Design

1. Overall Systems Design

1.1. Short Description Of The Main Parts Of The System

- Consultant Interface
    - This will be a simple interface with large hit boxes linking you to the separate parts of the program because the program will be used on phones and tablets.
    - There will be 2 functions from the main program that the consultants can use.
        - Show Projects
        - Alert Project Managers
    - The Show Projects function will show the employee all the projects that the company has completed and are currently undertaking.
    - The Alert Project Managers function will allow the consultants to change project details through a project manager if they find any problems.
- Office Managers Interface
    - This interface is for employees in Element Energy who are higher up in the hierarchy than the consultants.
    - These are the functions that the project managers and up will be able to use:
        - Show Projects
        - Show Clients
        - Show Invoices
        - Show Staff
        - Show Subcontractors
        - Edit Projects
        - Edit Clients
        - Edit Invoices
        - Edit Staff

        - Edit Subcontractors
        - Remove Projects
        - Remove Clients
        - Remove Invoices
        - Remove Staff
        - Remove Subcontractors
        - Override Validation
        - Show Project Statistics
    - The project managers will need to be able to manipulate data for the Clients, Subcontractors and Projects so they can add a new project to the database when a new project is assigned to them.
    - They also are given the functions to add invoices. When an invoice is raised that is related to their project they can link it to their project. They can also assign projects to staff members to show the teams that are working on a project.
    - They also have the edit function in case any data they have entered is incorrect, or if the data has changed such as a member of staff being promoted.
    - They can also remove data that is no longer required. For example if a proposed project is not started it can be removed from the database and the client to keep the system tidy.
    - There will be overrides to validation which office managers will be able to access. This is so that when there is a value for which the program does not accommodate, the program can still be used and no adjustments need be made.
    - Show Project Statistics will show how well the company is doing. Whether they have too many employees working or too few. It will show the total working hours spent over the period of a month and how many employees they would need to cover this if they were all working on it full time.

- Getting Details
    - When looking up a Project the user will be able to sort the data by Project Number, Project Title, Project Manager, Etc. There will also be a search bar so that if they know the job number it will be easy to find the project that they would like to update or find the details of.

- This is also the same for Clients, Staff, Subcontractors and Invoices.

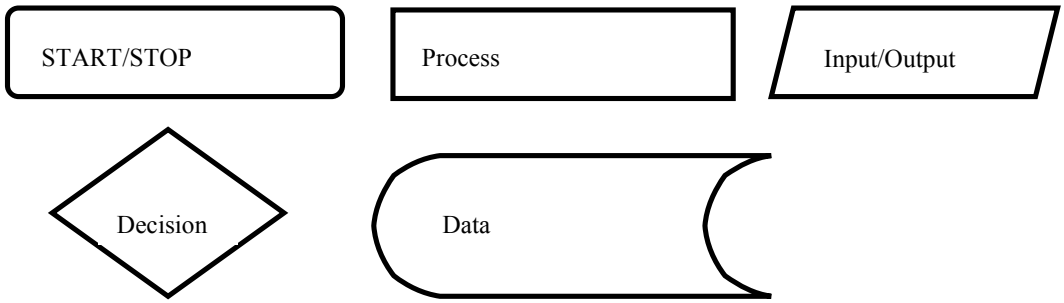
- Inputting Details
  - This will comprise of a form that the user fills in to input data into the project database. The form will ask the details of the project such as: Project Manager, Project Title, Date Start, Value Of Project…
  - All details will be validated. If the validation is too strict on the data the validation can be overridden to allow outliers into the database.
  - This will allow access to the database and modify a project if there is a mistake or if there is a change in the project such as the date moving to a different day.
  - There will also be a way to adjust the project manager list so that you can add or remove project managers.

