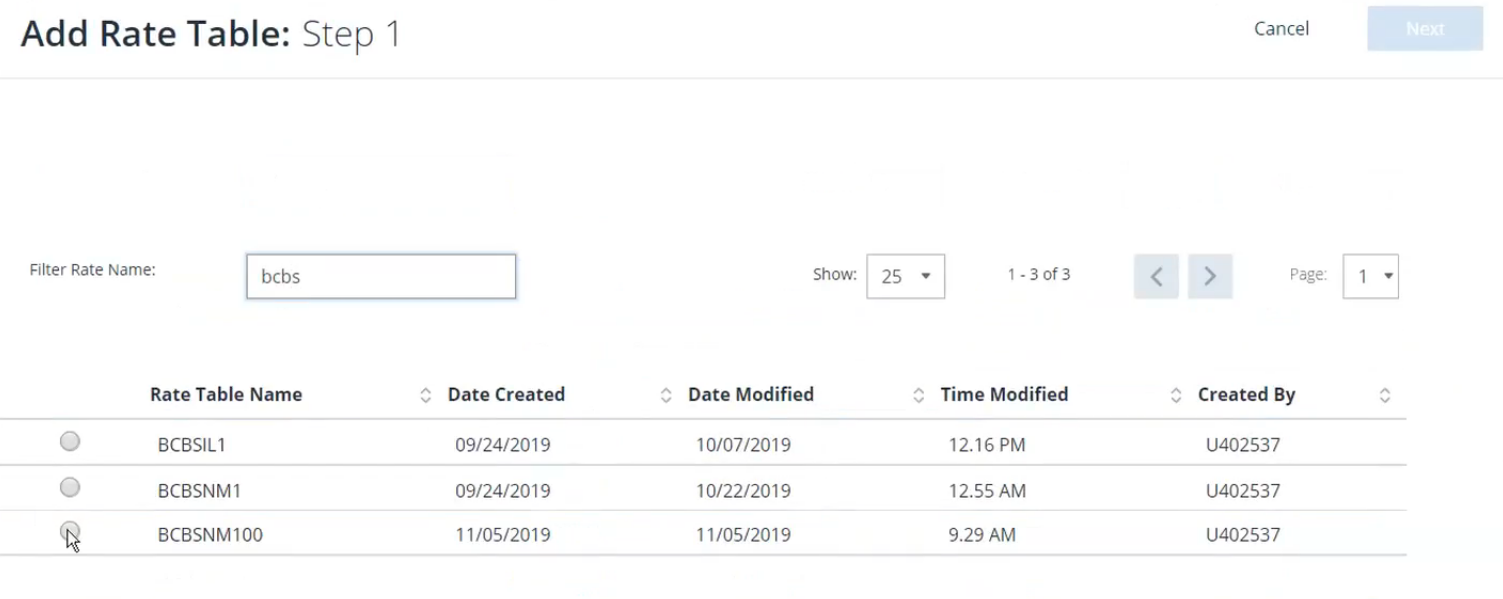
New Payment Arrangement screen (cont…) – add Subject



