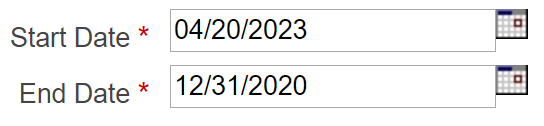
**BSI UI Analysis**

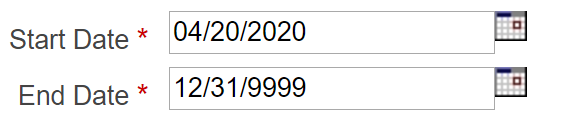
**Summary**

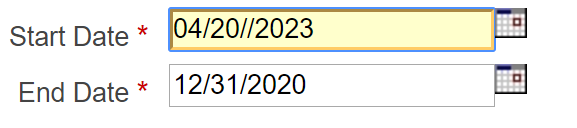
* Date Field Only Validates Date Format and does not check if the input date is of past or future date.
* Each field must have a description field added in the metadata
* There are dependent fields, one depends on other being populated first
* There are fields, which can enable disable other fields based on their selected value
* There are fields with tabs functionality, i.e. based on your tab selection different fields are shown below it.
* Related Activities needs to be added

1. **Date Field only validates date format but does not check past or future date (More than 20 fields have this)**

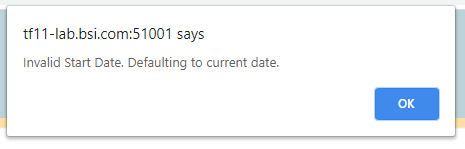


* Here it’s not checking if the Start Date is in the future, or end date is before start date.

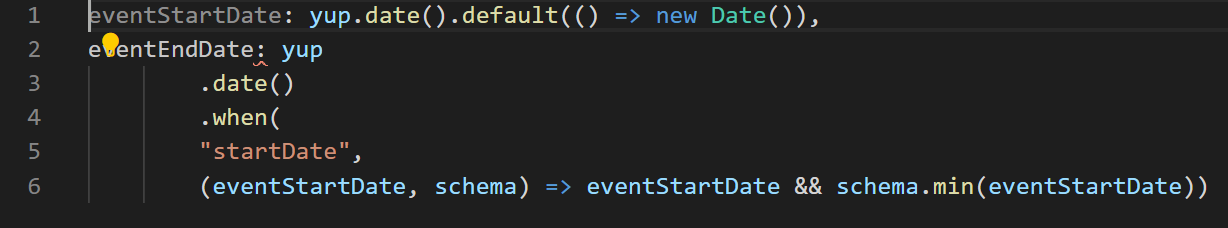




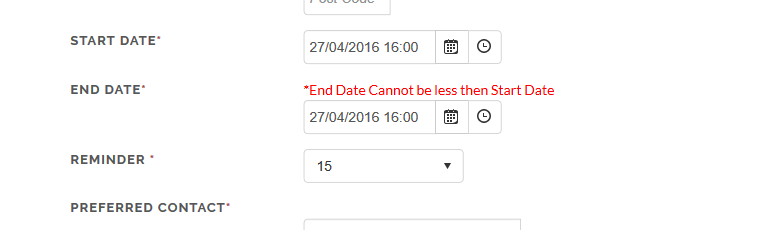
* Here if the date format is invalid, then only it’s showing the Invalid Start date error

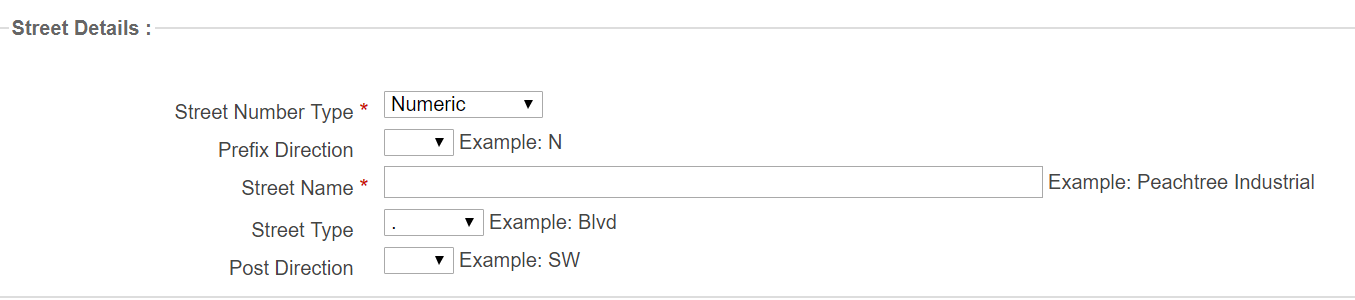


**Solution:** Currently, date field only validates the input date format, we can have additional validation that validates if the end date is before start date where we can pass custom validation prop and have all the logic there, other validation can be done server side.



