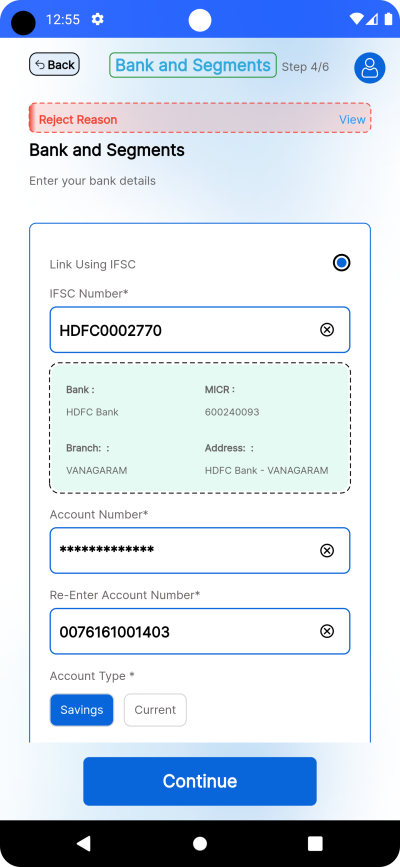
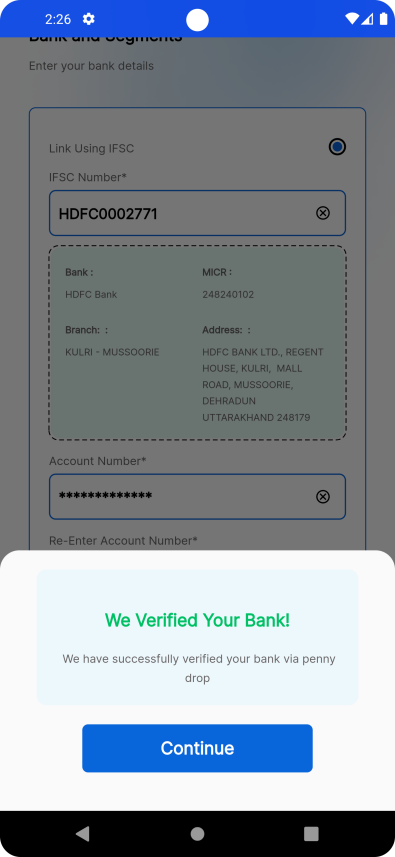
**Bank Screen:**

A6

B58

B60

B59

F87

F86

F85

Button Fields API

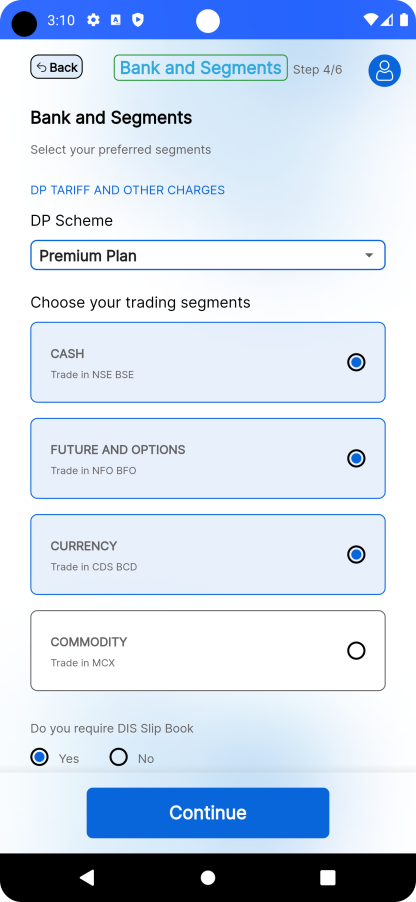
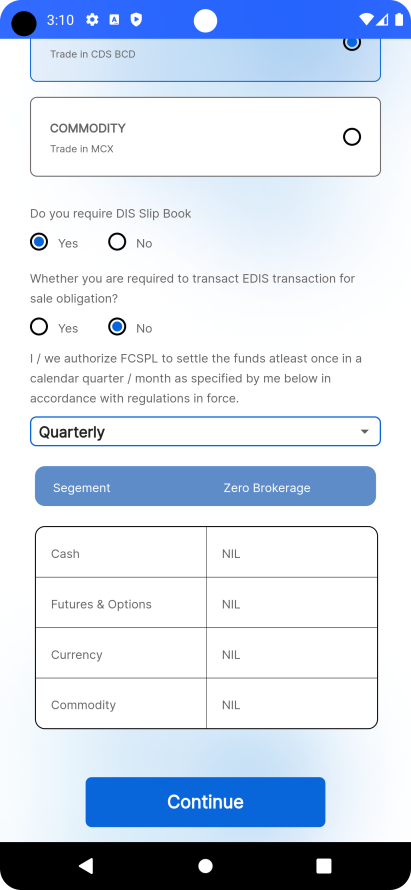
A\_

