Criterion B: Design

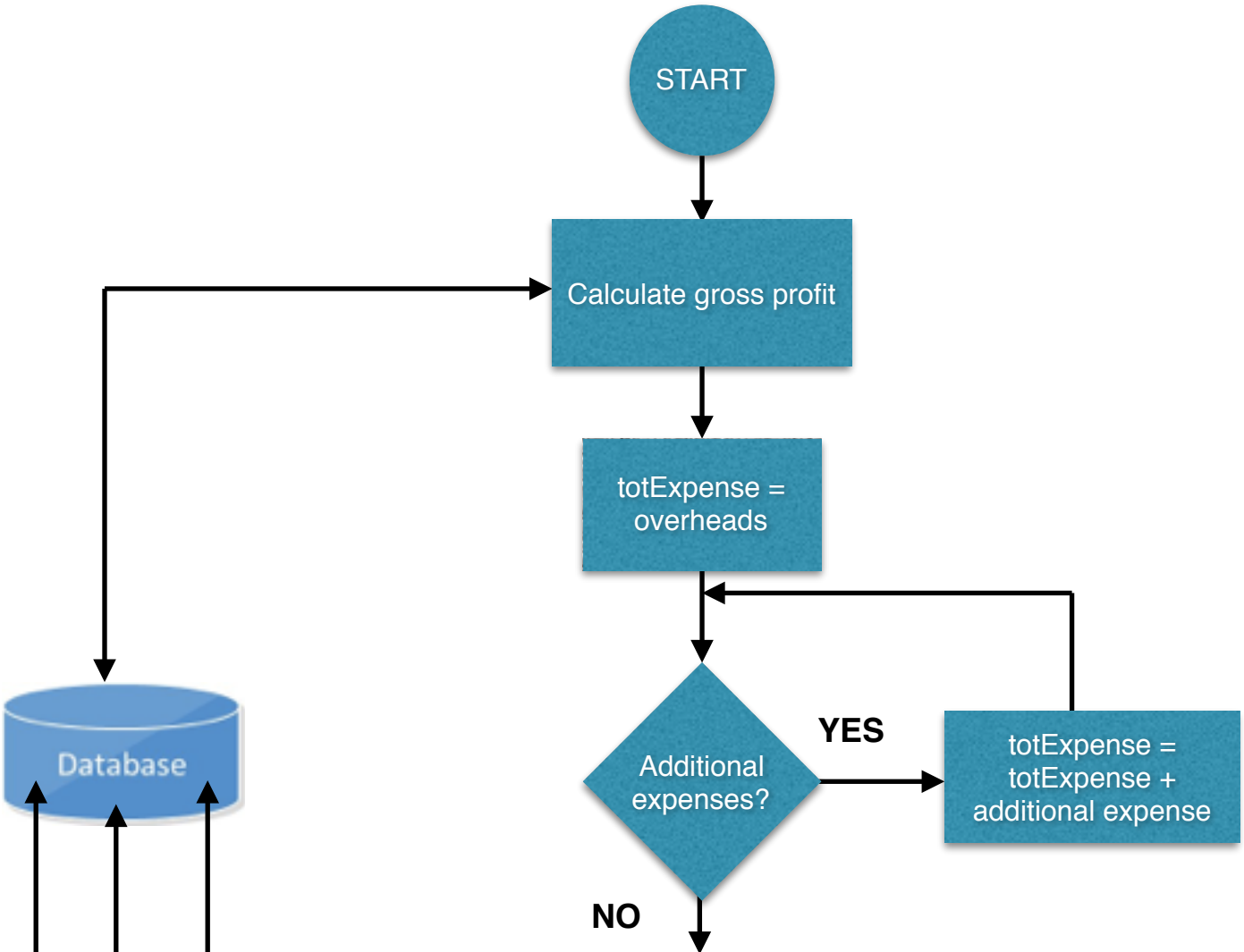
Design Overview

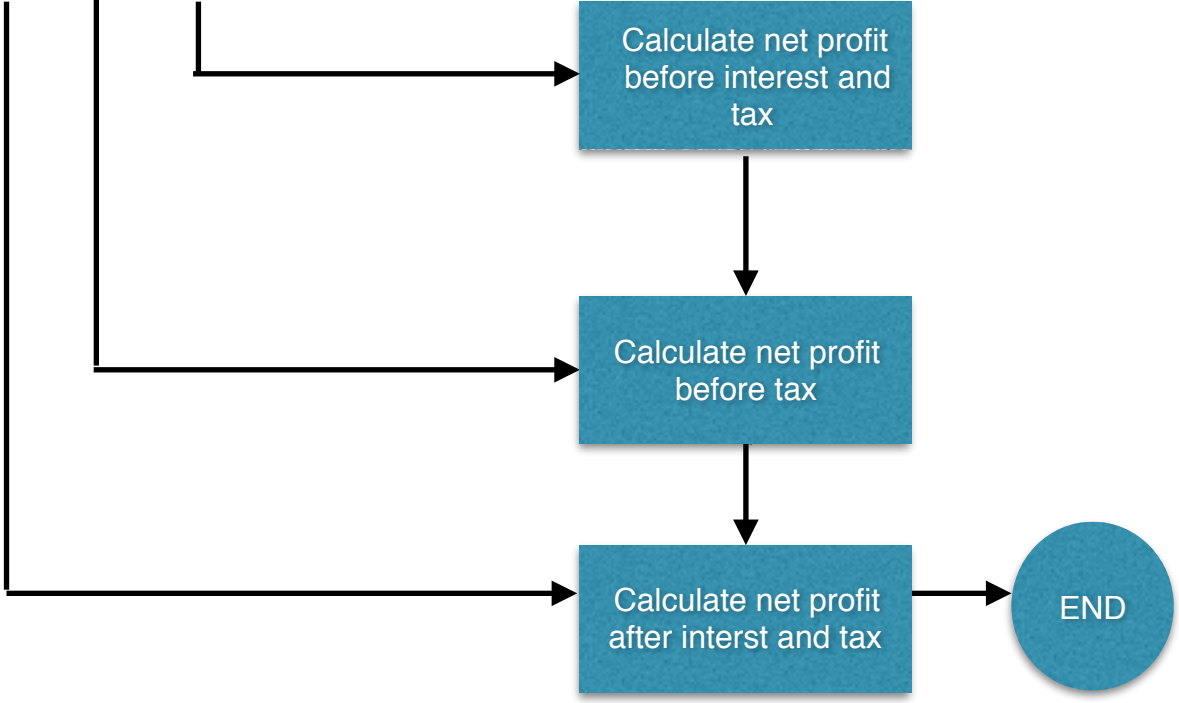
1. Pseudocode

Input sales revenue
Input cost of good sold
Input overhead
Input additional expenses
Input cost of money
Input tax
Calculate income statement (see flowchart)
Display income statement
Calculate gross profit margin + net profit margin
Display gross profit margin + net profit margin
Input target
Calculate suggestion according to target (see flowchart)
Display suggestion
Display income statement