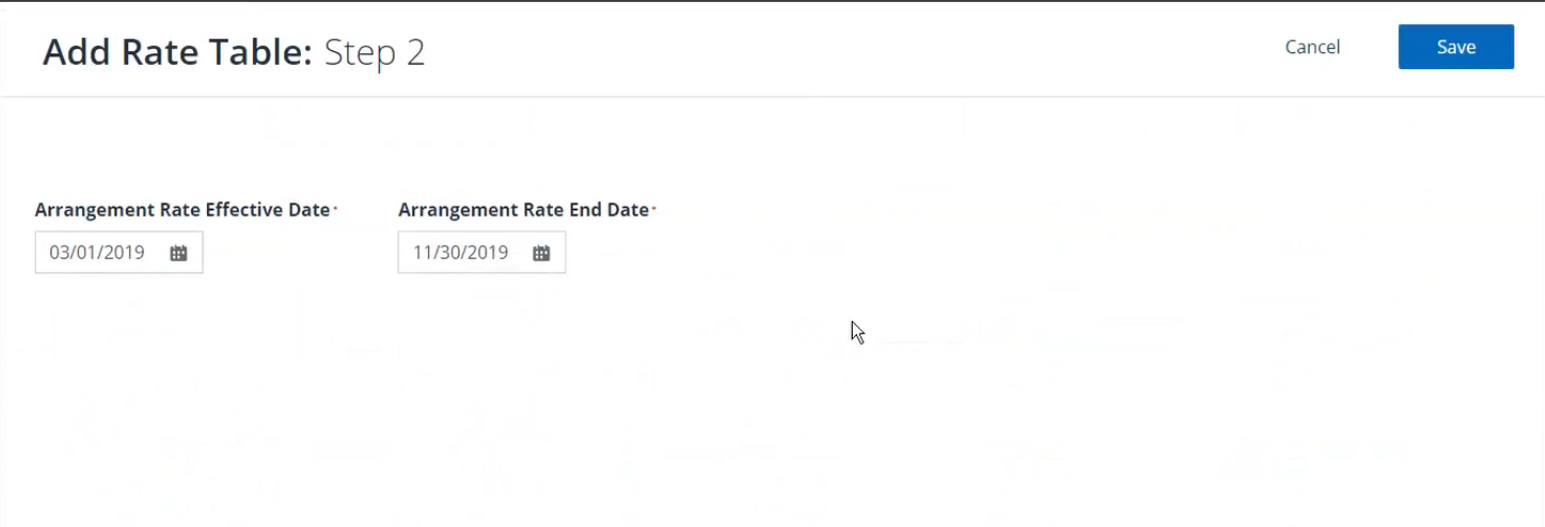
New Payment Arrangement screen (cont…) – add Rate table



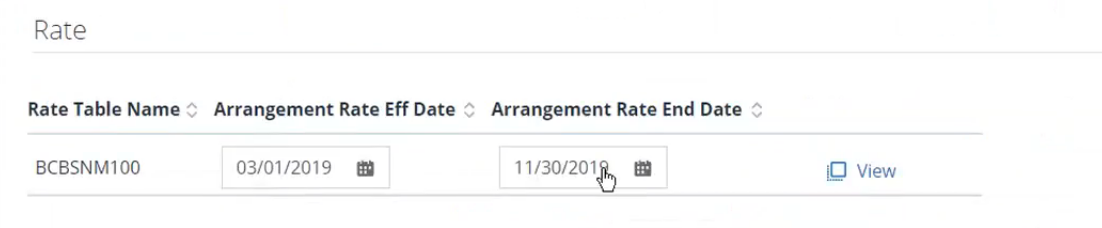
Add Rate Table – Search a Rate table from Vbr system



Add Rate Table – Enter the Arrangement Rate Effective date range



New Payment Arrangement screen (cont…) – Rate table added



In case of linking Rate table too to the Arrangement, we need to make sure that the Arrangement Rate date range is in alignment with the Vbr Rate date range.