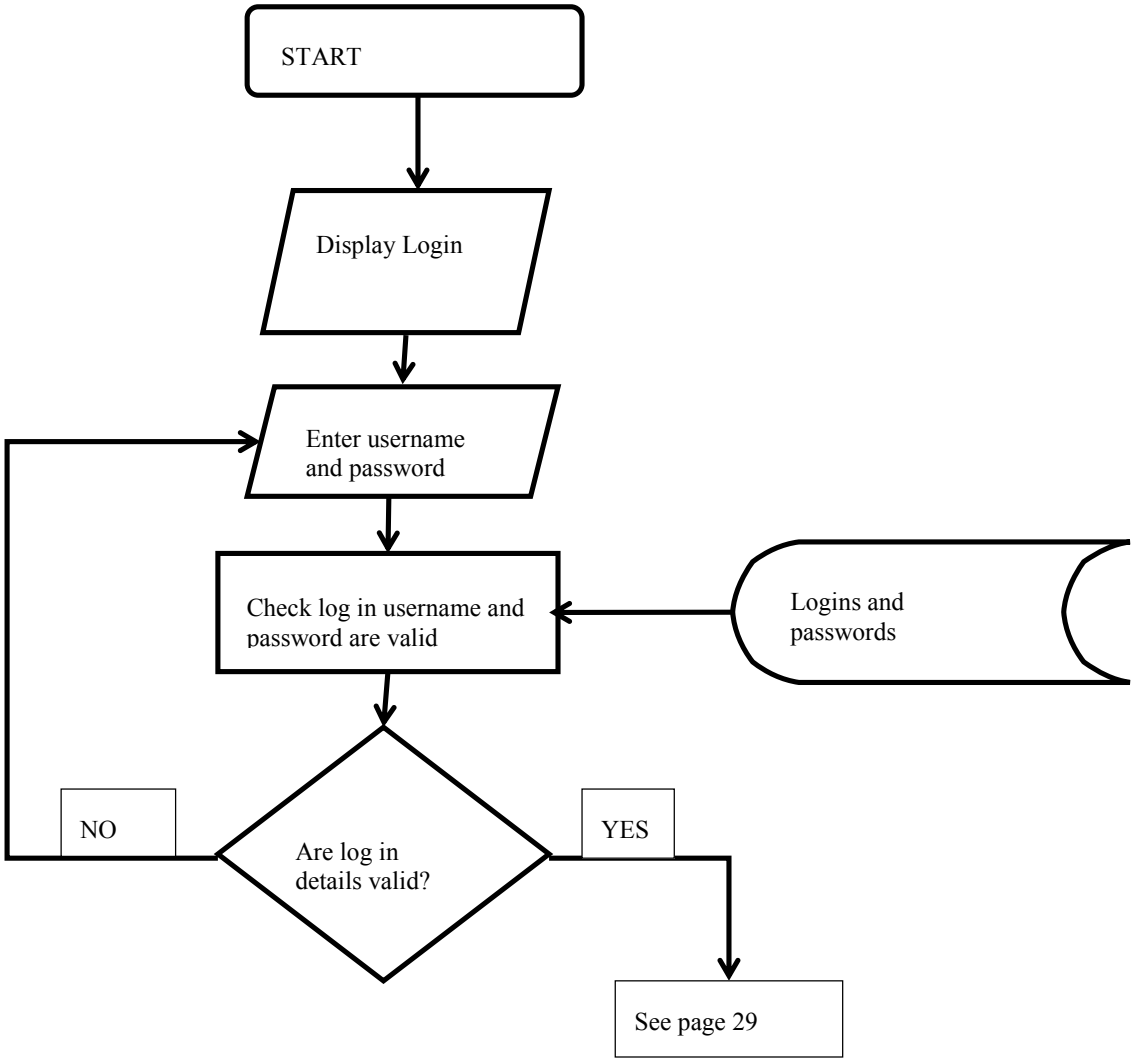
- Monitoring Project Activity
  - There will be a part of the program that checks the end and start dates of the project.
  - The program will notify the consultants who are working on a project when it is starting and when it is about to finish. If the dates have been adjusted it will provide them with a quick link to be able to change the date that the project starts or finishes.

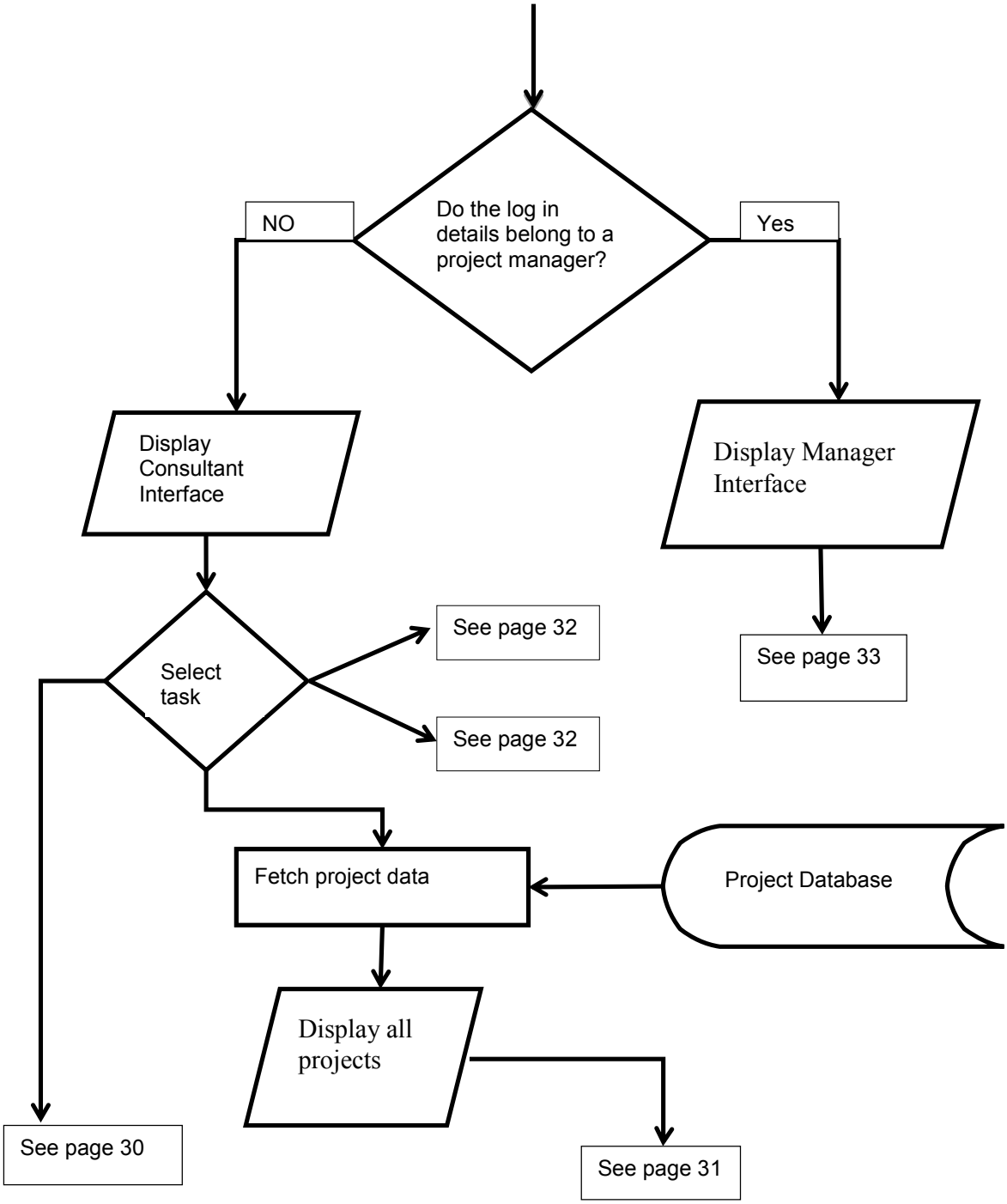
- Reporting Project Changes
  - There will be a program that checks whether there is a change in the data for a project and if there is it will send a mass email to the consultants working on the project so that they are aware of the change.