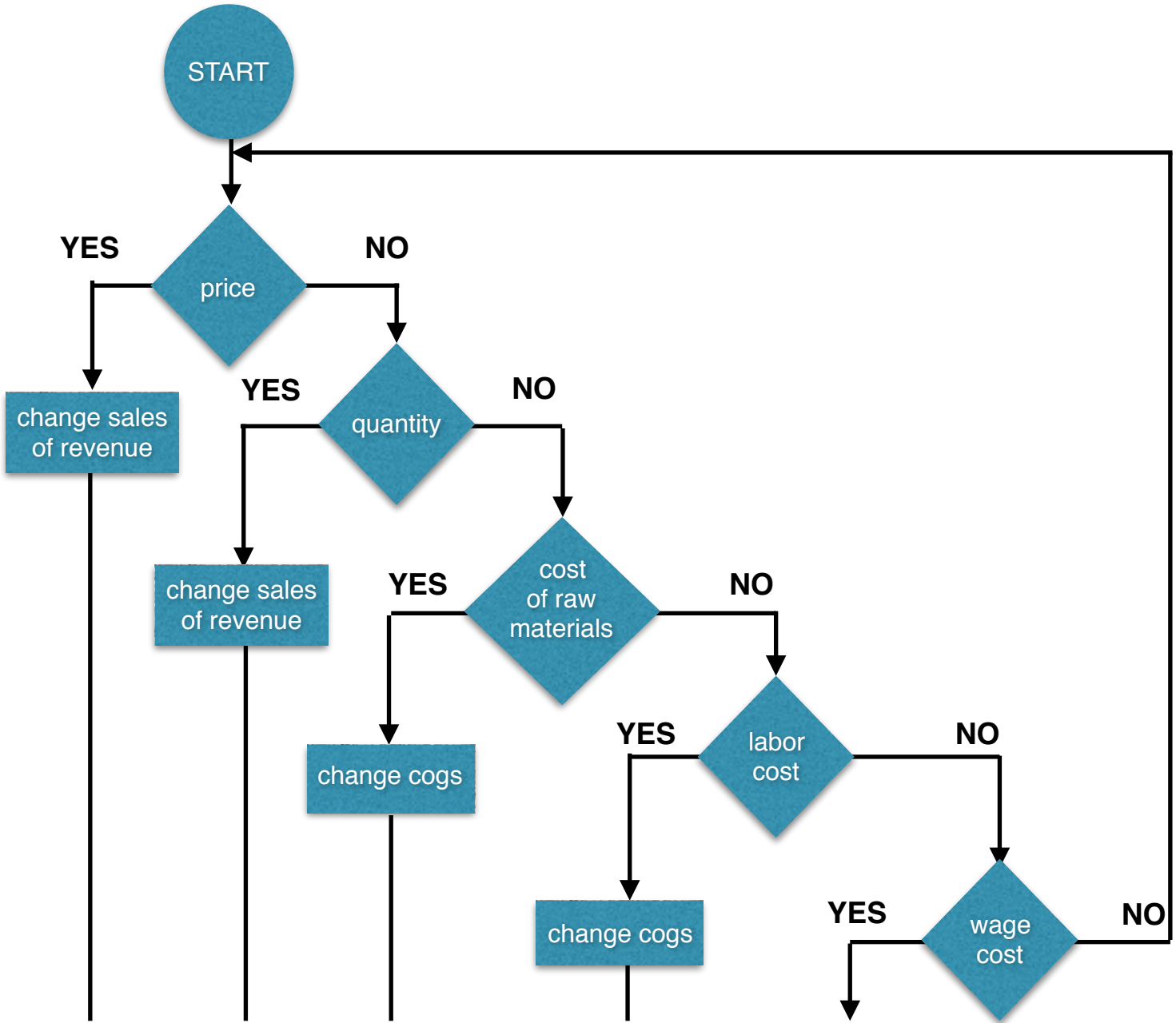
2. Flow Chart

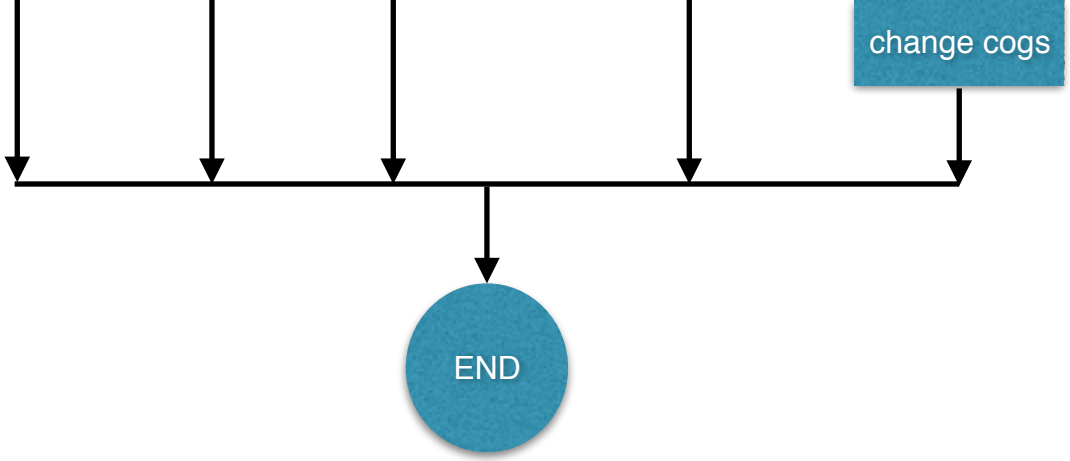
- Calculate Income Statement



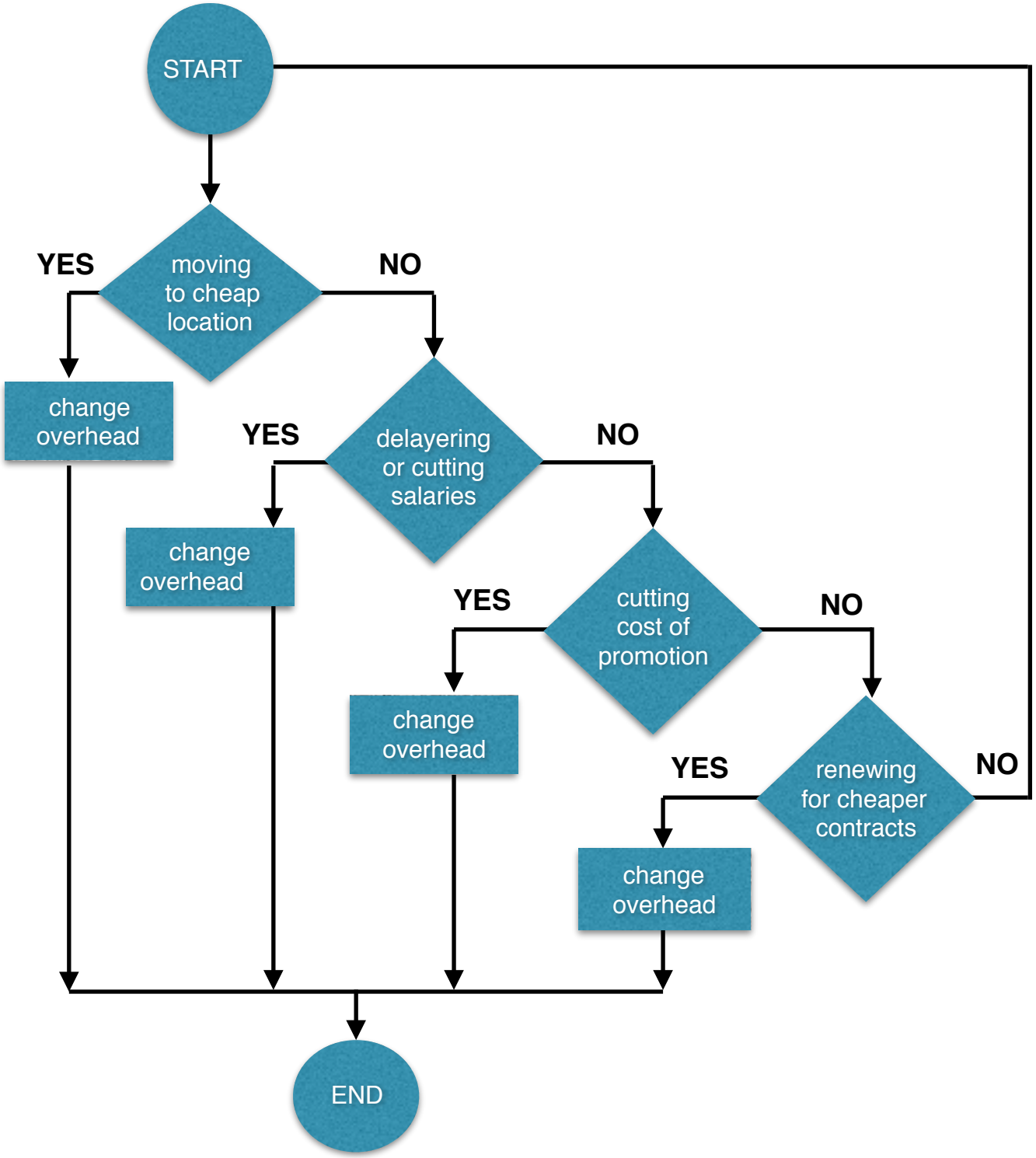


- Calculate suggestion for short term (gross profit margin)
  (cogs = cost of good sold)





- Calculate suggestion for medium term (net profit margin)



3. Database Tables

CompanyName	SalesRevenue	CostOfGoodSold	GrossProfit	Expenses	NetProfitBefore	Interest	NetProfitBeforeTax	Tax	NetProfitAfterInterstAndTax
	40	10	30	13	17	3	14	3	11

