

|  |
| --- |
| Gustav Jones |
| IT Graad 10 PAT – Fase 1 |
| Codehub Marketplace |
| You may add an image / your own logo (not required) |

Table of Contents

[**Define the task** 2](#_Toc145857759)

[Topic: 2](#_Toc145857760)

[Purpose of program: 2](#_Toc145857761)

[Possible solution: 2](#_Toc145857762)

[Scope: 2](#_Toc145857763)

[**User Stories** 2](#_Toc145857764)

[**Acceptance Test – List FOUR Functions** 2](#_Toc145857765)

[**Data Dictionary** 2](#_Toc145857766)

[User defined function 2](#_Toc145857767)

[**Navigation** 3](#_Toc145857768)

[**GUI Design** 3](#_Toc145857769)

[Tab Sheet 1 (User 1): 3](#_Toc145857770)

[INPUT: 3](#_Toc145857771)

[OUTPUT: 3](#_Toc145857772)

[Tab Sheet 2 (User 2): 3](#_Toc145857773)

[INPUT: 3](#_Toc145857774)

[OUTPUT: 3](#_Toc145857775)

[**IPO – Software Design Tool** 4](#_Toc145857776)

[INPUT VALIDATION: 4](#_Toc145857777)

[PROCESSING: 4](#_Toc145857778)

# **Define the task**

## Topic:

A Development Freelancing application. My application will be a link between users looking for software developers to solve a problem or create a system for them and freelance developers that will create software for their needs. The software won't be for a specific company but will be for the individual software developer entrepreneurs that want to sell their services to the user.

## Purpose of program:

To bring customers straight to software developers and cut out the middle man by enabling a system of communication between the customer and the freelance developer. This will speed up new software development time and save more money than alternative solutions.

## Possible solution:

A task based system. This will include a form that the user will fill in from my application that will be stored by the application and then shown to the freelance developer within the application.This application will contain two interfaces. One for the customer (user) and one for the freelance developer.

**User Tabsheets:**

The user's interface will enable them to submit an application idea, time frame, description and specify whether the project is a priority or not. These tasks will then be shown within a checkout section where all the user's scheduled projects are located and will contain extra information such as developer comments, the amount of code lines, the status and the total cost when the project is completed.

**Developer Tabsheets:**

The developer's interface will contain a list of all the scheduled tasks from the customer. The developer will also have access to the project information such as lines of code, project status, developer comments and will be able to see read only information such as the project name, description and calculated cost. The developer will control the individual costs of their services from a separate tab.

## Scope:

This application will not be software used by corporate software companies to give tasks to developers. It will also not be a way for a single developer and single user to interact but is intended to serve multiple users and developers through multiple accounts. This software won't handle the payment of the developer for their services but only the request thereof, thus payment will be done after using the application.

# **User Stories**

User 1:

| **WHO** | **WHAT** | **WHY** |
| --- | --- | --- |
| Developer | Sign up to create a developer account | To use the developer's interface of the application |
| Developer | Select a project created by a user to complete | To create a software application for the user's needs |
| Developer | Change his prices for price per line of code, his consult fee and priority cost | To give the user an accurate cost of what the service will cost |
| Developer | Change his account password | To change his log in details in case of a security breach |

User 2:

| **WHO** | **WHAT** | **WHY** |
| --- | --- | --- |
| Customer (User) | Sign up to create a user account | To use the user's interface of the application |
| Customer (User) | Apply for a developer to complete his thought out application idea | To submit a request for the developer to complete the project |
| Customer (User) | Select requested projects and display application information | To see the application status, lines of code and developer comments |
| Customer (User) | Change his account password | To change his log in details in case of a security breach |

