Overall Algorithm for Product

```
If (Program Is Started)
    Welcome frame opens
    If (start is clicked)
        Log in frame opens
        Log in fields of username and password are displayed
        User enters fields
        If (log in information is in database)
            Dispose log in frame
            Open input frame
        If (sign up is selected)
            Sign up fields of username, password, email and security questions are displayed
            User enters sign up fields
            If (username already exists)
                Display error frame
            Else
                User sign up is added to database
        If (forget password is selected)
            User enters email
            If (email is in database)
                Display security question fields
                User enters fields
                If (fields match database fields)
                    User changes password and is updated in database
                Else
                    Display error
            Else
                Display error

    If (Input frame is opened)
        If (register business from JMenu is selected)
            If (business name is entered && register is clicked)
                Business name is registered in database
        If (generalHelp is selected from JMenu)
            General Help frame opens and is displayed
            If (return is clicked)
                General help frame is disposed
        If(calculationHelp is selected from JMenu)
            Calculation Help frame opens and is displayed
            If (return is clicked)
                Calculation help frame is disposed
        If (about is selected from JMenu)
```

```
        About frame is opened
        If (return is clicked)
                About frame is disposed
If (Net Present Value calculation is selected from JMenu)
        profit fields are displayed and user enters information
        If (calculate profit is clicked)
                Profit is calculated for specified business
                Profit information is added to database
                Profit is displayed in output frame
                If (return is clicked)
                        Output is disposed
        If (next is clicked)
                profit fields are displayed and user enters information
                If (calculate profit is clicked)
                        Profit is calculated for specified business
                        Profit information is added to database
                        Profit is displayed in output frame
                        If (return is clicked)
                                Output is disposed
        If (next is clicked)
                Risk fields are displayed and user enters information
                If (calculate risk is clicked)
                        Risk is calculated for specified business
                        Risk information is added to database
                        Risk is displayed in output frame
                        If (return is clicked)
                                Output is disposed
        If (next is clicked)
                Wacc fields are displayed and user enters information
                If (calculate wacc is clicked)
                        Wacc is calculated for specified business
                        Wacc information is added to database
                        Wacc is displayed in output frame
                        If (return is clicked)
                                Output is disposed
        If (next is clicked)
                Asset and liability fields are displayed and user enters information
                If (calculate working capital is clicked)
                        Working capital is calculated for specified business
                        Working capital information is added to database
                        Working capital is displayed in output frame
                        If (return is clicked)
                                Output is disposed
```

```
        If (next is clicked)
                Cash flow fields are displayed and user enters information
                If (calculate is clicked)
                        Cash flow is calculated for specified business
                        Certain Cash flow information is added to database
                        Cash flow is displayed in table
                        If (return is clicked)
                                Output is disposed

        If (next is clicked)
                Net Present value fields are displayed and user enters information
                        If (calculate net present value is clicked)
                                Net present value is calculated for specified business
                                Net present value information is added to database
                                Net present value is displayed in output frame
                                If (return is clicked)
                                        Output is disposed
                        If (display report is clicked)
                                Report is displayed
                                If (return is clicked)
                                        Report is disposed

If (overall report is clicked)
        Display table of all of the data from the database of the specified user
If (Investment comparison is selected from JMenu)
        Investment comparison fields are displayed
        User enters business name, investment amount in dollars, and time
        If (calculate is clicked)
                Net present value for all the businesses in the database is
                calculated
                Business with greatest net present value is displayed
```

Database Tables

Key:

Primary Keys

Combined Primary Keys

ProfileInformation

Email	Username	Password

SecurityQuestions

SecurityID	SecurityQuestion

SecurityAnswers

Email	SecurityID	Security Answer

NonTimeFinancials

Email	Business Name	Actual Equity	Actual Debt	Cost of Debt	Number of Employees	Years in Business	Number of Countries	Number of Competitors	Growth rate

RiskFactor

Number of	Years in	Number of	Number of	Risk Factor

Employees	Business	Countries	Competitors	

TimeFinancials

Email	Business Name	Time (Years)	Capital Expenditure (CapEx)	Assets	Liabilities	Depreciation	Revenue	Costs

Profits

Time (Years)	Revenue	Costs	Profits (Excluding Tax)

WorkingCapital

Time (Years)	Assets	Liabilities	Working Capital

FreeCashFlow

Time (Years)	Working Capital	Depreciation	Profits	Capital Expenditure (CapEx)	Free Cash Flow

WeightVariables

Actual Debt	Actual Equity	Weight of Debt	Weight of Equity

EquityCost

Weight of Debt	Risk Factor	Cost of Equity

Wacc

Cost of Debt	Cost of Equity	Weight of Debt	Weight of Equity	WACC (Weighted Average Cost of Capital)

NetPresentValue

Time (Years)	Free Cash Flow	WACC (Weighted Average Cost of Capital)	Net Present Value

<u>Data Types:</u>

Username: **varchar(100)**
Email: **varchar(100)**
Password: **varchar(100)**
Security Question: **varchar(100)**
Security ID: **Integer**
Security Answer: **varchar(100)**
Business Name: **varchar(100)**
Actual Equity: **Double**
Actual Debt: **Double**
Cost of Debt: **Double**
Weight of Debt: **Double**
Weight of Equity: **Double**

Number of Employees: **Integer**
Years in Business: **Integer**
Number of Countries: **Integer**
Number of Competitors: **Integer**
Growth rate: **Double**
Risk Factor: **Integer**
Time: **Integer**
Capital Expenditure: **Double**
Assets: **Double**
Liabilities: **Double**
Depreciation: **Double**
Revenue: **Double**
Costs: **Double**
Profits: **Double**
Working Capital: **Double**
Free Cash Flow: **Double**
Weighted Average Cost of Capital: **Double**
Net Present Value: **Double**

Class Descriptions:

Classes:	Descriptions:
Warning	An error frame that warns the user for invalid inputs
About	An about frame that discusses what this program does for the user and more information on the creator of this program
Input	An input frame that allows the user to input financial information in order for it to be calculated
LogIn	A log in frame that allows for the user to log in to their specified account and/or create an account
GeneralHelp	A general help frame that provides the user with general information on how to use the program and frequently asked questions
CalculationHelp	A calculation help frame that provides the user with information on the calculations that this program provides
Welcome	Introduces the user to the business application and what it does
Risk	Calculates for a business's risk based on multiple risk factors that a user can enter
BusinessFinancials	Calculates for a business's WAAC, income without tax, free cash flow, and net present value
BusinessReport	Displays a business report for the user based on a specified business
CashFlowTable	Displays a free cash flow table of a specified business
GeneralOutput	Outputs the answer of certain financial calculations
InstallDatabase	Allows for the database, with its tables, to be installed, for different financial information and user information to be stored

AccessDatabase	Allows for the access of business and user information available in the database
LogInManager	Manages the different users that log in and sign up with an account in the application and then provides that information to the log in frame

UML Diagrams Classes

BusinessFinancials

- wacc : double
- weightedEquity : double
- weightedDebt : double
- equityCost : double
- profit : double
- freeCashFlow : double
- netPresentValue : double
- workingCapital : double