Third Normal Form of the table

CompanyName	SalesRevenue	CostOfGoodSold	Expense	Interest	Tax

SalesRevenue	CostOfGoodSold	GrossProfit

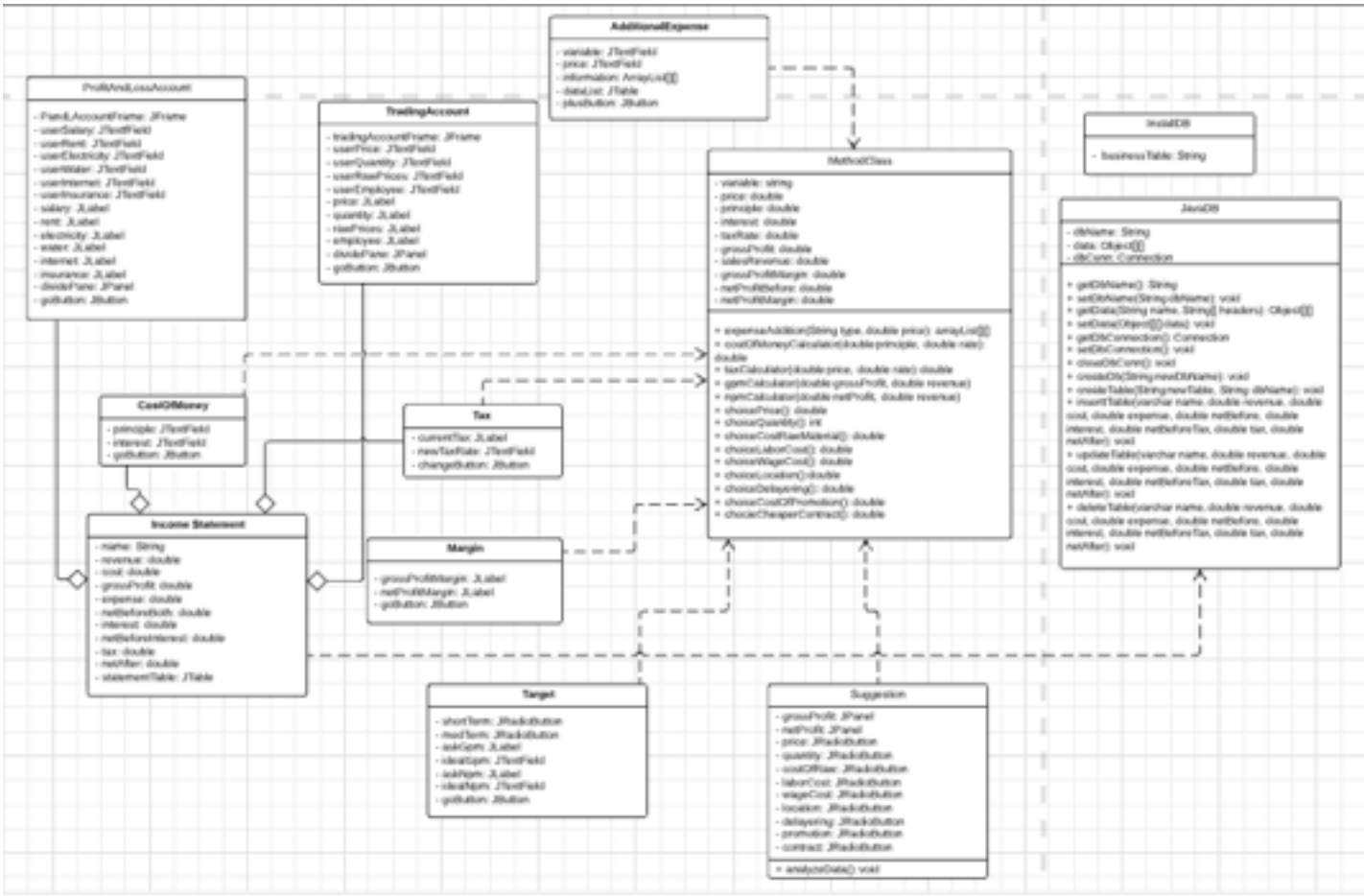
GrossProfit	Expenses	NetProfitBefore

NetProfitBefore	Interest	NetProfitBeforeTax

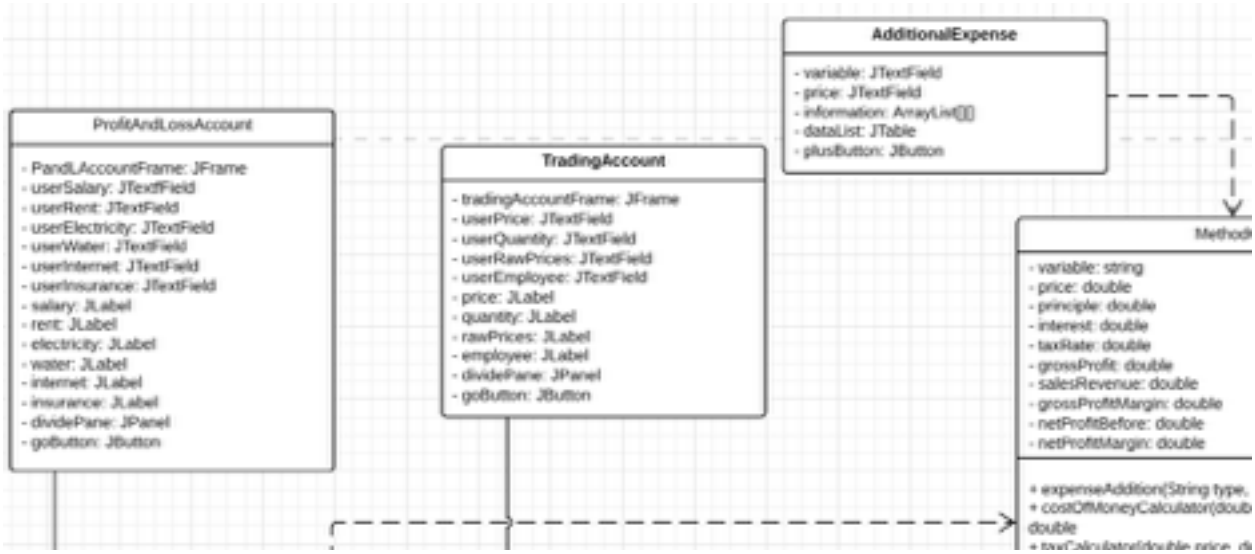
NetProfitBeforeTax	Tax	NetProfitAfterInterstAndTax