# **Acceptance Test – List FOUR Functions**

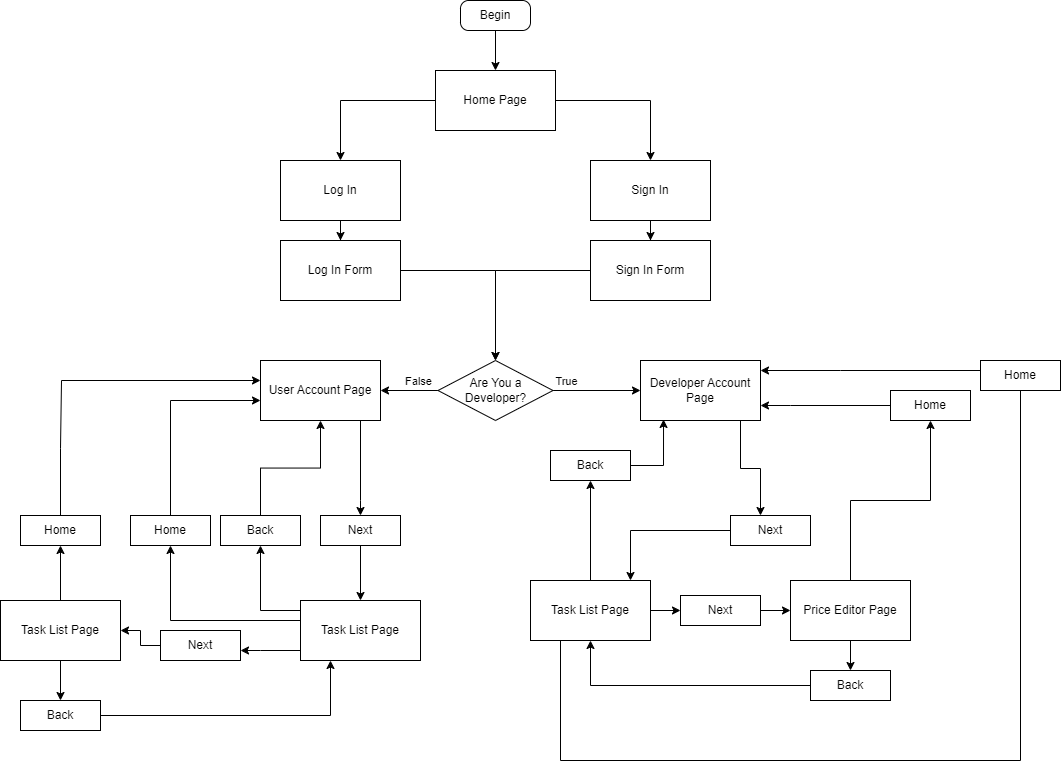
* The system will enable Users and Developers to be able to use unique usernames and passwords to Login and Sign up
* The system will store relevant user and developer information in a text file to create a flow between user and developer
* The system will enable the User to apply for a task the developer needs to solve for them by creating an application
* The system will enable the Developer to access these tasks and can track it's size (lines of code) and set their own pricing as needed

# **Data Dictionary**

## User defined function

|  |  |
| --- | --- |
| **Signature** | **Explanation** |
| GetTask(sTaskName : String) | Will extract the corrisponding task information such as it's Description, Progress and Lines of Code from a text file by referencing it's name |
| SetTask(sTaskName : String, TaskDescription : String, IsCompleted : Boolean, LinesOfCode : Integer) | Will write the relevant task information to a text file |