F\_

B\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Field ID.** | **Field Name** | **Field Type** | **Events, Descriptions & Validations** |
| 1. | F85 | ifscController | **Textformfield** | * It is used to enter the Ifsc number of the user’s bank. * when we edit this Field, the Ifsc Number is getting from the api“**getBankDetails** ” where “**ifsc**” as a key |
| 2 | F86 | accNumController | **Textformfield** | * It is used to enter the account Number of the user which is in Obscure Text. * when we edit this Field, the Account Number is getting from the api“**getBankDetails** ” where “**accno**” as a key |
| 3 | F87 | confirmaccnumController | **Textformfield** | * It is used to enter the account Number of the user second time to confirm that account number previously entered was correct. * when we edit this Field, the Re-Entered Account Number is getting from the api“**getBankDetails** ” where “**accno**” as a key |
| 4 | B58 | accType | **InkWell** | * It is a button that confirms the account type which is a Saving or Current Account * The Value of the Account type is getting from the api “**dropDowndata**” where “**lookupvaluearr**” as a key and the code is “**Bank Account type**” * when we edit this Field, the the Account Type is getting from the api“**getBankDetails** ” where “**acctype**” as a key |
| 5 | B59 | continueButton | **Elevated Button** | * This is a button that is used to move to the next Screen when the above field is Valid * When the user clicks the button it post the values from the screen to the Api “**addBankDetail**”.After successfully completed it shows the bottom sheet that verified. |
| 6 | A6 | DottedBorder  container | **DottedBorder** | * This is the Container only visible when the Ifsc of the bank is entered correctly.And it shows the bank details for the IFSC * when we edit this Field, the Details of the bank is getting from the api “**IfscDetails**” and stored in appropriate key as micr , branch , address , bank |
| 7 | B60 | continueButton | **Elevated Button** | * When the user click this button it calls the “**routerflow**” api. * the “**routerflow**” tells the route name for the next screen. |