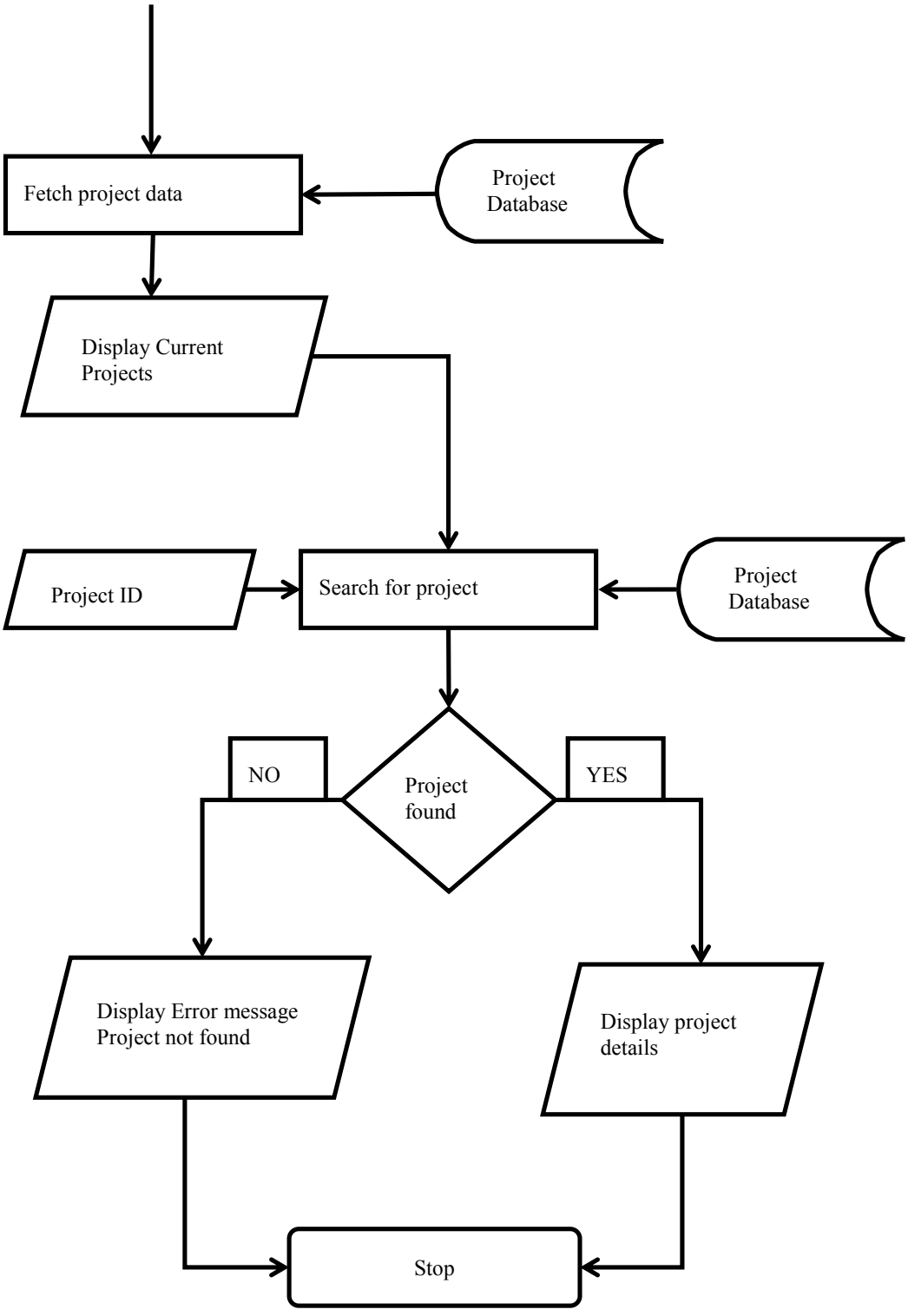
- Mobile and Tablet Access
  - There will be a simplified version of the program for checking the data in the project. There will not be a way for a person to change the data on a project via their phone or tablet to ensure that the data stays safe.