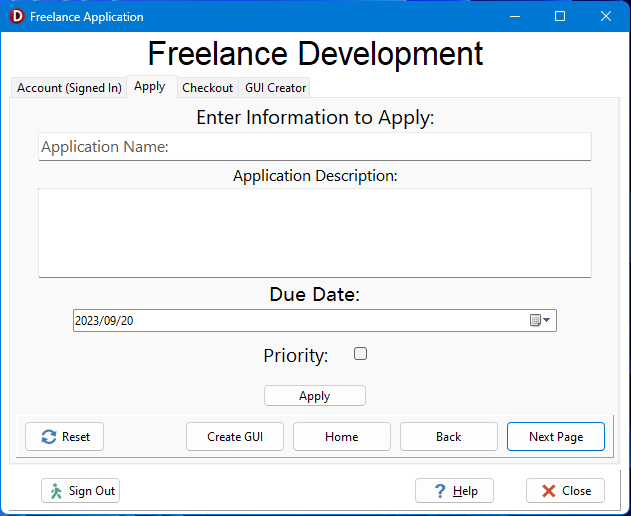
# **Navigation**



# **GUI Design**

## Tab Sheet 1 (User 1):

Screenshot



## INPUT:

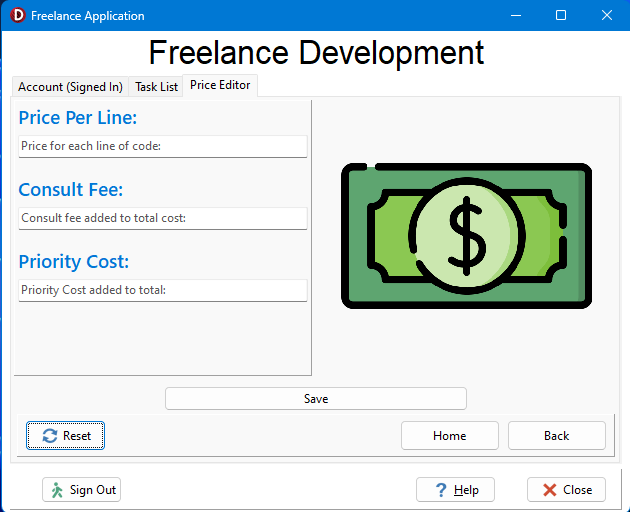
| **tsApply (Apply)** | | | | |
| --- | --- | --- | --- | --- |
| **Source** | **Variable name & data type** | **Format (size, M/F for gender, yyyy/mm/dd)** | **Object Name** | **Local/Global variable** |
| edtApplyTaskName | sTaskName : String | String with no more than 50 characters | Edit | Local |
| redApplyDescription | sDescription :  String | String that consists of at least one line | Richedit | Local |
| dtpApplyDueDate | dDueDate : Date | Date object to store the due date from picker | DateTimePicker | Local |
| chkApplyPriority | bPriority : Boolean | Boolean from checkbox indication priority true/false | Checkbox | Global |

## OUTPUT:

|  |  |  |
| --- | --- | --- |
| **tsCheckout (Checkout)** | | |
| **Output** | **Format (Type, size)** | **Component** |
| Task Information | Caption: <Task Name without Description as a String> | Listbox |
| Total Cost | Total Cost: <Cost of Task with Consult Fee as a Real formatted as a currency> | Label |
| Status | Status: <Status 'In Progress'/'Completed' as a String | Label |
| Lines Of Code | Lines Of Code: <Amount of lines as an Integer> | Label |
| Developer Comments | <Comments from the developer in lines of Strings> | Richedit |

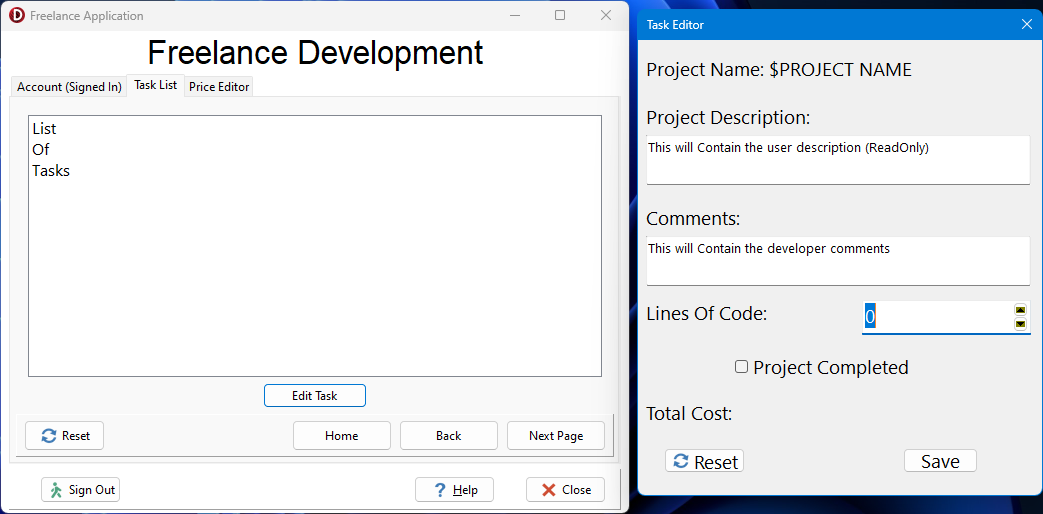
## Tab Sheet 2 (User 2):