+ BusinessFinancials() :
+ BusinessFinancials(double waac, double weightedEquity, double weightedDebt, double equityCost, double profit, double freeCashFlow, double netPresentValue) :
+ setWaac(double waac) : void
+ setWeightedEquity(double weightedEquity) : void
+ setWeightedDebt(double weightedDebt) : void
+ setEquityCost(double equityCost) : void
+ setProfit(double profit) : void
+ setFreeCashFlow(double freeCashFlow) : void
+ setNetPresentValue(double netPresentValue) : void
+ getWaac() : double
+ getWeightedEquity() : double
+ getWeightedDebt() : double
+ getEquityCost() : double
+ getProfit() : double
+ getFreeCashFlow() : double
+ getNetPresentValue() : double
+ waacCalculation(double weightedEquity, double weightedDebt, double equityCost, double debtCost) : double
+ weightedDebtCalculation(double actualEquity, double actualDebt) : double
+ weightedEquityCalculation(double actualEquity, double actualDebt) : double
+ equityCostCalculation(Integer riskFactor, double debtCost) : double
+ cashFlowCalculation(double workingCapital, double capitalExpenditure, double depreciation) : double
+ presentValueCalculation(Integer time, double freeCashFlow, double wacc) : double
+ toString() : String

DatabaseAccess

- dbName : String
- dbConn : Connection
- data : ArrayList <ArrayList<String>>

+ DatabaseAccess() :
+ DatabaseAccess(String dbName) :
+ setDbName(String dbName) : void
+ setDbConn(String dbConn) : void
+ setData(ArrayList<ArrayList<String>> data) : void
+ getDbName() : String
+ getDbConn() : Connection
+ getData(String tableName, String[] tableHeaders) : ArrayList<ArrayList<String>>
+ createDb(String newDbName) : void
+ createTable(String newTable, String dbName) : void
+ closeDbConn() : void
+ 2dArrayConvert(ArrayList <ArrayList<String>> data) : Object[][]
+ toString() : String

InstallDatabase

+ main(String[] args) : void

Risk

- employeeAmount : int
- businessYears : int
- countriesAmount : int
- competitorsAmount : int
- riskPercentage : int

+ Risk() :
+ Risk(int employeeAmount, int businessYears, int countriesAmount, int competitorsAmount, int riskPercentage) :
+ setEmployeeAmount(int employeeAmount) : void
+ setBusinessYears(int businessYears) : void
+ setCountriesAmount(int countriesAmount) : void
+ setCompetitorsAmount(int competitorsAmount) : void
+ setRiskPercentage(int riskPercentage) : void
+ getEmployeeAmount() : int
+ getBusinessYears() : int
+ getCountriesAmount() : int
+ getCompetitorsAmount() : int
+ getRiskPercentage() : int
+ calculateRisk(int employeeAmount, int businessYears, int countriesAmount, int competitorsAmount) : int
+ toString() : String

LogInManager

- userName : String
- email : String
- password : String
- securityQuestion : String
- securityQuestionId : int
- securityAnswer : String

+ LoginManager() :
+ LoginManager(String userName, String email, String password, String securityQuestion, int securityQuestionId, String securityQuestionId) :
+ setUserName(String userName) : void
+ setEmail(String email) : void
+ setPassword(String password) : void
+ setSecurityQuestion(String securityQuestion) : void
+ setSecurityQuestionId(int securityQuestionId) : void
+ setSecurityAnswer(String securityAnswer) : void
+ getEmail() : String
+ getUserName() : String
+ getPassword() : String
+ getSecurityQuestion() : String
+ getSecurityQuestionId() : int
+ getSecurityAnswer() : String
+ checkDuplicateEmail(String email) : boolean
+ checkDuplicateUser(String userName) : boolean
+ checkPassword(String userName, String email) : boolean
+ checkSecurityQuestions(String securityQuestion, int securityQuestionId, String securityAnswer)
+ toString() : String

GeneralOutput

- labels : JLabel
- buttons : JButton

+ GeneralOutput() :
+ actionPerformed() : void

GeneralHelp

- labels : JLabel
- buttons : JButton
- panels : JPanel

+ GeneralHelp() :
+ actionPerformed() : void

Welcome

- labels : JLabel
- buttons : JButton
- panels : JPanel

+ Welcome() :
+ actionPerformed() : void

Input

- labels : JLabel
- buttons : JButton
- panels : JPanel
- textFields : JTextFields

+ Input() :
+ actionPerformed() : void

LogIn

- labels : JLabel
- buttons : JButton
- panels : JPanel
- textFields : JTextFields

+ LogIn() :
+ actionPerformed() : void

Warning

- labels : JLabel
- buttons : JButton
- panels : JPanel

+ Warning(String message) :
+ actionPerformed() : void

About

- labels : JLabel
- buttons : JButton
- panels : JPanel

+ About() :
+ actionPerformed() : void

CalculationHelp

- labels : JLabel
- buttons : JButton
- panels : JPanel

+ GeneralHelp() :
+ actionPerformed() : void

BusinessReport

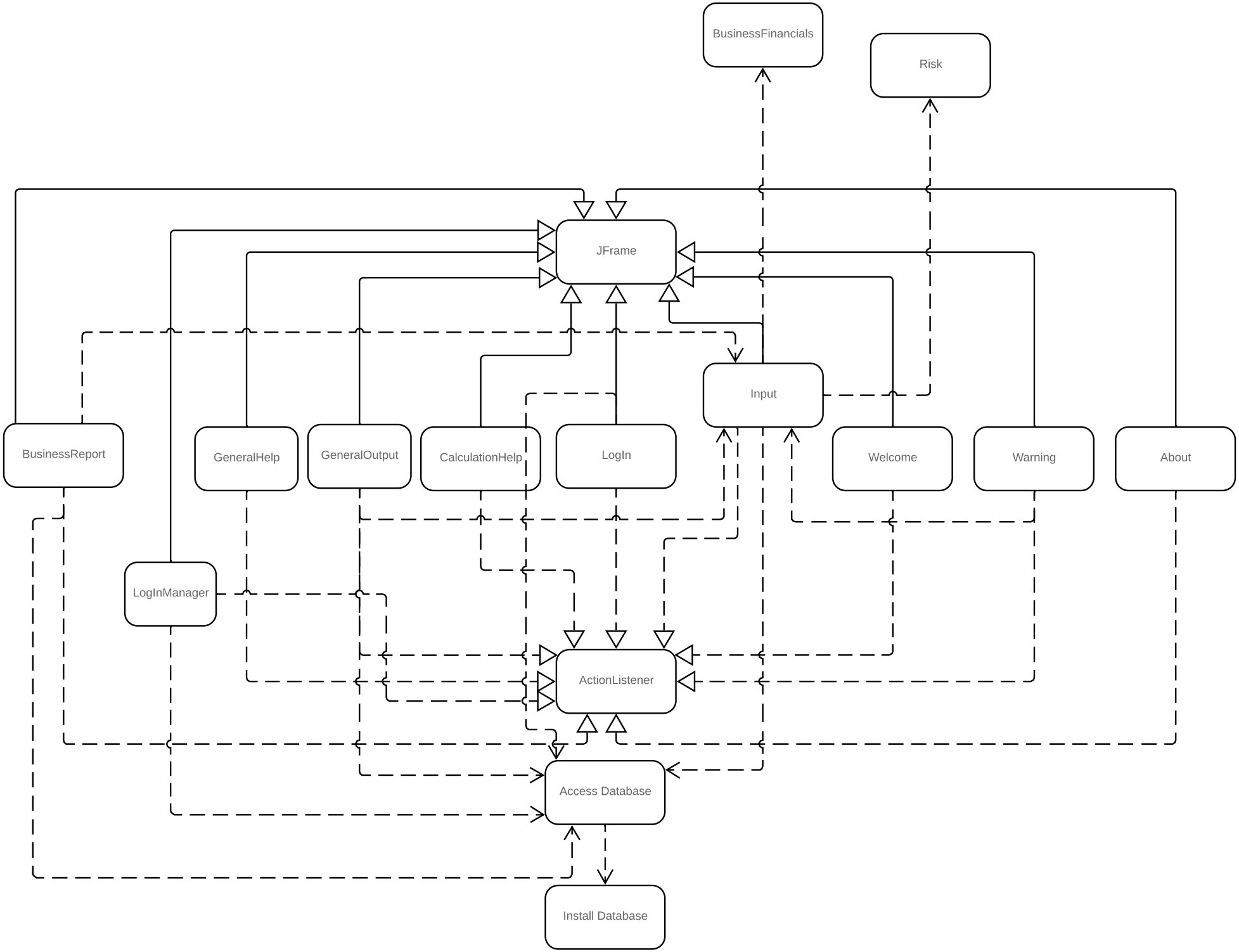
- labels : JLabel
- buttons : JButton
- panels : JPanel
- tables : JTable
- scrollPanes : JScrollPane
- headers : JTableHeader
- columns: TableColumn
- data : Object[][]
- TABLE_HEADER: String[]