**Segment Selection :**

** **

B62

B63

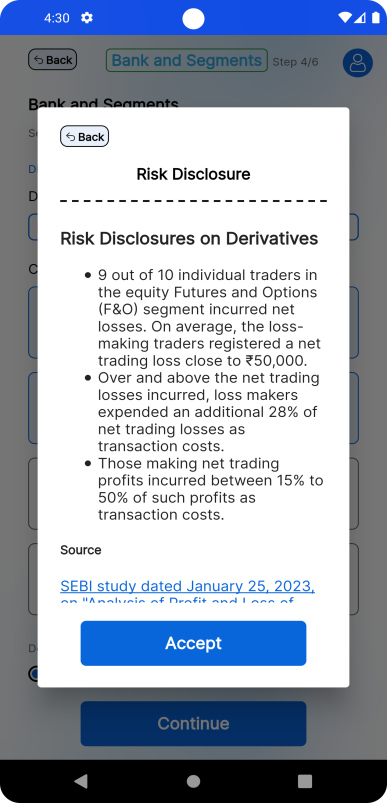
F90

F89

F88

A7

B61

****

B64

Button Fields API

A\_

F\_

B\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Field ID.** | **Field Name** | **Field Type** | **Events, Descriptions & Validations** |
| 1. | B61 | Text\_Field | **InkWell** | * It is used to view the Tariff and the other charges of the Dp Scheme. |
| 2 | F88 | dpSchemeController | **DropDown** | * It shows the Dp Scheme Plans. * when we edit this Field, the Dp Scheme is getting from the api “**dropDowndata**” where “**lookupvaluearr**” as a key and the code is “**DematData**”. |
| 3 | F89 | CustomTile CheckBox | **ListTile** | * It is used to select the trading segments which the user wants * The user can’t deselect the Cash Segment * when we edit this Field, the the trading Segment is getting from the api “**GetDematandService**” where “**service\_map**” as a key. |
| 4 | B62 | Dis & eDis value | **Radio Button** | * It is a radio button which selects that the user wants Dis slip book or not and the Edis transaction Details. * The user can select any one of the value as Yes. * when we edit this Field, the the trading Segment is getting from the api “**GetDematandService**” where “**dpscheme**” as a key for “**dematinfo**” key. |
| 5 | F90 | settlementWise | **DropDown** | * In this dropdown, the user can select how the FCSPL funds settle like Monthly or Quarterly * when we edit this Field, the the Dp Scheme is getting from the api “**GetDematandService**” where “**lookupvaluearr**” as a key and the code is “**Settlement\_Type**”. |
| 6 | A7 | BrokerageDetails | **Container** | * It is used to view the Brokerage Plan details of the Segment. * when we edit this Field, the the trading Segment is getting from the api “**GetDematandService**” where “**brokhead**” and “**brokdata**” as a key. |
| 7 | B63 | Continue Button | **Elevated Button** | * This is a button that is used to move to the next Screen when the above Screen is Valid * When the user clicks the button it post the values from the screen to the Api “**DematServeInsert**”.After successfully completed it calls the “**routerflow**” api. * If the user selects more segment along with the Cash segment means,this button show the Risk Disclosure from the Api “**getriskdisclosure**” where “**riskDisclosure**” as a key. * The “**routerflow**” tells the route name for the next screen. |
| 8 | B64 | Accept Button | **Elevated Button** | * When the user clicks the Accept button it calls the “**routerflow**” api. * The “**routerflow**” tells the route name for the next screen. |

**Views:**

Bank Screen :