Screenshot



## INPUT:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tsPriceEditor (Price Editor)** | | | | |
| **Source** | **Variable name & data type** | **Format (size, M/F for gender, yyyy/mm/dd)** | **Object Name** | **Local/Global variable** |
| edtPriceEditor | rPricePerLine : Real | Real with only a max of 2 desimal points formatted as a currency | Edit | Global |
| edtPriceEditorConsultFee | rConsultFee : Real | Real with only a max of 2 desimal points formatted as a currency | Edit | Global |
| edtPriceEditorPriorityCost | rPriorityCost : Real | Real with only a max of 2 desimal points formatted as a currency | Edit | Global |



| **tsTaskList (Task List)** | | | | |
| --- | --- | --- | --- | --- |
| **Source** | **Variable name & data type** | **Format (size, M/F for gender, yyyy/mm/dd)** | **Object Name** | **Local/Global variable** |
| redComments | sComments :  String | Lines of Strings that contain the developer comments from the Richedit | Richedit | Local |
| sedLinesOfCode | iLinesOfCode : Integer | Integer of the total amount of code lines | Spinedit | Local |
| chkCompleted | bCompleted : Boolean | True/False value for if the project is completed | Checkbox | Local |

## 

## OUTPUT:

| **tsTaskList (Task List)** | | |
| --- | --- | --- |
| **Output** | **Format (Type, size)** | **Component** |
| Total Cost | Total Cost: <Cost of Task with Consult Fee formatted as a currency> | Label |
| Project Description | <Paragraph formatted in lines from the stored values in the text file> | Richedit |
| Project Name | Project Name: <String from stored value in text file> | Label |

# **IPO – Software Design Tool**

## INPUT VALIDATION:

| **Input** | **Validation technique and description** | **Associated error message** |
| --- | --- | --- |
| iLinesOfCode (Integer) | Range Check: Check if the value is in a certain range (iLinesOfCode > 100) | 'Minimum lines of code not reached! Cannot charge consult fee for less than 100 lines of code' |
| lstTaskListItems.ItemIndex (Integer) | Presence Check: Check if an Item was selected from the lstTaskListItem | 'Please Select an item!' |
| sTaskName (String) | Allowed Character Check: Check if special characters were used in the Task Name | 'Please remove special characters from Application Name' |
| sTaskName (String) | Range Check: Check if the Task Name has at least 1 Character and at most 50 | 'Application Name must be between 1 and 50 characters' |
| TextFile (FILE) | File Existance: Check if a file exists before trying to open and checking if it has contents | 'File Not Found. Please make sure the file exists' |

## PROCESSING:

|  |  |
| --- | --- |
| **What processing will be done** | **Algorithms, formulas, etc.** |
| **List:** | **Algorithms:** |
| 1 Total Cost calculation | iLinesOfCode <- input from text file  rCost <- iLinesOfCode \* rPricePerLine  rTotalCost <- rCost + rConsultFee  if bPriority  rTotalCost <- rTotalCost + rPriorityFee  Display rTotalCost formatted as a currency |
| 2 Sign up of an account and log in | sName <- input from user  sUsername <- input from user  sPassword <- input from user  sPassword2 <- input from user  iAge <- input from user  bIsUser <- input from user  if iAge >= 18  if sPassword = sPassword2  store sName, sUsername, sPassword, bIsUser in text file  if bIsUser  enable tsApply  enable tsCheckout  enable tsGUICreator  enable tsAccountH  disable tsAccount  enable bmbSignOut |
|  |  |