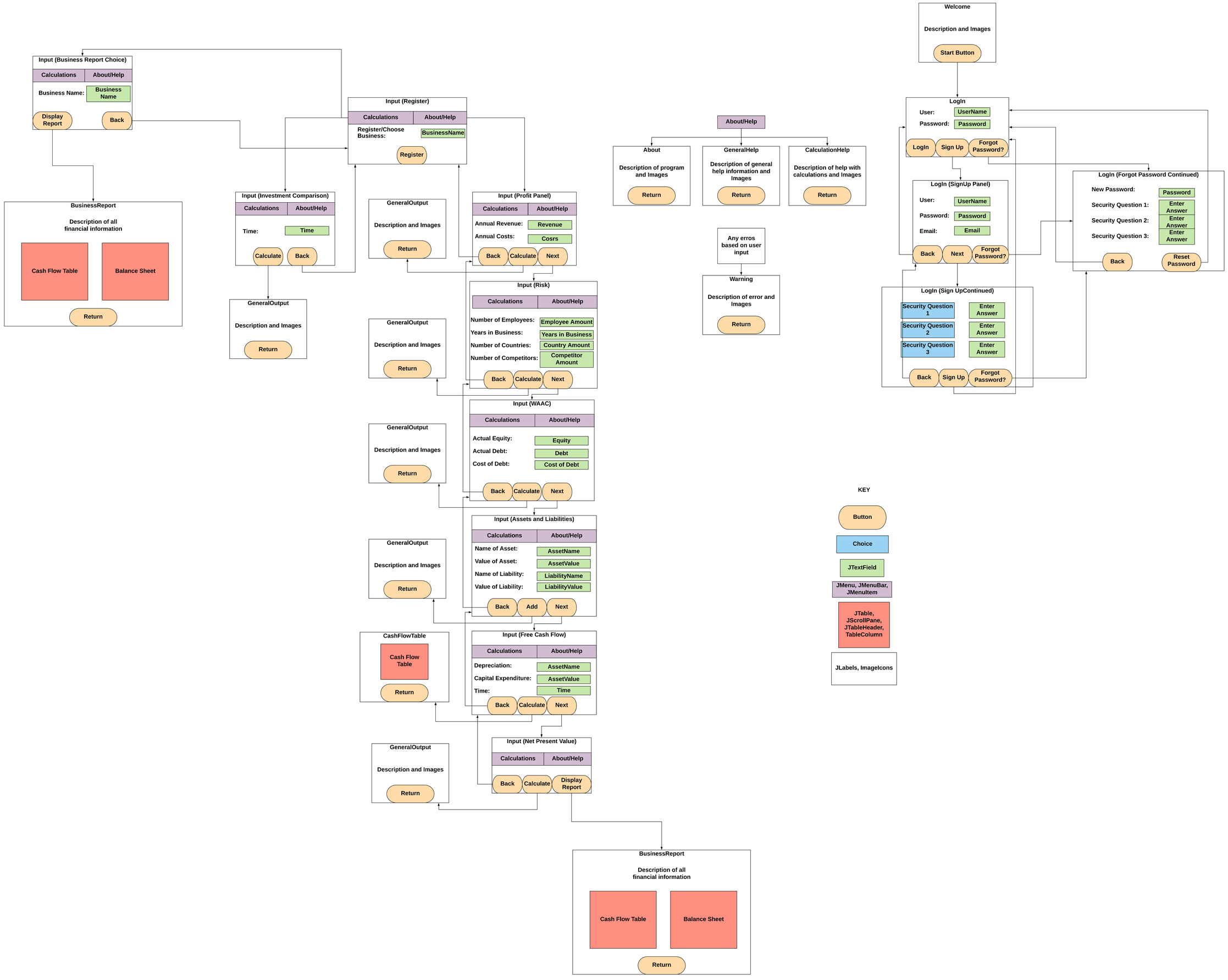
+ BusinessReport(Object[][] data) :
+ actionPerformed() : void

CashFlowTable

- labels : JLabel
- buttons : JButton
- panels : JPanel
- tables : JTable
- scrollPanes : JScrollPane
- headers : JTableHeader
- columns: TableColumn
- data : Object[][]
- TABLE_HEADER: String[]

+ CashFlowTable(Object[][] data) :
+ actionPerformed() : void

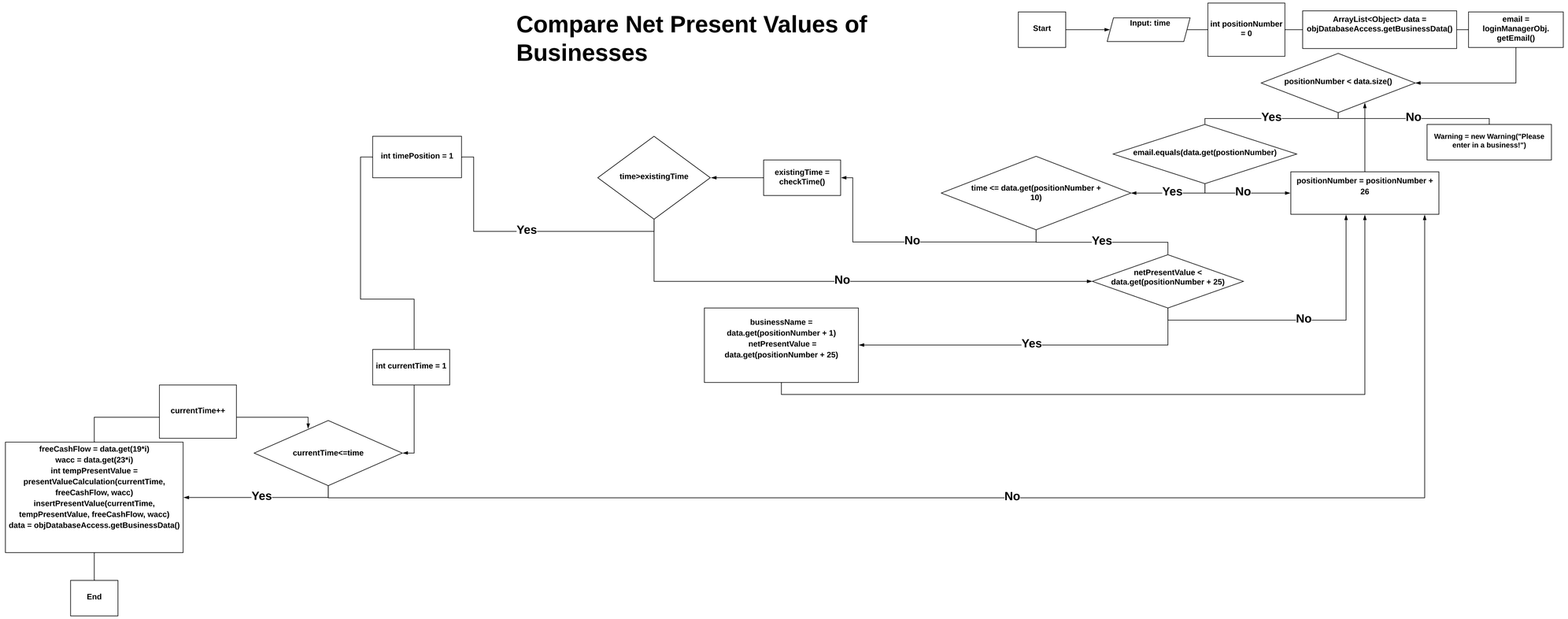

BusinessFinancials
Risk
JFrame
Input
BusinessReport
GeneralHelp
GeneralOutput
CalculationHelp
LogIn
Welcome
Warning
About
LogInManager
ActionListener
Access Database
Install Database