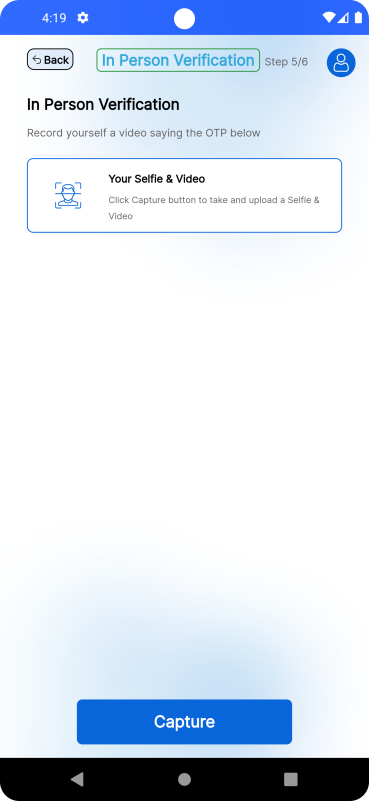
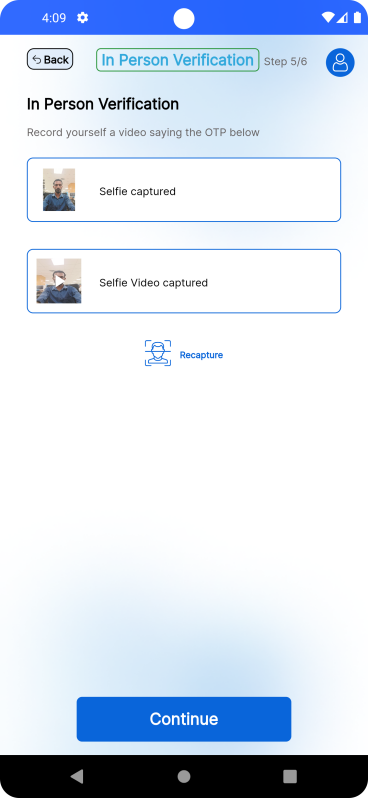
|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **bankscreen.dart** | | |
| **Custom Components Used** | 1. **[StepWidget](#_BidContainer:)**  * This widget have the common template used in all the screen.  1. [showSnackbar](#_SnackBar:)  * This widget is used to show the message in snackbar.  1. [AccountType](#_SnackBar:)  * This widget is used for identify which account type button which uses multiple times  1. [CustomRadioButton](#_SnackBar:)  * This widget is used for showing Radio Button which is Customize.  1. [Custom](#_SnackBar:)FormField  * This widget is used for showing Text Form Field where the validation and decorations are used in common.  1. [Custom](#_SnackBar:)Button  * This widget is used for showing Elevated Button which is used in many Screen. | | |
| **Variables** | **Variables** | **Datatype** | **Variable Definition** |
| isLoadingDetails | bool | This variable is used to show the UI when all the details are fetched from the Api |
| fetchBankDetails | FetchBankDetailByIfsc | This variable stores the value of the Bank Details in a class |
| ifscController | TextEditingController | This variable holds the value of the Ifsc which the user entered |
| accNumController | TextEditingController | This variable holds the value of the account number which the user entered |
| confirmaccnumController | TextEditingController | This variable holds the value of the account number which the user entered |
| isIFSCFinished | bool | This variable tells that Ifsc details Api is Finished or not. |
| upiBankType | String | This variable holds the Account type of the Bank that user selects |
| \_formKey | GlobalKey<FormState> | It is used to enable the form validation and submission. |
| isFormValid | bool | It tells if the form is valid or not |
| scrollController | ScrollController | This variable is used for scroll in Listview |
| accountDetail | Map | It holds the Account Details entered by the User previously from the Api as a Map. |
| accountTypes | List | This List holds the all Account type of the Bank from the Api. |
| **Hooks Used** | 1. **initState()**  * This method is used to call the **getBankDetails()** function before the page Loading. | | |
| **Methods** | 1. **getBankDetails()**  * The getBankDetails method retrieves and processes bank account details asynchronously. It shows a loading indicator, fetches account types and specific bank account details, updates the user interface, and performs validation before the UI is loading * These Details are filled by the user previously.  1. **changeUpiBankType()**  * This method is used to change the Account type as Current or Savings Account.  1. **fetchBankDtlFromIfsc()**  * This method is used to fetch the Bank Details according to the IFSC code which is entered by the User.  1. **formValidation()**  * The formValidation method checks if the form fields for IFSC code, account number, and confirmation account number are filled. It validates the form and updates the state accordingly.  1. **insertBankDetails()**  * The insertBankDetails method collects bank account details from user input, prepares them in a structured format, and sends them to an API to insert the details into the database. It also handles UI updates and navigation based on the response.  1. **getNextRoute()**  * The getNextRoute method determines the next route to navigate to by sending the current route information to an API and receiving the next route endpoint. It handles UI updates such as showing and hiding a loading indicator during the process. | | |
| **API’s** | 1. **getBankDetailsAPI()**  * This method is used to get the bank details of the particular IFSC through API  1. **getDropDownValues()**  * This method is used get the drop down values for the Account type Like Saving or Current through API.  1. **getBankWithAccountDetailsAPI()**  * This method is used to get the bank details which already filled by the user through API  1. **insertBankDetailsAPI()**  * This method is used to post the Details which are entered by the user to the Api and get the response according to that. | | |
| **Custom Actions** | N/A | | |

Segment Selection:

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **segmentselection.dart** | | |
| **Custom Components Used** | 1. **[StepWidget](#_BidContainer:)**  * This widget have the common template used in all the screen.  1. [showSnackbar](#_SnackBar:)  * This widget is used to show the message in snackbar.  1. showTerms  * This widget is used show the Terms and Condition to the User for the Risk  1. [CustomRadioButton](#_SnackBar:)  * This widget is used for showing Radio Button which is Customize.  1. CustomDropDown  * This widget is used to show the customized Drop Down which are used commonly through out the app | | |
|  | **Variables** | **Datatype** | **Variable Definition** |
| dpSchemeController | TextEditingController | This variable holds the dp Scheme value selected from the Drop Down |
| settleOfFCSPLController | TextEditingController | This variable holds the Settle Of FCSPL value selected from the Drop Down |
| demantDropDownValues | List | This variable holds the all the Dp Scheme List of Map from the api. |
| demantDropDownOptions | List | This variable holds the all the Dp Scheme values from the demantDropDownValues . |
| settleOfFCSPLDropDownValues | List | This variable holds the all the settle Of FCSPL List of Map from the api. |
| settleOfFCSPLDropDownOptions | List | This variable holds the all the settle Of FCSPL values from the settle Of FCSPL Drop Down Values. |
| disValue | String | This variable holds the value of dis Slip book that Yes or No |
| eDisValue | String | This variable holds the value of eDis Transaction that Yes or No |
| titles | List | This variable holds the title of the Trading segment. |
| subTitles | List | This variable holds the subTitles of the Exchange segment. |
| selectedTile | List | This variable holds what are the segments are selected by the user |
| userSelects | List | This variable holds what are the segments can select or deselect by the user |
| brokerageHeading | List | This variable holds the heading for the brokerage tariff. |
| brokerageData | List | This variable holds the Data for the brokerage tariff. |
| tariffDetailsUrl | String | This variable holds the Url for the tariff |
| serviceData | Map | This variable holds the details of the Exchange segment from the Api. |
| oldDemantServ | Map | In this map contains the previously entered Demant and service values |
| isEdit | bool | This value tells that the screen is edited or not |
| scrollController | ScrollController | This variable is used for scroll in Listview |
| htmlData | Map | This map contains the Html value for the risk disclosure. |
| **Hooks Used** | 1. **initState()**  * This method is used to call the **fetchData()** function before the page Loading. | | |
| **Methods** | 1. **fetchData()**  * The fetchData method retrieves various dropdown values and brokerage details asynchronously, processes the data, updates the user interface with the fetched information, and manages the state accordingly. * In this some Details are fetched that are filled by the user previously.  1. **createDematServiceJson()**  * The createDematServiceJson method constructs a JSON object containing demat account details, selected services, and brokerage information. This JSON object is used to update or manage demat account data and also update some of the values in Service Data based on the User Selects.  1. **insertSegmentDetails()**  * The insertSegmentDetails method updates the demat account service details by constructing a JSON object, comparing it with the existing details, and sending it to an API if changes are detected. After updating, it navigates to the next route.  1. **getHtmlData()**  * The getHtmlData method retrieves HTML content related to risk disclosure, processes it, and displays the content in a terms and conditions dialog.  1. **insertData()**  * The insertData method validates and processes selected trading segments before either fetching risk disclosure data or inserting segment details, based on the selected segments by the user.  1. **getNextRoute()**  * The getNextRoute method determines the next route to navigate to by sending the current route information to an API and receiving the next route endpoint. It handles UI updates such as showing and hiding a loading indicator during the process. | | |
| **API’s** | 1. **getServeBrokerDetailsApi()**  * This method is used to get the Service and Brokerage details which are already filled by the user  1. **getDropDownValues()**  * This method is used get the drop down values for the Demant Scheme and another for settle Of FCSPL Drop Down Values from the Api.  1. **getRouteNameInAPI()**  * This method is used to get the route name based on the previous route from the Api.  1. **insertDemantserveApi()**  * This method is used to post the Demant and Service details which are entered by the user to the Api and get the response according to that.  1. **getDishClosureData()**  * This method is used to get the Risk And Disclosure Data from the Api. | | |
| **Custom Actions** | N/A | | |

|  |
| --- |
|  |

**IPV:**

** **

B67

B66

B68

B65

After Digio

Before Digio

Button

B\_

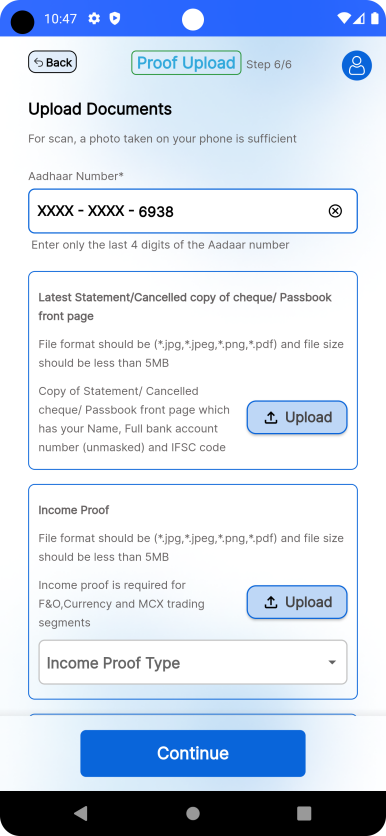
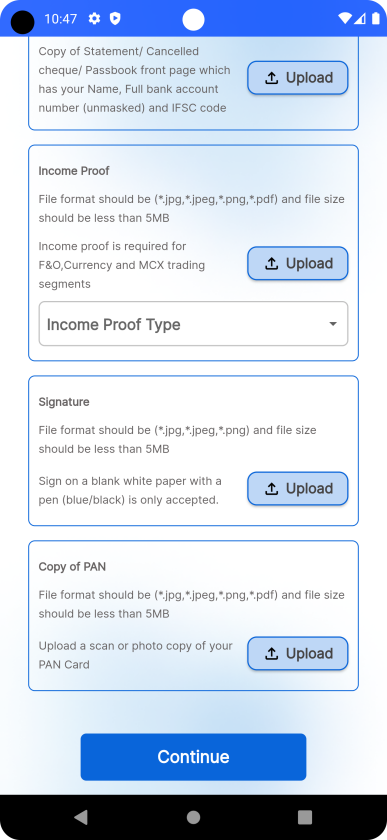
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Field ID.** | **Field Name** | **Field Type** | **Events, Descriptions & Validations** |
| 1 | B65 | Capture | **Elevated Button** | * When the User Click the button it Launch the Digio url. |
| 2 | L66 | Recapture | **Inkwell** | * After the IPV Section completed, the Recapture button shows. * When we click this button it again calls the Digio Url |
| 3 | B67 | Continue | **Elevated Button** | * After completion of Digio the Capture Button changed into Continue Button * When the user clicks the button it post the data from the screen to the Api “**DematServeInsert**”.After successfully completed it calls the “**routerflow**” api. * the “**routerflow**” tells the route name for the next screen. |
| 4 | B68 | image | **Gesture Dedector** | * When the user clicks any of the fields it sgows the captured photo or video. |