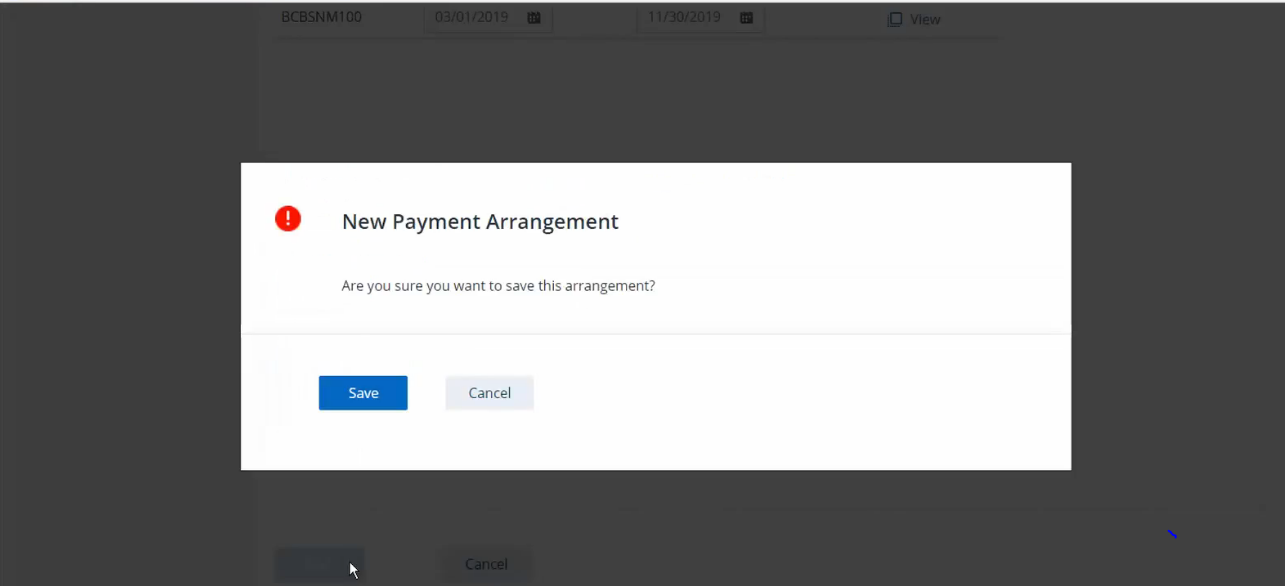
For ex.: Say suppose we have the Rate table in Vbr system whose Effective date range is Feb, 2019 to Dec, 2019. But when we link it to the arrangement and enter the Arrangement Rate Effective date range between Jan, 2019 - Nov, 2019, then we get the error msg as: Arrangement rate date range should fall within Vbr Rate date range

We need to correct this before linking this Rate table to the given arrangement.

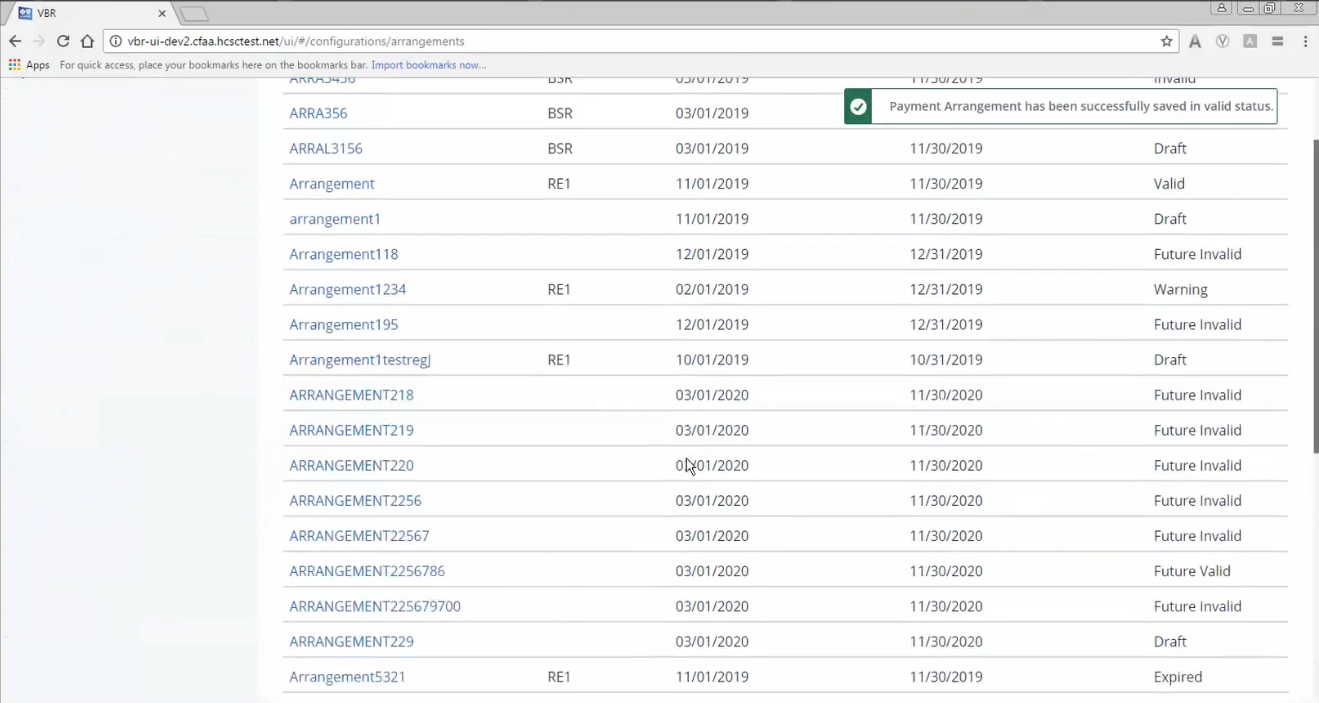
And then if the Arrangement duration is Jan, 2019 – Dec, 2019. And we add a Rate table for the period of only March, 2019- Nov, 2019, then we see the warning msg: “Arrangement Rate table does not cover the full duration of the arrangement”. We see this warning msg due to the Validation that happens during the Save process, and then as a result the Arrangement will be saved in Invalid status.