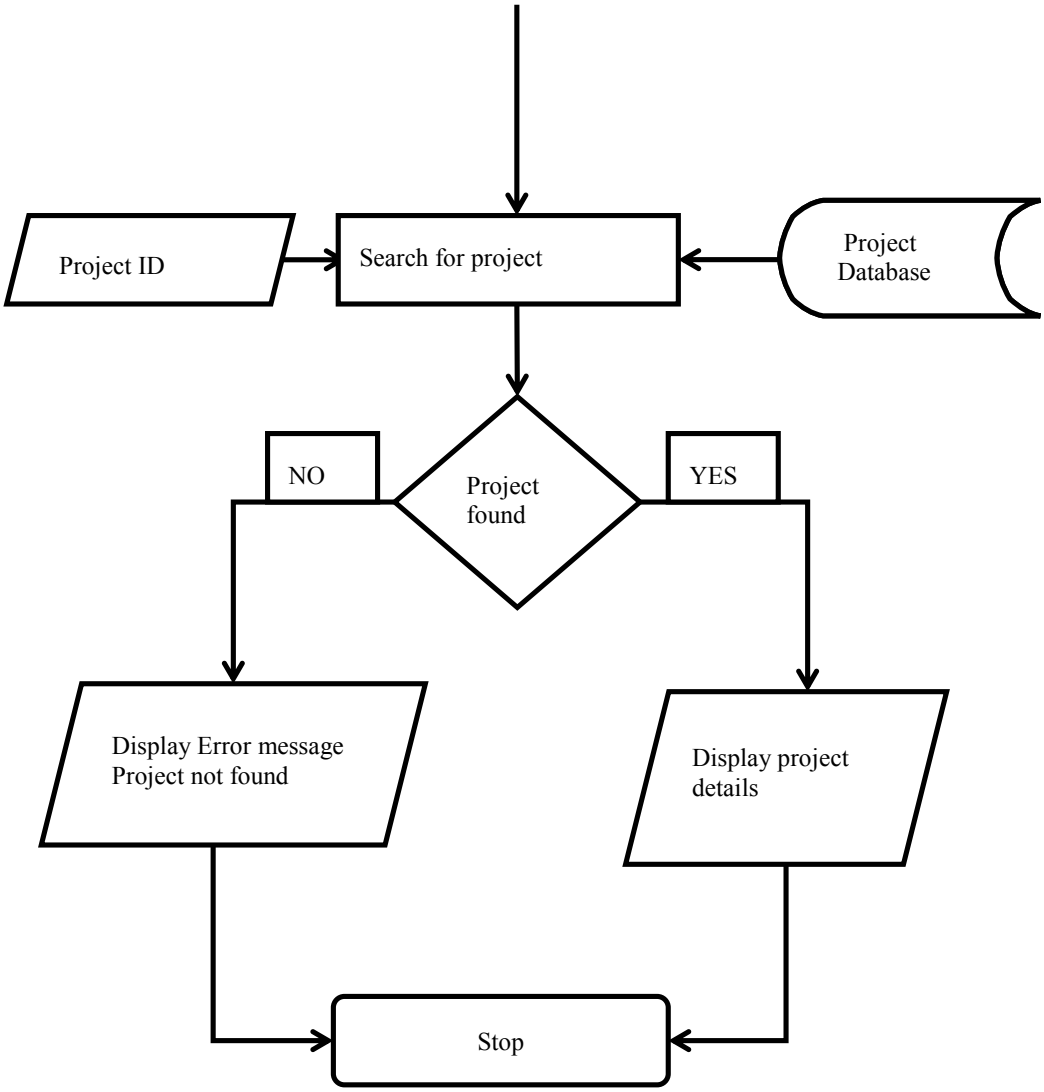
1.2. System Flowcharts Showing An Overview Of The Complete System