Welcome
Description and Images
Start Button
LogIn
User:
UserName
Password:
Password
LogIn
Sign Up
Forgot Password?
LogIn (SignUp Panel)
User:
UserName
Password:
Password
Email:
Email
Back
Next
Forgot Password?
LogIn (Forgot Password Continued)
New Password:
Password
Security Question 1:
Enter Answer
Security Question 2:
Enter Answer
Security Question 3:
Enter Answer
Back
Reset Password
LogIn (Sign UpContinued)
Security Question 1
Enter Answer
Security Question 2
Enter Answer
Security Question 3
Enter Answer
Back
Sign Up
Forgot Password?
About/Help
About
Description of program and Images
Return
GeneralHelp
Description of general help information and Images
Return
CalculationHelp
Description of help with calculations and Images
Return
Any erros based on user input
Warning
Description of error and Images
Return
Input (Business Report Choice)
Calculations
About/Help
Business Name:
Business Name
Display Report
Back
Input (Register)
Calculations
About/Help
Register/Choose Business:
BusinessName
Register
BusinessReport
Description of all financial information
Cash Flow Table
Balance Sheet
Return
Input (Investment Comparison)
Calculations
About/Help
Time:
Time
Calculate
Back
GeneralOutput
Description and Images
Return
GeneralOutput
Description and Images
Return
Input (Profit Panel)
Calculations
About/Help
Annual Revenue:
Revenue
Annual Costs:
Cosrs
Back
Calculate
Next
GeneralOutput
Description and Images
Return
Input (Risk)
Calculations
About/Help
Number of Employees:
Employee Amount
Years in Business:
Years in Business
Number of Countries:
Country Amount
Number of Competitors:
Competitor Amount
Back
Calculate
Next
GeneralOutput
Description and Images
Return
Input (WAAC)
Calculations
About/Help
Actual Equity:
Equity
Actual Debt:
Debt
Cost of Debt:
Cost of Debt
Back
Calculate
Next
GeneralOutput
Description and Images
Return
Input (Assets and Liabilities)
Calculations
About/Help
Name of Asset:
AssetName
Value of Asset:
AssetValue
Name of Liability:
LiabilityName
Value of Liability:
LiabilityValue
Back
Add
Next
CashFlowTable
Cash Flow Table
Return
Input (Free Cash Flow)
Calculations
About/Help
Depreciation:
AssetName
Capital Expenditure:
AssetValue
Time:
Time
Back
Calculate
Next
GeneralOutput
Description and Images
Return
Input (Net Present Value)
Calculations
About/Help
Back
Calculate
Display Report
BusinessReport
Description of all financial information
Cash Flow Table
Balance Sheet
Return
KEY
Button
Choice
JTextField
JMenu, JMenuBar, JMenuItem
JTable, JScrollPane, JTableHeader, TableColumn
JLabels, ImageIcons

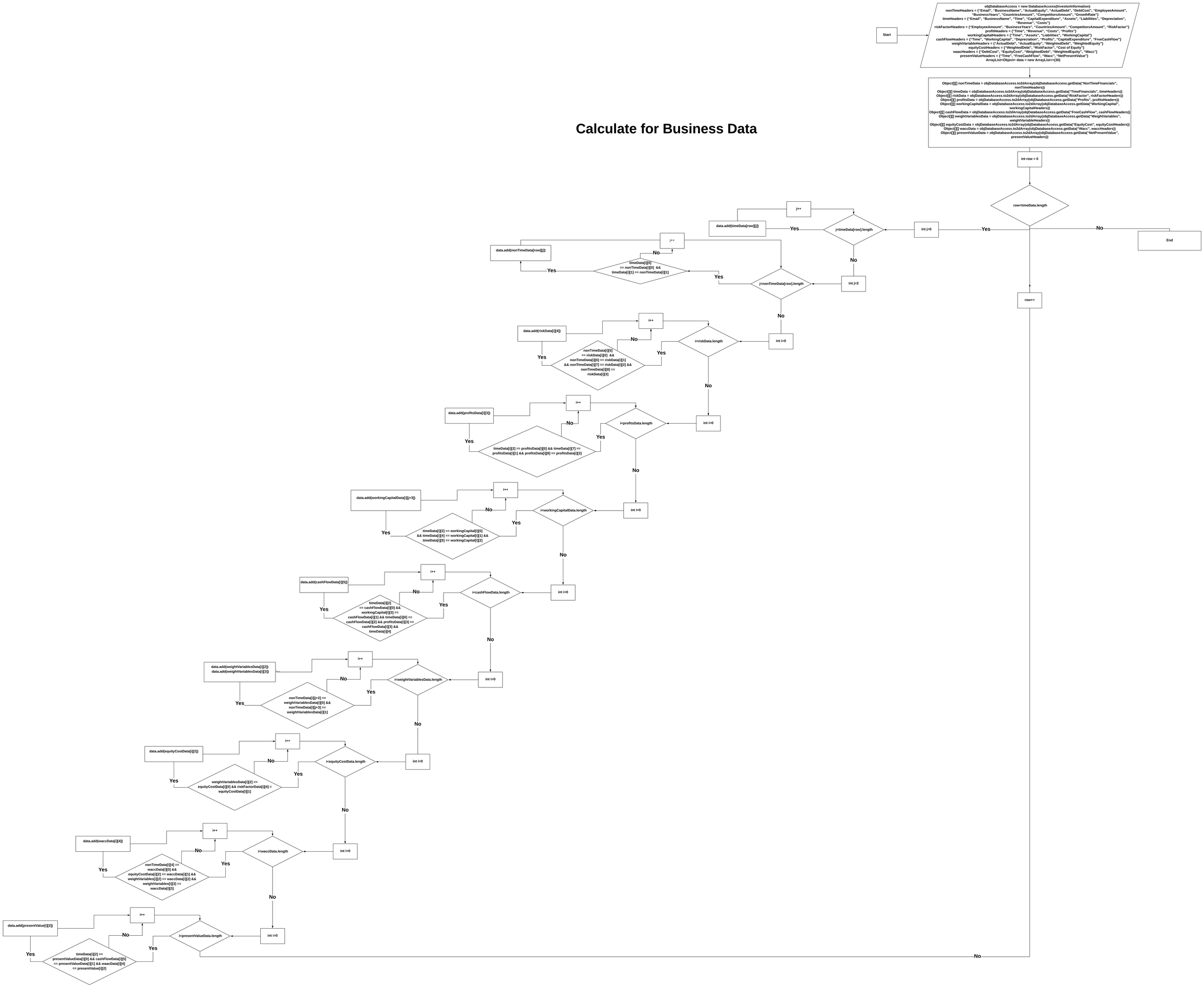
Data Structures:

Data Structure	**Use**
ArrayList <ArrayList<String>>	Holds database table information
ArrayList <Object>	Holds combined information from all the database tables
Object[][] (Array)	Allows for calculations and data from the database tables to be manipulated
String[] (Array)	Stores/Holds column headers for database tables
Object[][] (Array)	Used to display database tables