We can fix this issue and do the “Save” again to save the Arrangement in Valid status.

So after filling out everything, we click on “Save”, it asks for confirmation as shown below:



Payment Arrangements screen - Confirmation msg is displayed:



Code flow for the saveArrangement

PaymentArrangementController:saveArrangement()

Input: SaveRequest

Sends control to PaymentService

Output: SaveReponse



PaymentArrangementService:savePaymentArrangement()

Input: SaveRequest

Output: SaveReponse

Checks for if name or desc empty and then Validation starts

PaymentArrangementValidator:validateSave()

Input: SaveRequest

Output: Status, Errors

Fetches the Rate DTO and Payee DTO.

1.Effective start&end date

2. Frequency code

3. Arrangement name & description

4. Subject

5. Rate tables

6. Payee

Validates at the Arrangement level

1.Arrangement duration with Arrangement Payee date duration

2. Arrangement Payee date duration within Vbr Payee duration

3. Arrangement Payee validity for processing month

4. Vbr Payee validation w.r.t. Premier Provider

5. Gap-overlap validation

6.Duplication

Validates at Arrangement Payee level

1.Arrangement duration with Arrangement Rate date duration

2. Arrangement Rate duration with Flat rate.

3. Arrangement Rate validity for processing month

4. Gap-overlap validation

5.Duplication

Validates at Arrangement Rate level

On unsuccessful validation(in case of actual errors found during validation process)

On successful validation(in case of errors found during validation process have the severity level “W”)

PaymentArrangementServicer:savePaymentArrangement()

If errors found then throws the ApplicationException

PaymentArrangementService:savePaymentArrangementRequest()

Input: SaveRequest

Output: SaveReponse

Refers the PaymentArrangementMapper to extract the required instances of DataRecord for saving into different respositories.

PaymentArrangmentRepository

paymentArrangementHistoryRepository

paymentArrangementMemberSubjectRepository

PaymentArrangmentPayeeRepository

VbrPayeeRepository

PaymentArrangementService:savePaymentArrangement()

Input: SaveRequest

Output: SaveReponse

Based on the ValidationStatusCode the Success or Warning message is set in the paymentArrangementSaveResponse

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_EXPIRED*** = "EX";

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_FUTURE\_INVALID*** = "FI";

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_DRAFT*** = "DF";

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_INVALID*** = "IV";

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_VALID*** = "VA";

**public** **static** **final** String ***ARRANGEMENT\_STATUS\_WARNING*** = "WA";