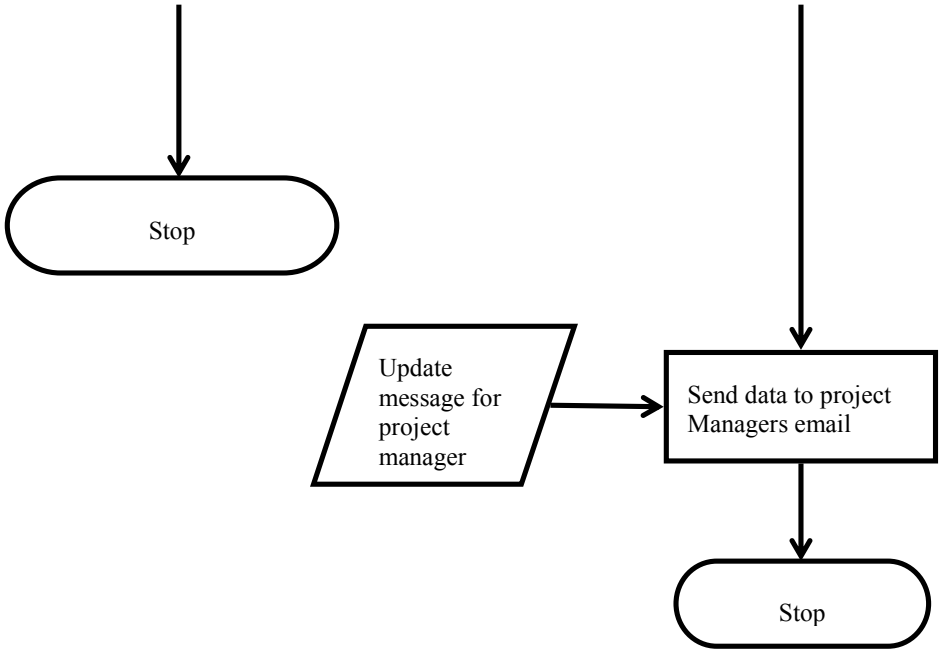


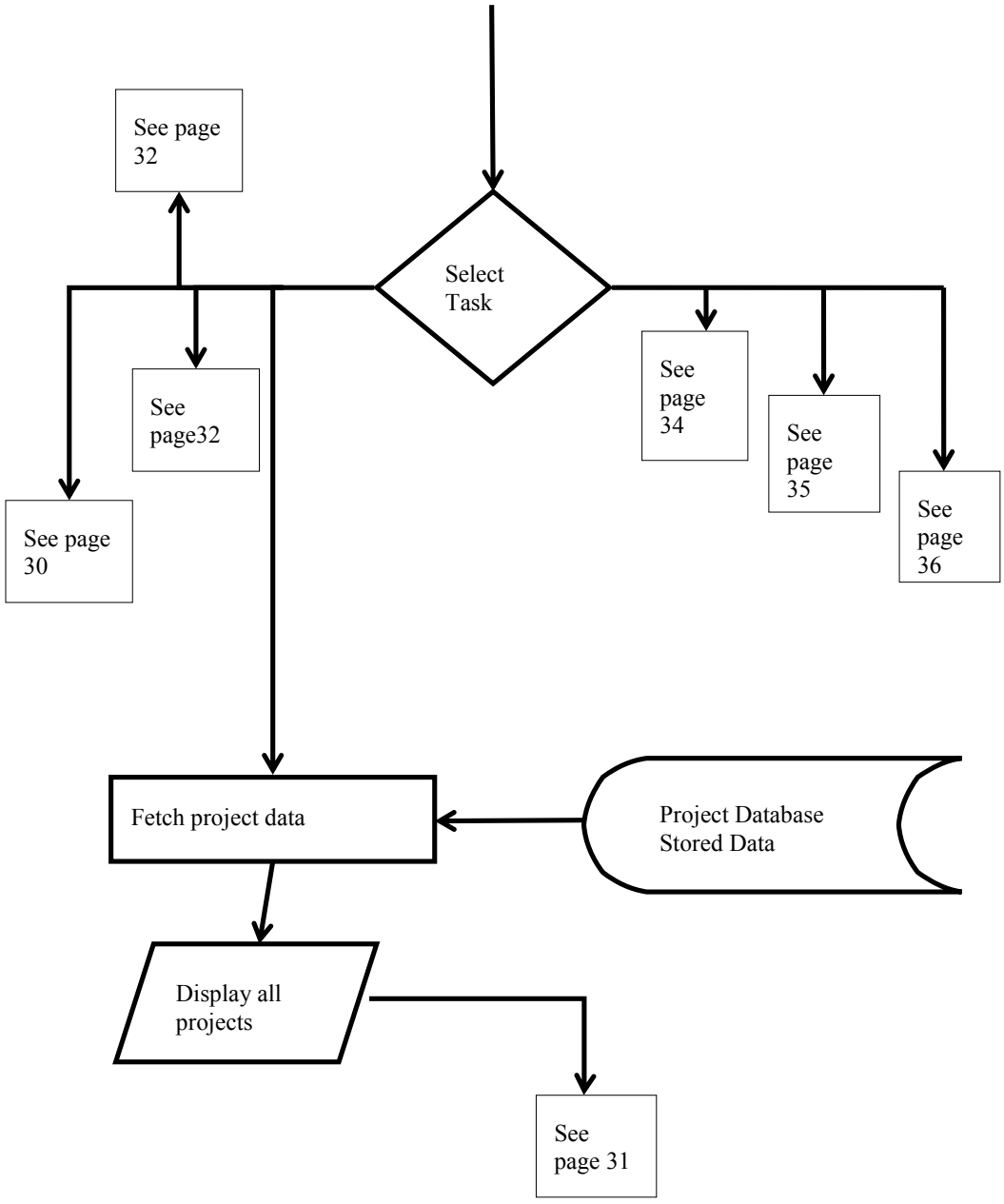


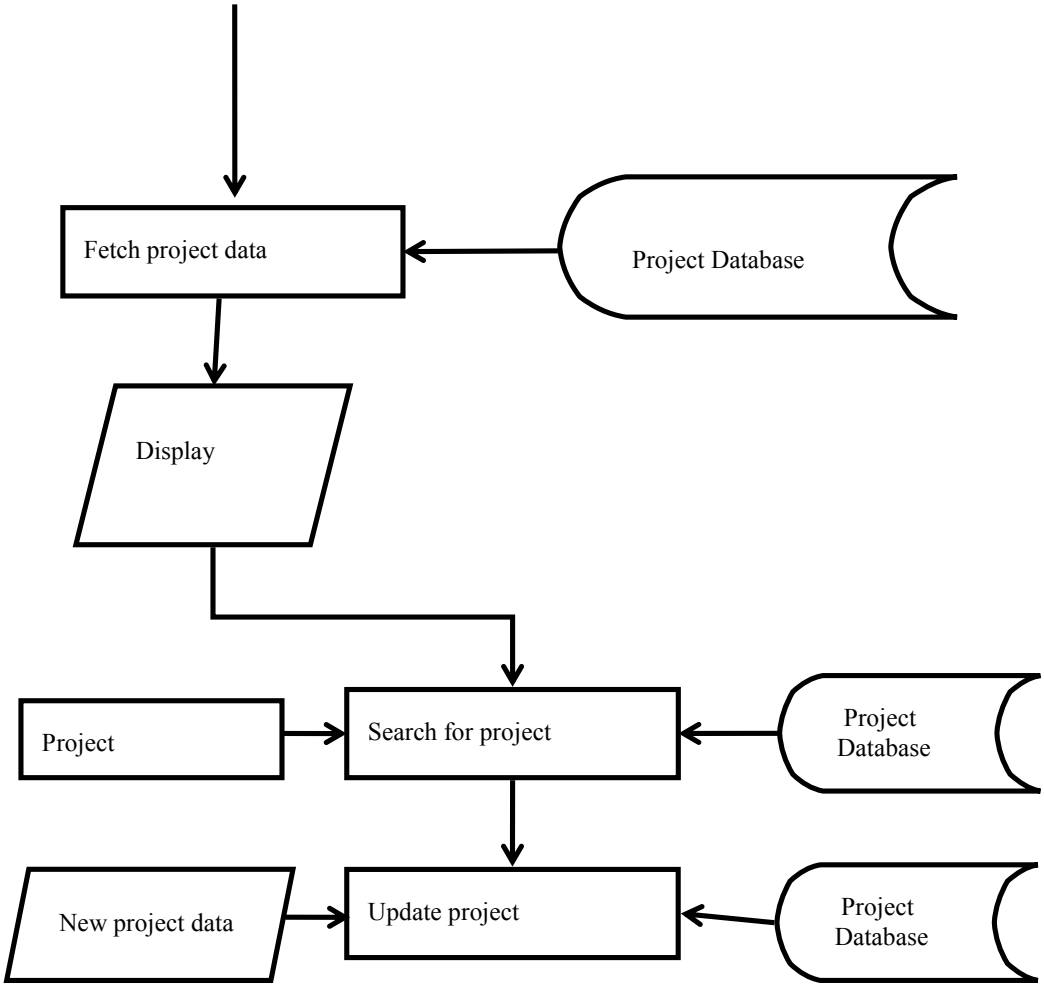

Do the log in details belong to a project manager?
NO
Yes
Display Consultant Interface
Display Manager Interface
Select task
See page 32
See page 32
See page 33
Fetch project data
Project Database
Display all projects
See page 30
See page 31