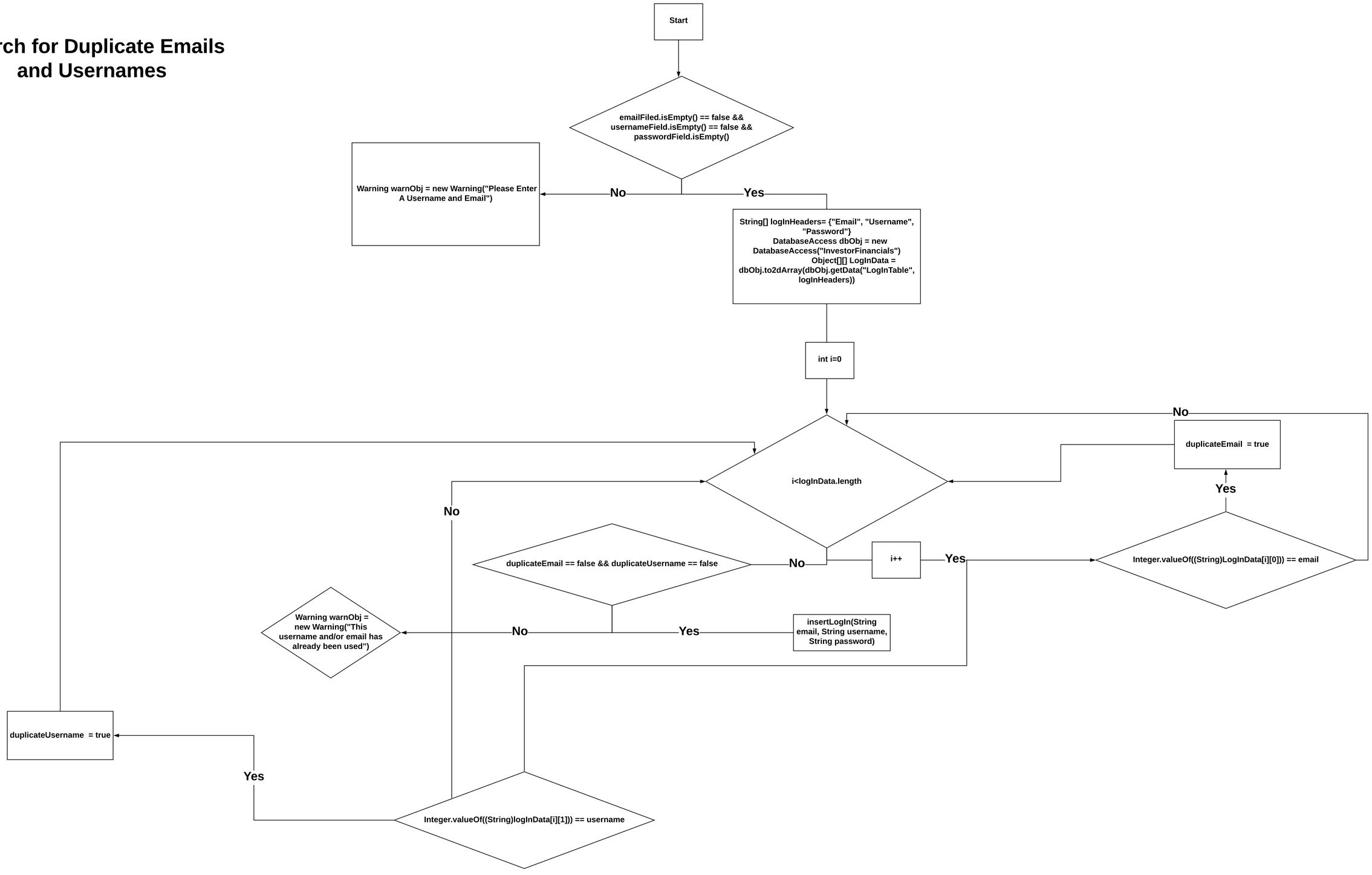
Compare Net Present Values of Businesses



Calculate for Business Data



Search for Duplicate Emails and Usernames



Testing Plan:

Success Criteria	Test Plan
Database must be only accessible by login information, and each new login should add information to the login database table.	Test through printing login database table information and attempting to log in with a user account that was not registered.
System must be able to calculate for the Weighted Average Cost of Capital of a business.	Test through manual multiplication and addition and client approval.
System must be able to calculate for the Risk of a business based on different risk factors.	Test through manual conversion and addition and client approval.
System must be able to calculate for the Working Capital of a business.	Test through manual subtraction and client approval.
Feature to generate a balance sheet with a business's liabilities and assets.	Test through manual calculation, understanding the different assets and liabilities that the client entered, and client approval.
System must be able to calculate the profits of a business for certain times.	Test through manual subtraction and client approval.
Feature a method to calculate and display the annual cash flow of a business.	Test through manual addition and subtraction and client approval.
Feature a method for generating the Net Present Value based on financial information given by the user.	Test through manual multiplication, addition, and division and client approval.
Feature a method to store all this financial information in a database table.	Test through printing all the information within the database table.

Feature a way for the user to update given financial information on a business.	Test through printing all the updated information within the database table.
Feature a method for resetting password if the user forgets their login password.	Test through printing reseted password in login database.
Provide a user-friendly interface that allows for meaningful errors when the user enters incorrect information.	Test through client approval and identifying the different errors that the program provides.
Feature a display of all the previous businesses that were submitted into the account in alphabetical order.	Test through printing all the information within the database table and identifying whether or not they are in alphabetical order.
Allow the user to enter a certain financial investment amount, and present the business that was already logged with the best net present value based on that amount.	Test through calculation of net present value and comparing them to the other businesses that are registered. Also includes client approval.

Amendments

Assets

Email	Business Name	Assets	Asset Value

Liabilities

Email	Business Name	Liabilities	Liability Value

Data Types:

AssetValue: **Double**
LiabilityName: **Double**

Through iteration, these two tables and variables were deemed necessary and thus, they were used

Data Structures:

Data Structure	**Use**
char[]	In order to obtain the password from a password field
ArrayList<Double>	In order to gain the balance sheet and cash flow data for the display

As a result of the iteration process, these two data structures were added as they were necessary for the application's functionality