The validation prop will have similar logic that can be passed to the component for additional validation.

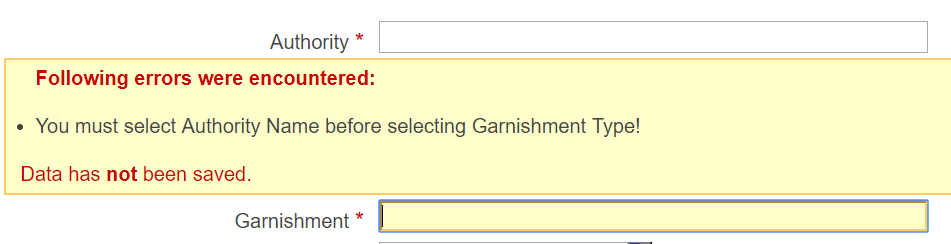


1. **Some fields have short description explaining how the input will behave based on the input. (From 10 -15 fields have this)**

* Here, On Street name, the Example: Peachtree Industrial, is like a description to the field. There are many other fields with similar description. So, it’s better to add description field in metadata.
* **Solution:** We can have additional “description” field in metadata, that displays the sample input for different fields.

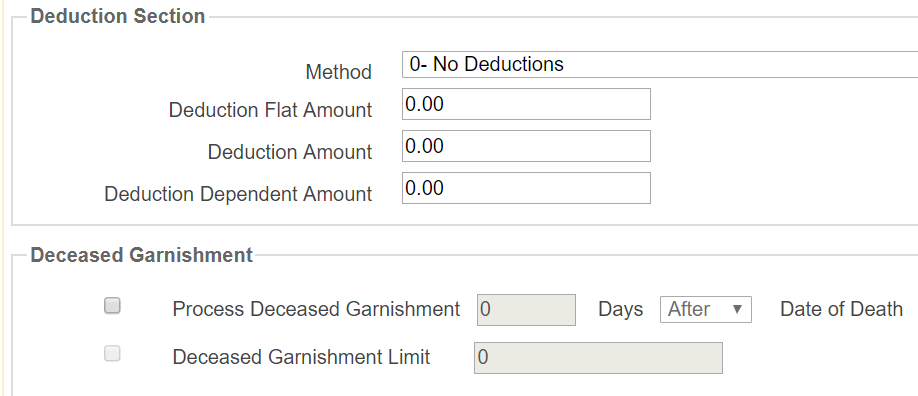


1. **There are dependent fields, one depends on other being populated first (Around 10 – 15 fields have this)**

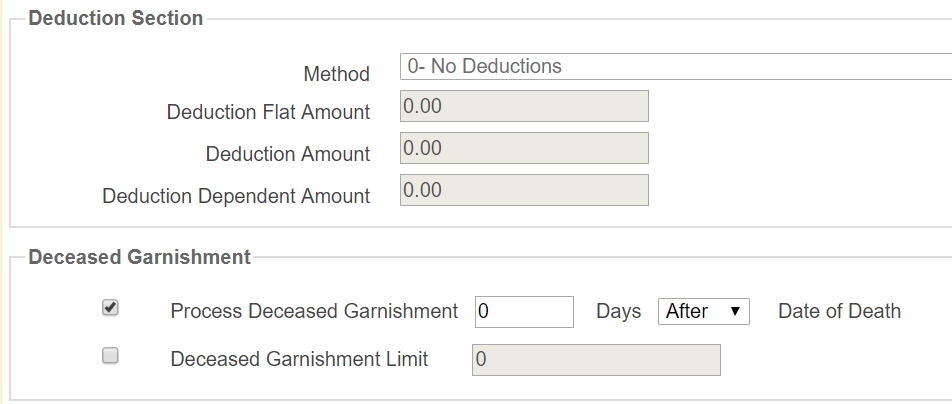


* Here, if you try to fill Garnishment Field, it displays error, saying Authority Name must be selected first
* **Solution:** So, we can have a validation prop that is passed to the component, that returns the field info, and we can send all the additional logic to schemaValidator on top of the existing validation.