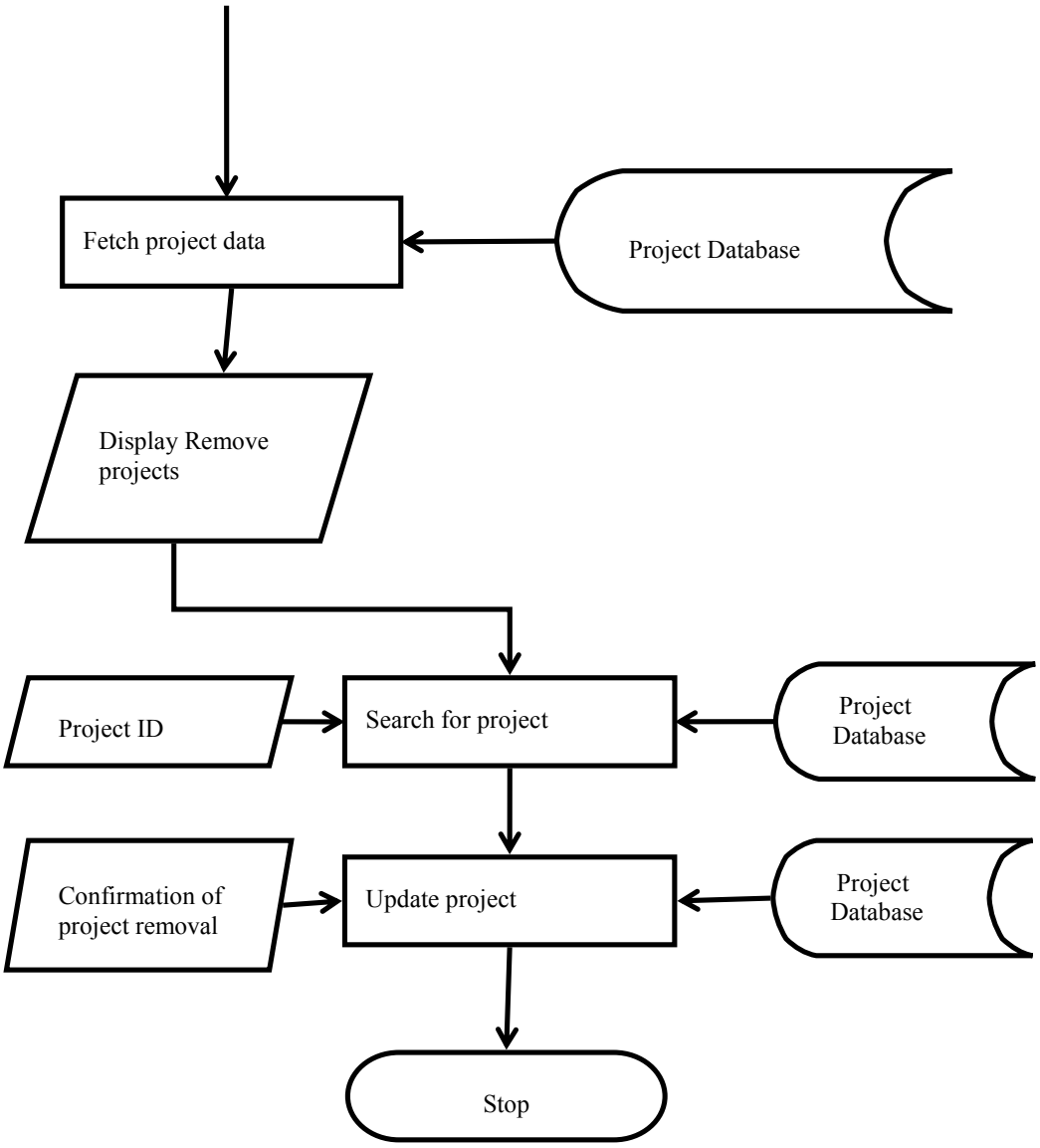

Fetch project data
Project Database
Display Current Projects
Project ID
Search for project
Project Database
NO
Project found
YES
Display Error message Project not found
Display project details
Stop