**Views :**

Ipv Screen:

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **IPVPage.dart** | | |
| **Custom Components Used** | 1. **[StepWidget](#_BidContainer:)**  * This widget have the common template used in all the screen.  1. [showSnackbar](#_SnackBar:)  * This widget is used to show the message in snackbar.  1. VideoPlayerInReview  * This widget is used to show the video which will captured in Digio.  1. LoadImage  * This widget is used to show the image which will captured in Digio.  1. openAlertBox  * This widget is used to show the alert box for the permission required for Digio. | | |
| **Variables** | **Variables** | **Datatype** | **Variable Definition** |
| ipvDetails | Map | This variable holds the information about the IPV getting from the Api as map. |
| image | String | This variable hold the id of the image |
| otp | String | This variable holds the value of the otp. |
| video | String | This variable hold the id of the video. |
| kidId | String | This variable hold the kidId of the IPV. |
| gwtId | String | This variable hold the gwtId of the IPV. |
| email | String | This variable holds the user’s mailId |
| \_scrollController | ScrollController | This variable is used for scroll in Listview |
| **Hooks Used** | 1. **initState()**  * This method is used to call the **getIPVDetails()** function before the page Loading. | | |
| **Methods** | 1. **getIPVDetails()**  * The getIPVDetails method is responsible for fetching IPV (In-Person Verification) details from an API, processing the response, and updating the UI accordingly.  1. **getUserDetails()**  * The getUserDetails method retrieves user details necessary for In-Person Verification (IPV) from an API, handling platform-specific permissions for iOS, and processing the response accordingly and if the necessary ids are get it calls the method **function()**.  1. **function()**  * The function method initializes the DigiO KYC (Know Your Customer) workflow plugin, starts the KYC workflow, handles workflow events, and saves the IPV (In-Person Verification) details in the API.  1. **getPermission()**  * The getPermission method requests permissions for accessing location, camera, and microphone from the user. It displays an alert box informing the user about the required permissions and gives them the option to grant or deny them when the user is using iOs as Platform.  1. **getNextRoute()**  * The getNextRoute method determines the next route to navigate to by sending the current route information to an API and receiving the next route endpoint. It handles UI updates such as showing and hiding a loading indicator. | | |
| **API’s** | 1. **getUserDetailsForIPVInAPI()**  * This method is used for get the necessary Ids of the IPV when the Digio process.  1. **saveIPVDetailsInAPI()**  * This method is used save the data from the Digio Process.  1. **getIPVDetailsAPI()**  * This method is used to get the IPV details such as Image and Video before the page loading.These information are stored by the user previously. | | |
| **Custom Actions** | N/A | | |