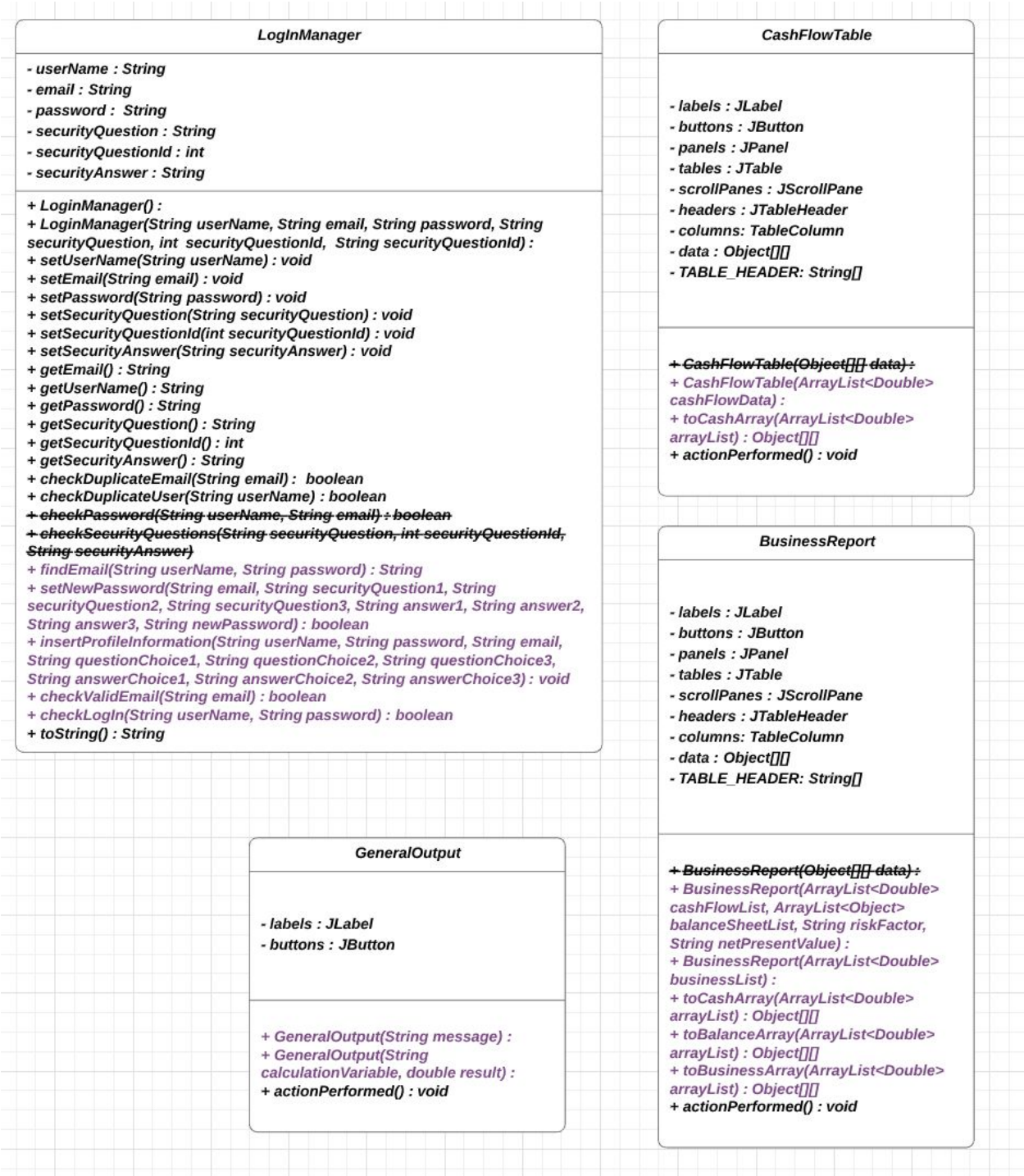


Figure 1: During the implementation process, many different methods were added and removed from these classes in order to add functionality and improve efficiency

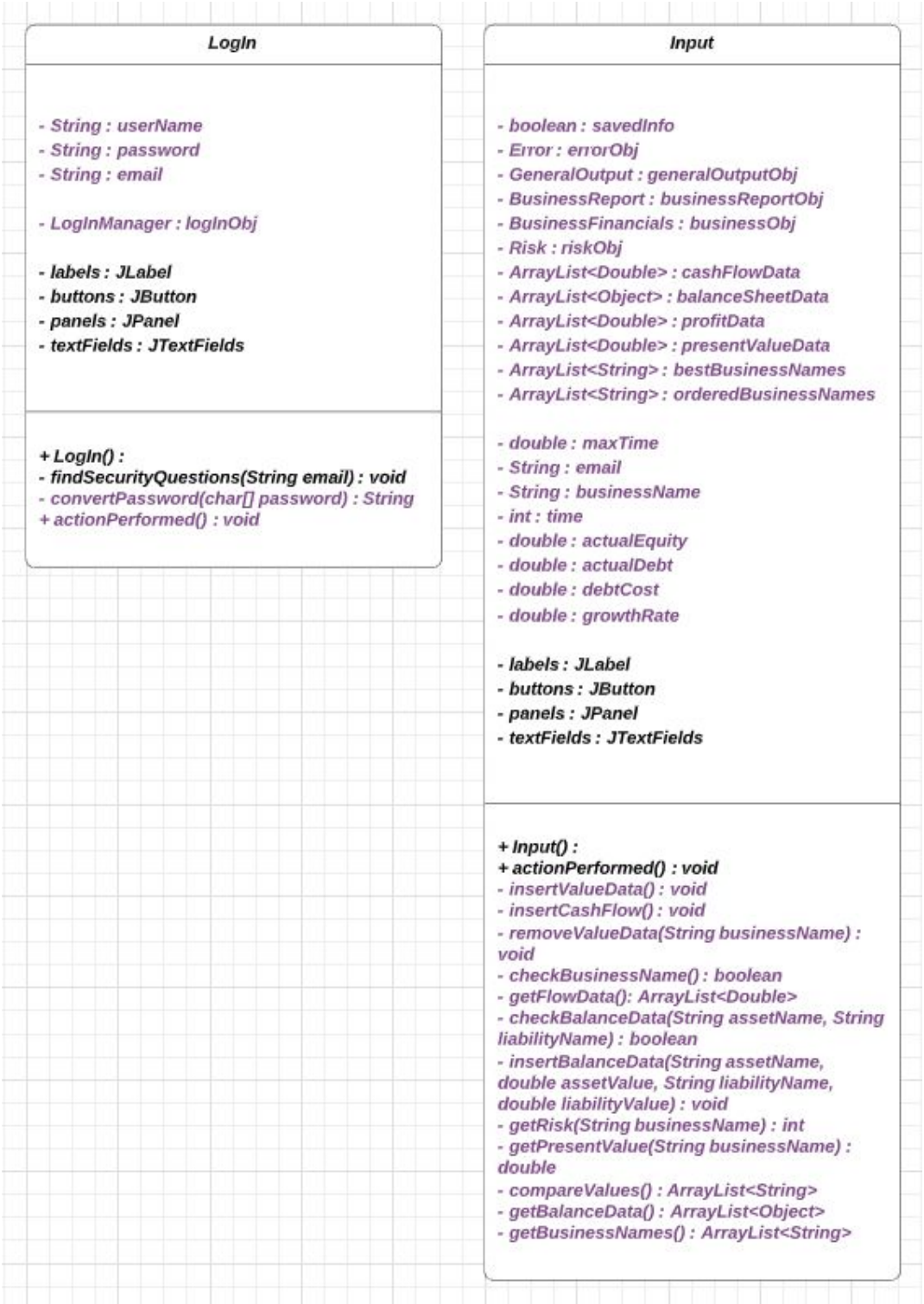


Figure 2: During the implementation process, many different attributes were needed in order to enhance the efficiency and functionality of the product

