1. **Class: MainActivity**

***void onCreate(Bundle savedInstanceState):***

* This methods checks if the user is new or existing.
* If a pin exists, then it is an existing user, else it is a new user.
* If the user is new, set the layout "first\_time\_login".
* Else set the layout "login\_page".
* In the former, the user is prompted to set a pin.
* In the latter, the user is prompted to authenticate.

***void onLogin(View view):***

* In the "login\_page", the user enters the PIN and clicks the Login button
* In this method, the existing PIN in the database is compared against the PIN user entered
* If it is the same, then set the layout "home"
* Else display the message "Incorrect PIN entered, please check"

***void openIncomePage(View view):***

* This page allows the users to enter the Income details.
* It is invoked when the user clicks on the button “Income” from home page
* It sets the layout “income”
* The page contains the following fields: Date (calendar help), Amount (numeric text box), Recurring (checkbox) and Optional notes (alphanumeric textbox).

***void saveIncome(View view)***

* This method is used to save the Income details entered by the user.
* This method is invoked when the user clicks on the Submit button in the "Income" page.
* Retrieve the value from the fields and typecast them into String.
* Call the method addData from class DatabaseIncomeExpense to save these details to the database.

***void openExpensePage(View view):***

* This page allows the users to enter the Expense details.
* It is invoked when the user clicks on the button “Expense” from home page
* It sets the layout “expense”
* It contains the following fields: Category (dropdown), Date (calendar help), Amount (Numeric text box), Recurring (checkbox - if the income is repetitive over certain frequency), Payment Method(dropdown), and Optional notes (AlphaNumeric textbox).
* The entries in the "Category" dropdown list box are pre-populated from the database via method getData of class DatabaseCategories.

***void saveExpense(View view)***

* This method is used to save the expense details entered by the user.
* This method is invoked when the user clicks on the Submit button in the "Expense" page.
* Retrieve the value from the fields and typecast them into String.
* Call the method addData from class DatabaseIncomeExpense to save these details to the database.

***void onSetCurrency(View view):***

* This method is invoked when user clicks on the button “Set Currency” in settings page.
* It is used to select the currency unit.
* A pop-up is launched which contains a list of different currency units.

***void openSummaryPage(View view)***

* This method is invoked when a user clicks on the button “View Summary” in the settings page.
* It is used to view the summary of income or expense incurred by the user.
* It sets the layout "summary".
* The user is prompted to enter a start date and end date & click OK button.
* Uses setOnDateListener function to launch calendar for setting the date fields.

***void showSummary(View view)***

* This method is invoked when a user clicks on the button OK in the View Summary page
* Retrieve the value from user fields and typecast to String format.
* Call the method getData from class DatabaseIncomeExpense to get these details from the database.
* Display it in form of pie chart with the data retrieved.

***void openSettingsPage(View view)***

* This method is invoked when a user clicks on the button “Settings” in home page
* It sets the layout “settings”.

***void onDateSet(DatePicker view, int year, int month, int dayOfMonth)***

* This method is invoked when the user selects a date field
* It is used to format the date type

***void onClick(View v)***

* This method is invoked whenever a button is clicked
* A switch statement is constructed with ID of different buttons to segregate button-specific operations.
  + ***R.id.set\_btn***

check if the PIN entered and re-entered by the user match or not, if it matches save the PIN to database else show an error message.

* + ***R.id.edit\_categories***

event handler for the button “Edit Categories”. Sets the layout “category\_popup”.

* + ***R.id.deleteCategoryBtn***

Deletes the selected categories from database via method deleteData from class Database Categories

* + ***R.id.addCategoryBtn***

prompt user to enter a new category

* + ***R.id.btnOKAddCategory***

Add the new category to database

* + ***R.id.modifyCategoryBtn***

Prompts the user to change category

* + ***R.id.btnOKModifyCategory***

Saves the modified value to the database

* + ***R.id.changePin***

Prompts the user to enter new PIN

* + ***R.id.change\_pin\_ok***

Saves the modified PIN value to the database

1. **Class: DatabaseHandler**

***void onCreate(SQLiteDatabase db)***

* Create a table PIN

***void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)***

* Delete the table PIN

***boolean addData(String pin)***

* Adds the PIN data to database table PIN

***Cursor getPinData()***

* Retrieve the PIN value saved in database table PIN

***boolean modifyData(String newPin, String oldPin)***

* Update the existing PIN value

1. **Class: DatabaseCategories**

***void onCreate(SQLiteDatabase db)***

* Create a table CATEGORIES

***void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)***

* Delete the table CATEGORIES

***boolean addData(String category)***

* Appends a new category to the database table CATEGORIES

***boolean deleteData(String category)***

* Delete the category from the database table CATEGORIES

***boolean modifyData(String new\_category, String old\_category)***

* Update the category value

***List<String> getData()***

* Retrieves the table contents

1. **Class: DatabaseCurrency**

***void onCreate(SQLiteDatabase db)***

* Create a table CURRENCY

***void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)***

* Delete the table CURRENCY

***boolean addData(String currencyCode)***

* Appends Currency code to the database table

***boolean modifyData(String new\_unit, String old\_unit)***

* Updates the existing currency code

***Cursor getData()***

* Retrieves the table contents of table CURRENCY

1. **Class: Transactions**

***void onCreate(SQLiteDatabase db)***

* Create a table TRANSACTIONS

***void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)***

* Delete the table TRANSACTIONS

***boolean addData(String category, String startDate,String endDate,String amount,***

***String code, String paymMethod, String note, String indicator)***

* Adds the expense or income details to Database

***Cursor getData(Date startDate, Date endDate)***

* Retrieves the table contents of TRANSACTIONS table

1. **Class: DatePicker**

***Dialog onCreateDialog(@Nullable Bundle savedInstanceState)***

* Loads the calendar with default date as current date