**Upload Proof:**

** **

B73

B71

B72

F92

F91

B70

B69

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Field ID.** | **Field Name** | **Field Type** | **Events, Descriptions & Validations** |
| 1 | F91 | Aadhaar Number | **TextFormField** | * In this Field the user have to enter the last four digit of the Aadhaar Number for verification * This field is only visible when the value of “**aadhaarFlag**” key is “Y” from the Api. |
| 2 | F92 | Income proof drop down | **DropDown** | * It shows the type of Proof for Income Validation * when we edit this Field, the Income proof is getting from the api “**dropDowndata**” where “**lookupvaluearr**” as a key and the code is “**IncomeProof**”. |
| 3 | B69 | Bank Proof | **Elevated Button** | * This button is used to upload a Bank statement for proof. |
| 4 | B70 | Income Proof | **Elevated Button** | * This button is used to upload a proof for Income verification. |
| 5 | B71 | Signature | **Elevated Button** | * This button is used to upload the signature of the user. |
| 6 | B72 | Copy of Pan | **Elevated Button** | * This button is used to upload the Pan for proof. |
| 7 | B73 | continue | **Elevated Button** | * This is a button that is used to move to the next Screen when the above field is Valid * When the user clicks the button it post the values from the screen to the Api “**FileUploads**” . After successfully completed it calls the “**routerflow**” api. * the “**routerflow**” tells the route name for the next screen. |

**Views :**

File Upload:

|  |  |  |  |
| --- | --- | --- | --- |
| **File Name** | **file\_upload.dart** | | |
| **Custom Components Used** | 1. **[StepWidget](#_BidContainer:)**  * This widget have the common template used in all the screen.  1. [Custom](#_SnackBar:)FormField  * This widget is used for showing Text Form Field where the validation and decorations are used in common.  1. pickFileBottomSheet  * This widget is used to show the bottom sheet that user have to select which type of file has been upload  1. CustomUpload  * This widget is used to handle file uploads with additional features like showing a dropdown menu, previewing the uploaded file, and displaying error messages.  1. [CustomRadioButton](#_SnackBar:)  * This widget is used for showing Radio Button which is Customizable and reusable widget. | | |
| **Variables** | **Variables** | **Datatype** | **Variable Definition** |
| incomeproofController | TextEditingController | This variable holds the income proof value selected from the Drop Down |
| buttonIsEnable | bool | This variable tells that when the continue button is enabled. |
| selectedProofType | String | This variable holds what income proof is selected the dropdown. |
| incomeProofTypeOptions | List | This variable hold the type of proof for income . |
| incomeProofTypes | List | This variable hold the response for type of proof for income from the Api. |
| fileUploadDetails | List | This variable hold the response for details about the File Upload. |
| oldProofCode | String | This variable holds code for Income proof that is code is previously entered. |
| oldAadhaarNo | String | This variable holds aadhaar number that is previously entered. |
| aadhaarVisible | bool | This variable hold that when the aadhaar field is visible. |
| scrollController | ScrollController | This variable is used for scroll in Listview |
| aadhaar | TextEditingController | This variable holds the value for the Aadhaar number |
| countinueButtonIsTriggered | bool | This bu`tton tells if the continue button is triggered or not. |
| \_formKey | GlobalKey<FormState> | It is used to enable the form validation and submission. |
| **Hooks Used** | 1. **initState()**  * This method is used to call the **fetchDropdownValues()** function before the page Loading. | | |
| **Methods** | 1. **fetchDropdownValues()**  * The fetchDropdownValues method is responsible for fetching the necessary drop down values from the api and also calls the **fetchFileId** method  1. **fetchFileId()**  * The fetchFileId method retrieves the file ID and related user information from an API, processes the response, and give it to the variables accordingly.  1. **uploadFile()**  * The uploadFile method processes and uploads files along with other form data to a specified API. It navigates to the next route if the upload is successful.  1. **checkButtonEnable()**  * The checkButtonEnable method determines if a button should be enabled based on specific conditions related to the state of the form and its fields. It updates a state variable buttonIsEnable and triggers a state update to reflect the change in the UI.  1. **getNextRoute()**  * The getNextRoute method determines the next route to navigate to by sending the current route information to an API and receiving the next route endpoint. It handles UI updates such as showing and hiding a loading indicator. | | |
| **API’s** | 1. **getDropDownValues()**  * This method is used get the drop down values for the IncomeProof Values from the Api.  1. **fetchFileIdAPI()**  * This method is used to retrieves the file ID and related user information from an API.  1. **fileUploadAPI()**  * This method is used to insert all the files for proof to the api.  1. **getRouteNameInAPI()**  * This method is used to get the route name based on the previous route from the Api. | | |
| **Custom Actions** | N/A | | |