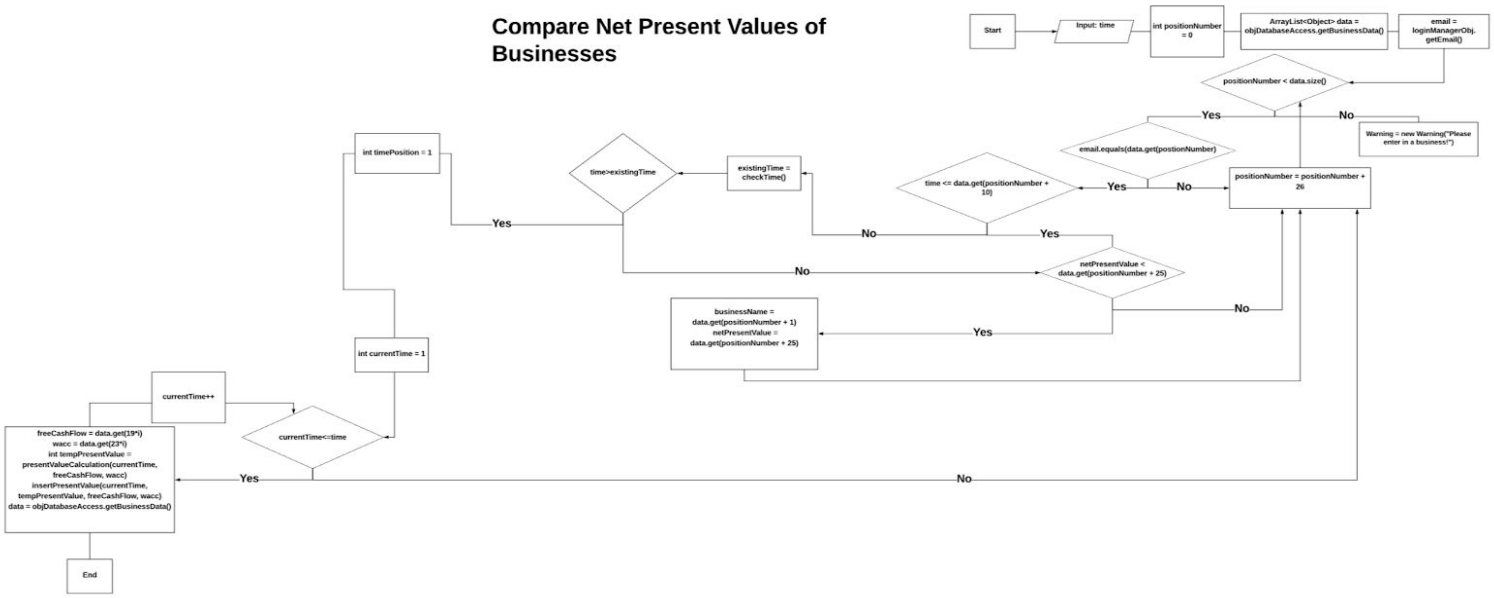


Figure 3: During the iteration process, many of the algorithms were enhanced and changed to allow for a more systematic and effective product

Added attributes and methods are in purple

Removed attributes and methods are ~~crossed out~~

LogInManager

- userName : String
- email : String
- password : String
- securityQuestion : String
- securityQuestionId : int
- securityAnswer : String

+ LoginManager() :
+ LoginManager(String userName, String email, String password, String securityQuestion, int securityQuestionId, String securityQuestionId) :
+ setUserName(String userName) : void
+ setEmail(String email) : void
+ setPassword(String password) : void
+ setSecurityQuestion(String securityQuestion) : void
+ setSecurityQuestionId(int securityQuestionId) : void
+ setSecurityAnswer(String securityAnswer) : void
+ getEmail() : String
+ getUserName() : String
+ getPassword() : String
+ getSecurityQuestion() : String
+ getSecurityQuestionId() : int
+ getSecurityAnswer() : String
+ checkDuplicateEmail(String email) : boolean
+ checkDuplicateUser(String userName) : boolean
~~+ checkPassword(String userName, String email) : boolean~~
~~+ checkSecurityQuestions(String securityQuestion, int securityQuestionId, String securityAnswer)~~
+ findEmail(String userName, String password) : String
+ setNewPassword(String email, String securityQuestion1, String securityQuestion2, String securityQuestion3, String answer1, String answer2, String answer3, String newPassword) : boolean
+ insertProfileInformation(String userName, String password, String email, String questionChoice1, String questionChoice2, String questionChoice3, String answerChoice1, String answerChoice2, String answerChoice3) : void
+ checkValidEmail(String email) : boolean
+ checkLogIn(String userName, String password) : boolean
+ toString() : String

LogIn

- String : userName
- String : password
- String : email

- LogInManager : logInObj

- labels : JLabel
- buttons : JButton
- panels : JPanel
- textFields : JTextFields

+ LogIn() :
- findSecurityQuestions(String email) : void
- convertPassword(char[] password) : String
+ actionPerformed() : void

Input

- boolean : savedInfo
- Error : errorObj
- GeneralOutput : generalOutputObj
- BusinessReport : businessReportObj
- BusinessFinancials : businessObj
- Risk : riskObj
- ArrayList<Double> : cashFlowData
- ArrayList<Object> : balanceSheetData
- ArrayList<Double> : profitData
- ArrayList<Double> : presentValueData
- ArrayList<String> : bestBusinessNames
- ArrayList<String> : orderedBusinessNames

- double : maxTime
- String : email
- String : businessName
- int : time
- double : actualEquity
- double : actualDebt
- double : debtCost
- double : growthRate

- labels : JLabel
- buttons : JButton
- panels : JPanel
- textFields : JTextFields

+ Input() :
+ actionPerformed() : void
- insertValueData() : void
- insertCashFlow() : void
- removeValueData(String businessName) : void
- checkBusinessName() : boolean
- getFlowData(): ArrayList<Double>
- checkBalanceData(String assetName, String liabilityName) : boolean
- insertBalanceData(String assetName, double assetValue, String liabilityName, double liabilityValue) : void
- getRisk(String businessName) : int
- getPresentValue(String businessName) : double
- compareValues() : ArrayList<String>
- getBalanceData() : ArrayList<Object>
- getBusinessNames() : ArrayList<String>

BusinessReport

- labels : JLabel
- buttons : JButton
- panels : JPanel
- tables : JTable
- scrollPanes : JScrollPane
- headers : JTableHeader
- columns: TableColumn
- data : Object[][]
- TABLE_HEADER: String[]

~~+ BusinessReport(Object[][] data) :~~
+ BusinessReport(ArrayList<Double> cashFlowList, ArrayList<Object> balanceSheetList, String riskFactor, String netPresentValue) :
+ BusinessReport(ArrayList<Double> businessList) :
+ toCashArray(ArrayList<Double> arrayList) : Object[][]
+ toBalanceArray(ArrayList<Double> arrayList) : Object[][]
+ toBusinessArray(ArrayList<Double> arrayList) : Object[][]
+ actionPerformed() : void

GeneralOutput

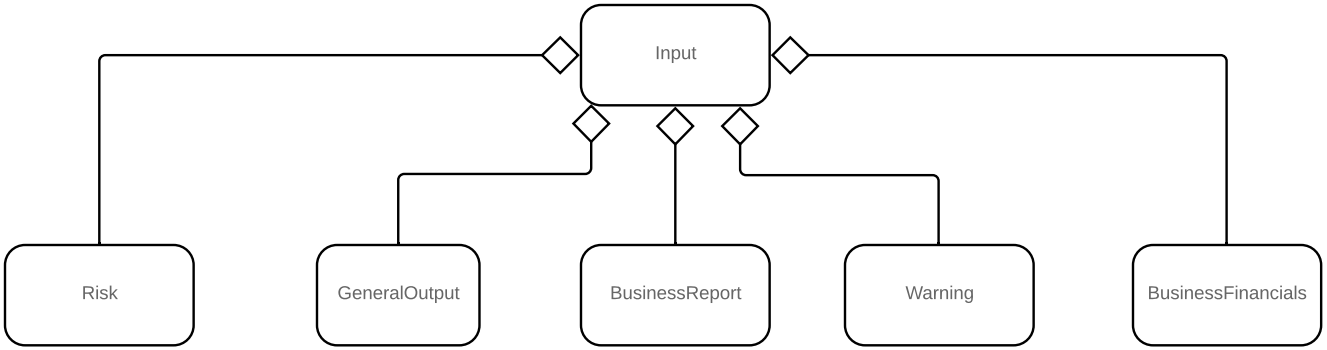
- labels : JLabel
- buttons : JButton

+ GeneralOutput(String message) :
+ GeneralOutput(String calculationVariable, double result) :
+ actionPerformed() : void