1. **There are fields, which can enable disable other fields based on their selected value (About 5 – 10 fields have this)**

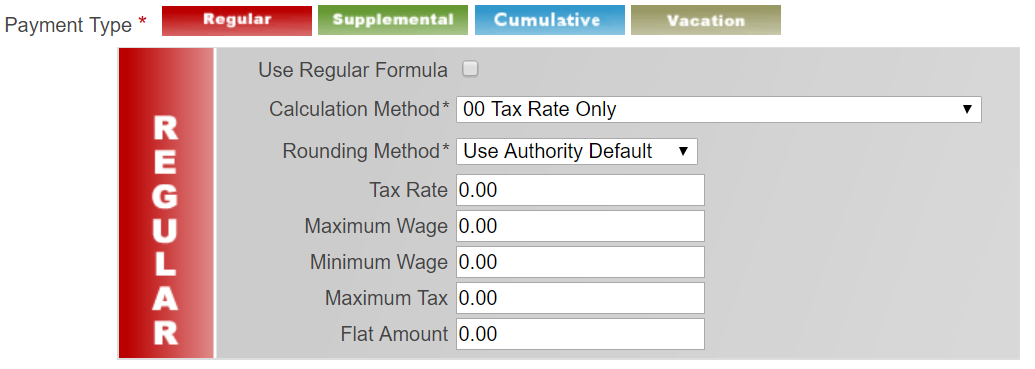


* Here When Process Deceased Garnishment is not selected then Deduction Section fields are editable.

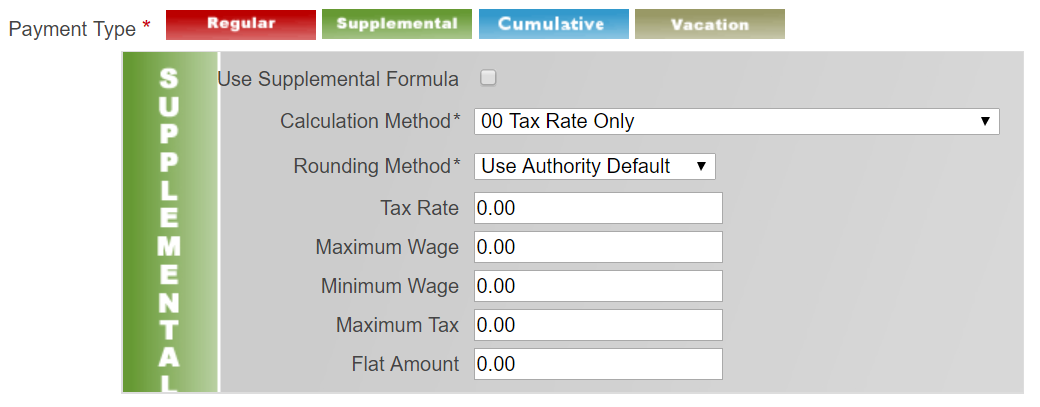


* Here, once the Process Deceased Garnishment is Checked, the Deduction Section is disabled. **Solution:**
* Since the logic for disabling the field is already there, and we are disabling fields based on metadata, in this case, we need to check the field value and add it to the Field Schema on the fly. Should have an array list that contains the id of field that it can disable on selection.

1. **There are fields with tabs functionality, i.e. based on your tab selection different fields are shown below it. (Total Fields ~ 1 - 2)**

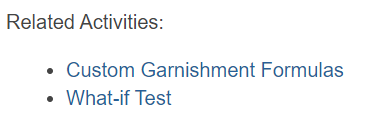


* Selecting Regular Displays Regular fields.



* Selecting Supplemental Displays Supplemental fields. This type of field is very rare, so far only this one, so we can render this manually, without using the library.
* **Solution:** Tab fields are very rare maybe around 2-5 fields at max in the library so it’s better to keep them separated from the library

1. **Related Activities needs to be added (Total Fields ~ 15)**



There are some pages, which have Related Activates link, like recent usage, so we might need to implement this feature in the library.

* **Solution:**  Just like recent usage implementation, we can replicate exact functionality for Related Activities.