4. UML

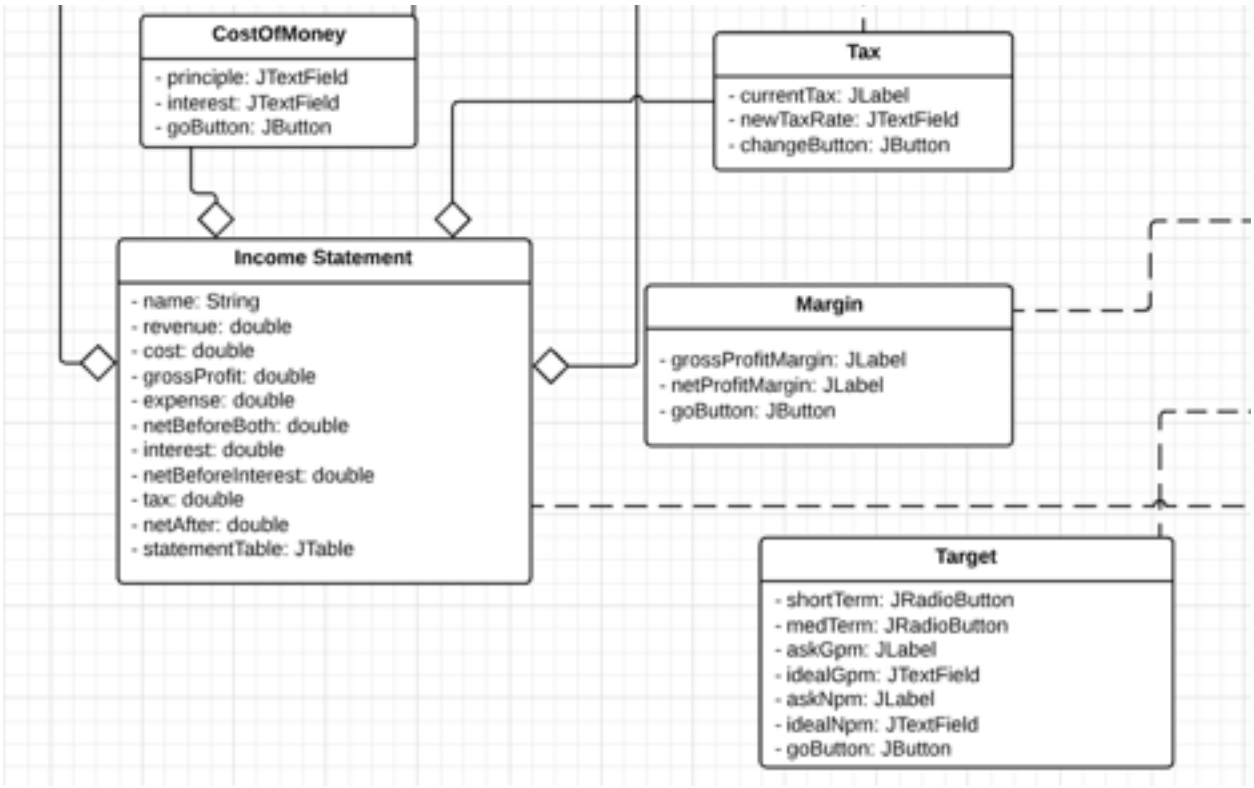
Overview:



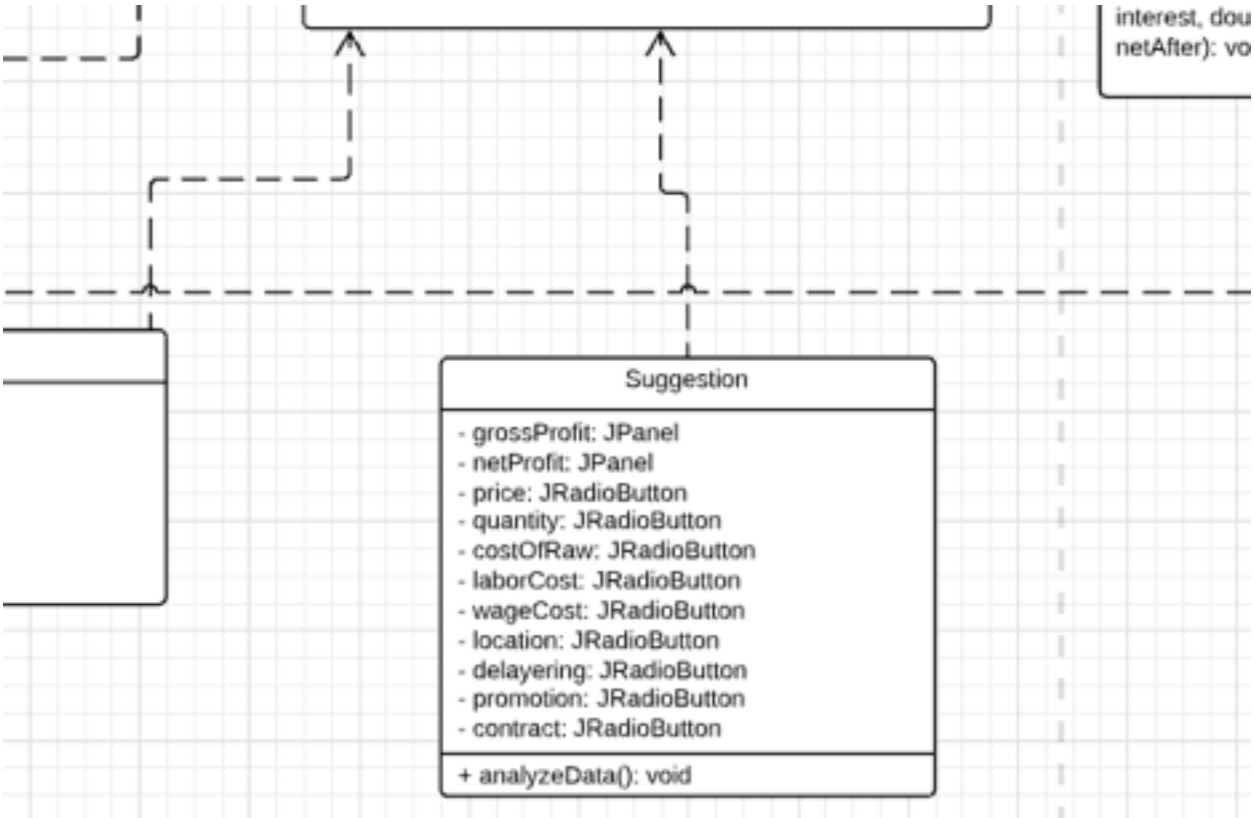
Part 1:



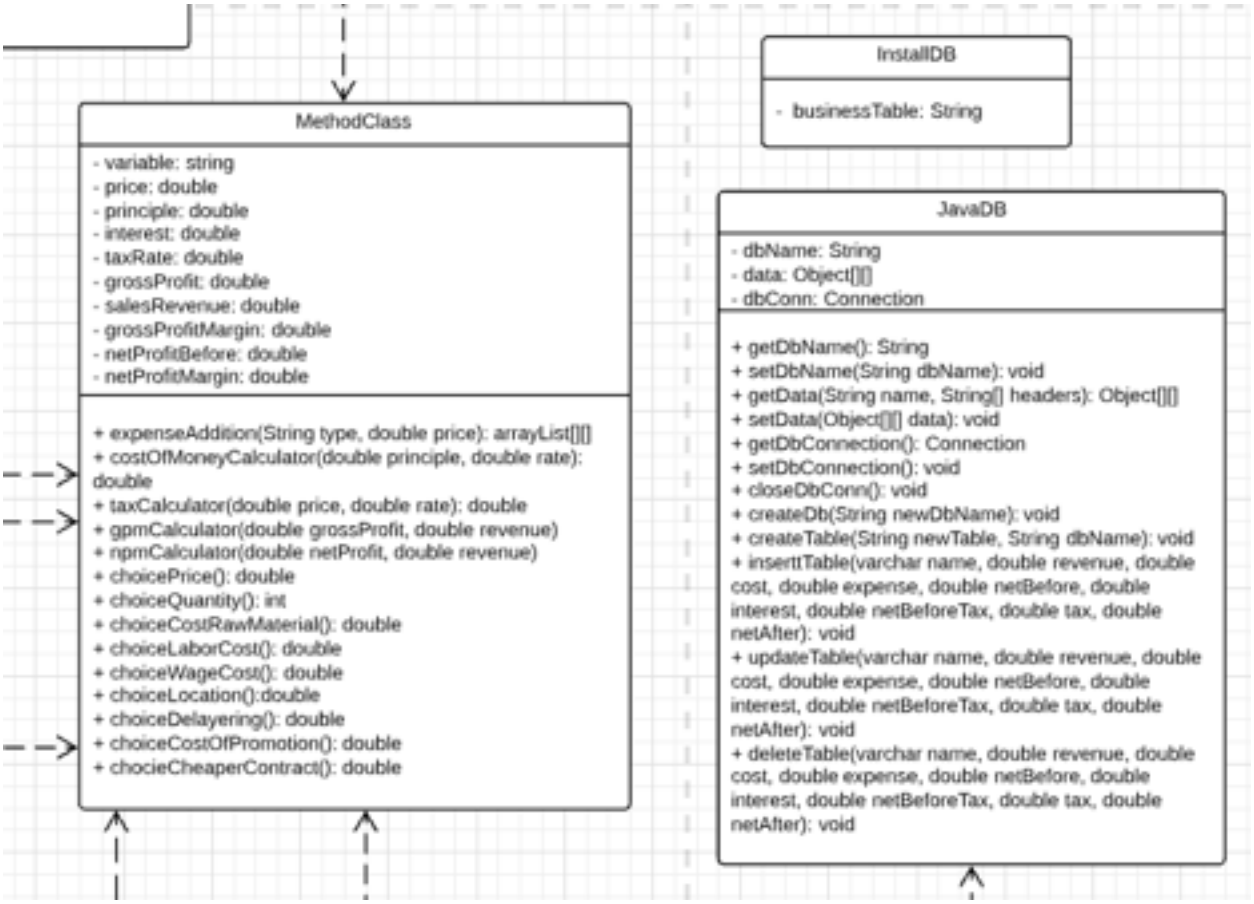
Part 2:


CostOfMoney
- principle: JTextField
- interest: JTextField
- goButton: JButton
Tax
- currentTax: JLabel
- newTaxRate: JTextField
- changeButton: JButton
Income Statement
- name: String
- revenue: double
- cost: double
- grossProfit: double
- expense: double
- netBeforeBoth: double
- interest: double
- netBeforeInterest: double
- tax: double
- netAfter: double
- statementTable: JTable
Margin
- grossProfitMargin: JLabel
- netProfitMargin: JLabel
- goButton: JButton
Target
- shortTerm: JRadioButton
- medTerm: JRadioButton
- askGpm: JLabel
- idealGpm: JTextField
- askNpm: JLabel
- idealNpm: JTextField
- goButton: JButton

Part 3:


Suggestion
- grossProfit: JPanel
- netProfit: JPanel
- price: JRadioButton
- quantity: JRadioButton
- costOfRaw: JRadioButton
- laborCost: JRadioButton
- wageCost: JRadioButton
- location: JRadioButton
- delayering: JRadioButton
- promotion: JRadioButton
- contract: JRadioButton
+ analyzeData(): void

Part 4:



Test Plan

Actions to be Tested	Method of Testing
Database records can be edited through program.	Check database whether there is a changes correspond to user's inputs of financial information in the registration and the additional expenses.
Income Statement table can be generated based on the datas extracted from database.	Check the displayed table whether it extracts the correct data within company or data from correct company.

Actions to be Tested	Method of Testing
Program can catch the out of range error when user inputs their ideal monetary values as nonsense.	Run the program and input the target data which is out of range. Check if it shows you the error message.
Tax is calculated and generated correctly.	Whenever it generates the income statement, check if the table has correct value of tax based on the financial information of the business.
Product is functional.	By installing jar file on the other computer and check if everything is working correctly.
Program can ask users if there is any missing data.	Run the program and check if there is a frame asking the additional expense. Then check if the inputs that you make additionally are also added to the calculation of total expense by watching the displayed income statement.
Users get several options of targets to choose.	Run the program and check if the program is providing two targets such as short term(gross profit margin) or medium term(net profit margin) properly.
Program show a list of suggestions and give options of which to change.	Run the program and check if the program is suggesting several solutions to achieve their target and options of which to check. According to the choice of option, program needs to make correct changes by itself.
Users can move around the program easily through direction buttons: home button, back button.	Run the program and check if there is home button or back button. Program should be exiting the current frame and properly going to the next frame which it is supposed to go whenever the button is clicked.
Program have an option of going back and changing.	Run the program and check if there is an edit button so that user can go back and make changes for their final projected income statement.