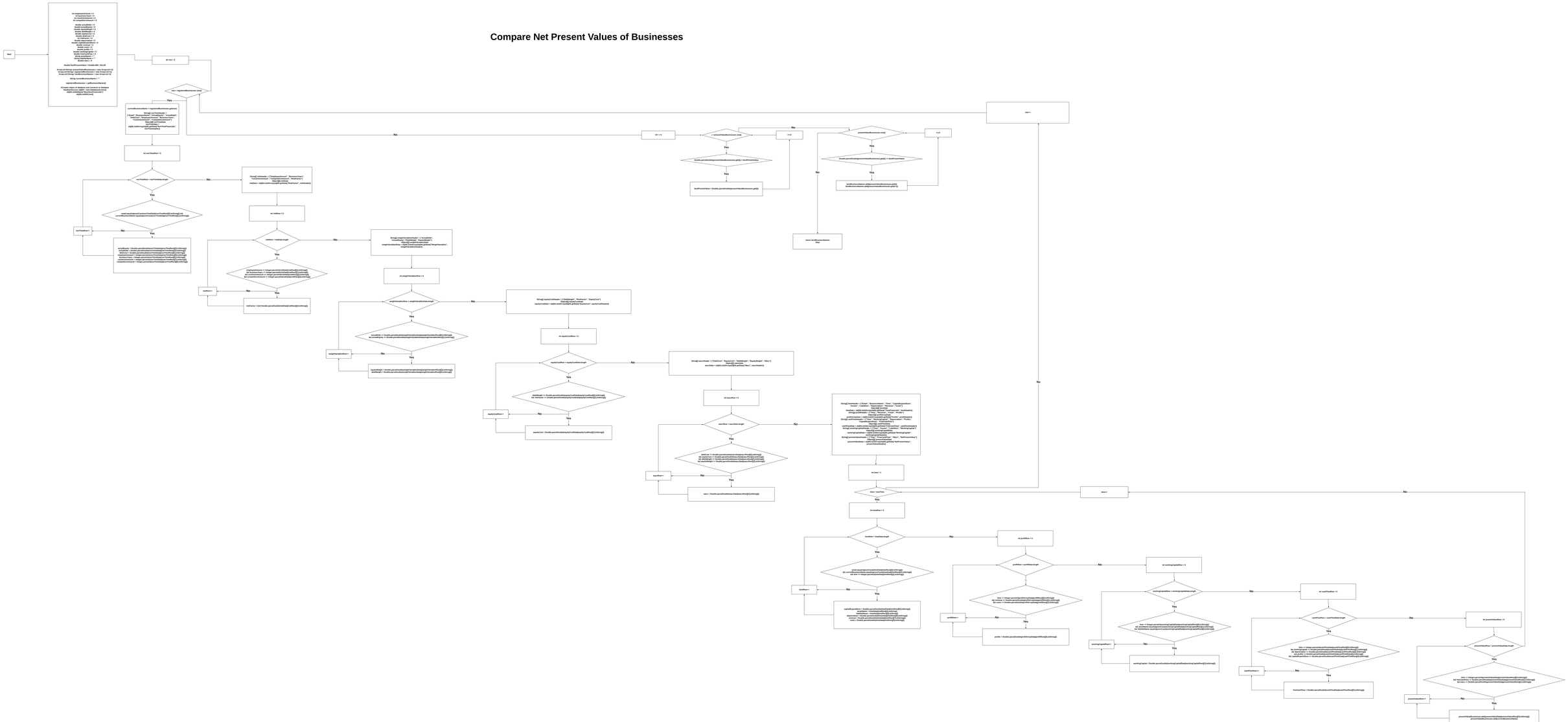
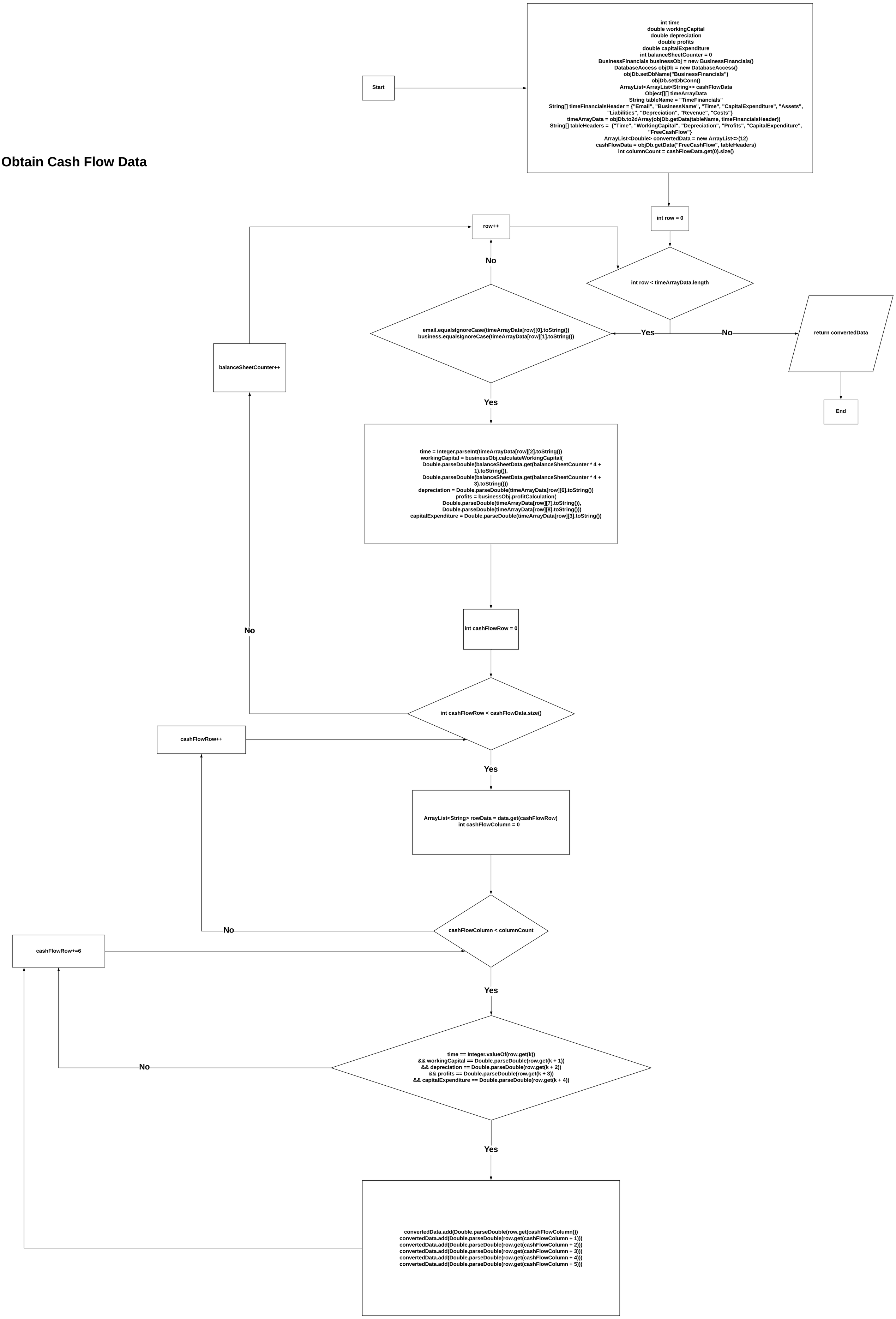
CashFlowTable

- labels : JLabel
- buttons : JButton
- panels : JPanel
- tables : JTable
- scrollPanes : JScrollPane
- headers : JTableHeader
- columns: TableColumn
- data : Object[][]
- TABLE_HEADER: String[]

~~+ CashFlowTable(Object[][] data) :~~
+ CashFlowTable(ArrayList<Double> cashFlowData) :
+ toCashArray(ArrayList<Double> arrayList) : Object[][]
+ actionPerformed() : void


Input
Risk
GeneralOutput
BusinessReport
Warning
BusinessFinancials

Compare Net Present Values of Businesses



Obtain Cash Flow Data