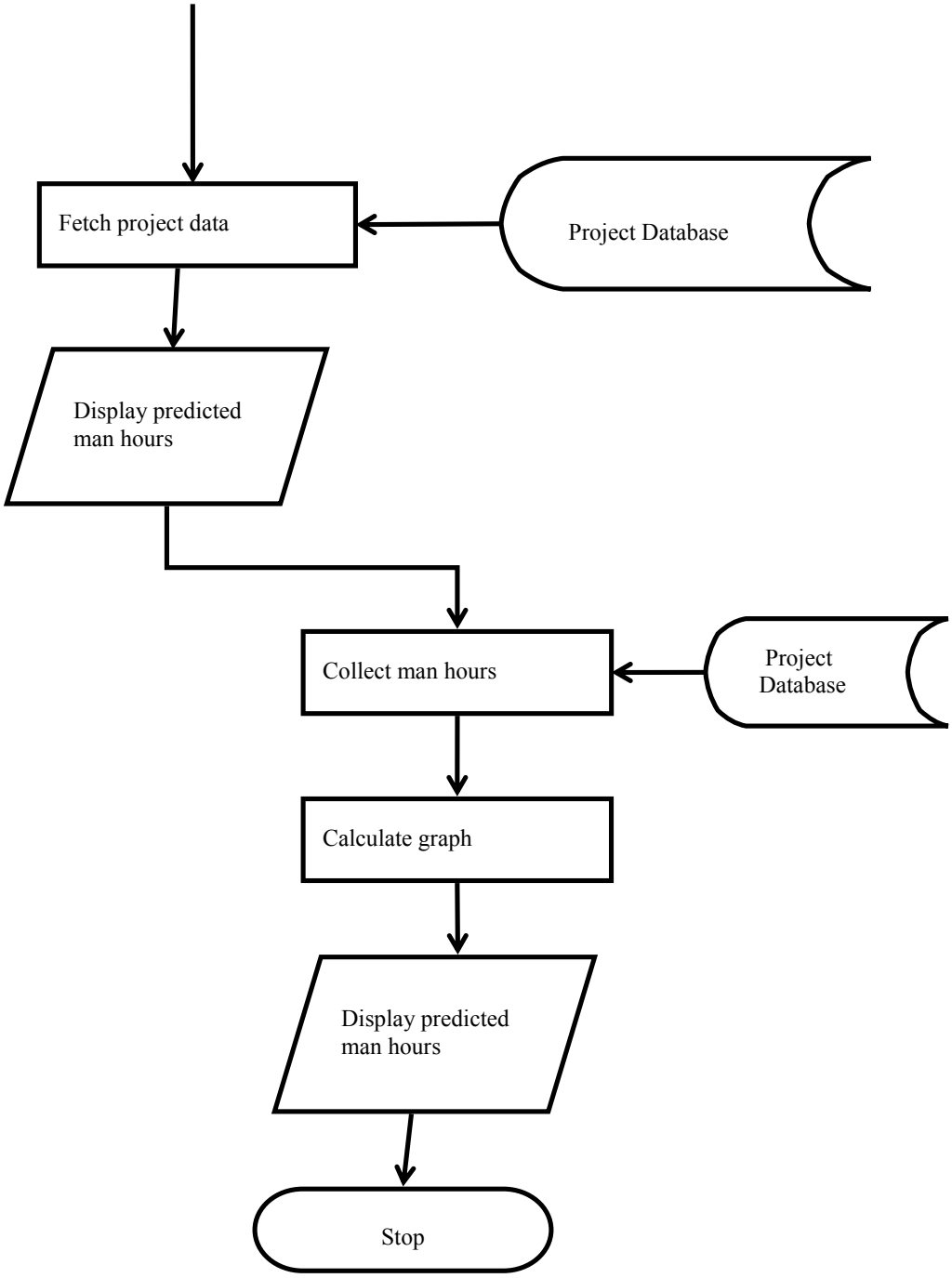

Project ID
Search for project
Project Database
NO
Project found
YES
Display Error message Project not found
Display project details
Stop


Stop
Update message for project manager
Send data to project Managers email
Stop

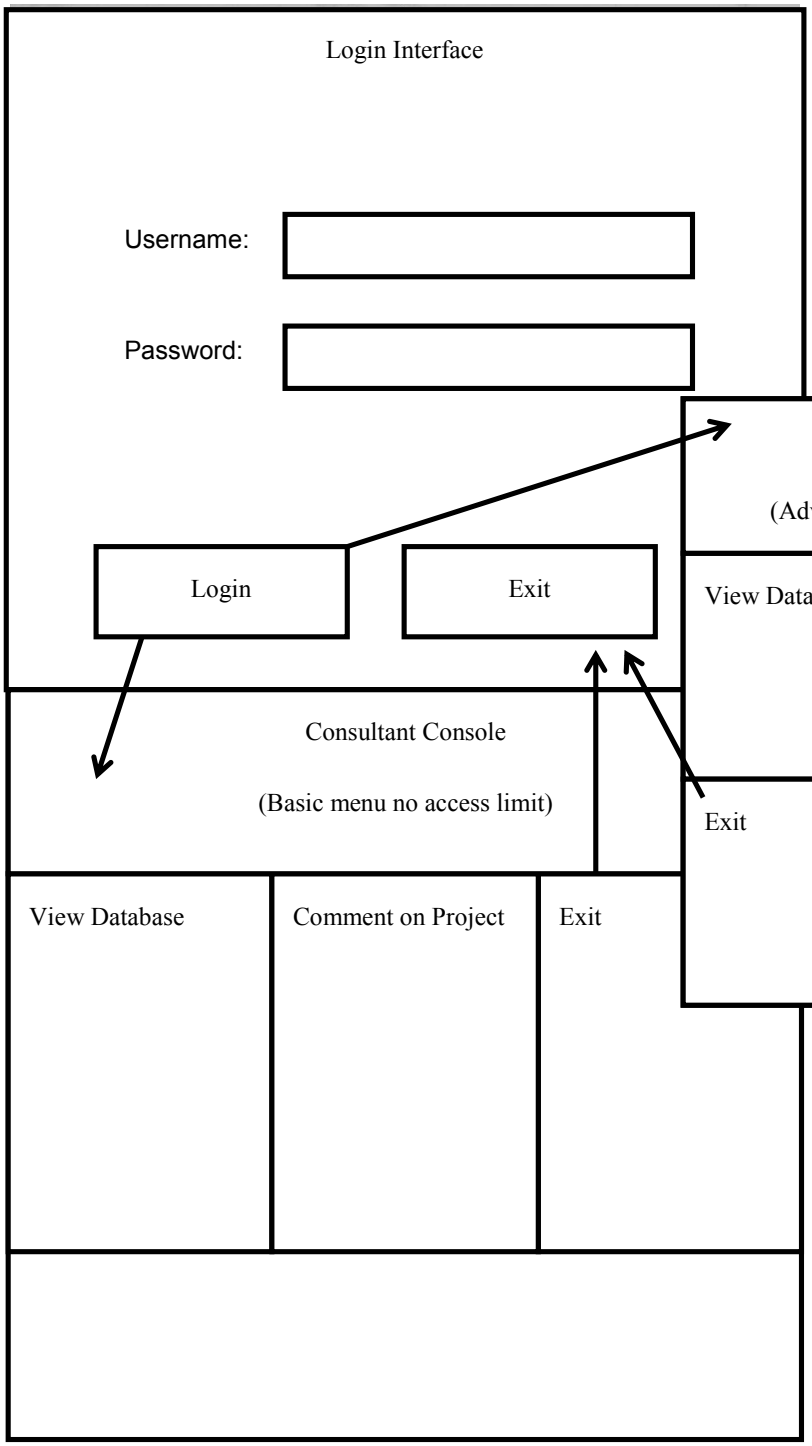

See page 32
Select Task
See page32
See page 34
See page 35
See page 36
See page 30
Fetch project data
Project Database Stored Data
Display all projects
See page 31


Fetch project data
Project Database
Display
Project
Search for project
Project Database
New project data
Update project
Project Database


Fetch project data
Project Database
Display Remove projects
Project ID
Search for project
Project Database
Confirmation of project removal
Update project
Project Database
Stop


Fetch project data
Project Database
Display predicted man hours
Collect man hours
Project Database
Calculate graph
Display predicted man hours
Stop

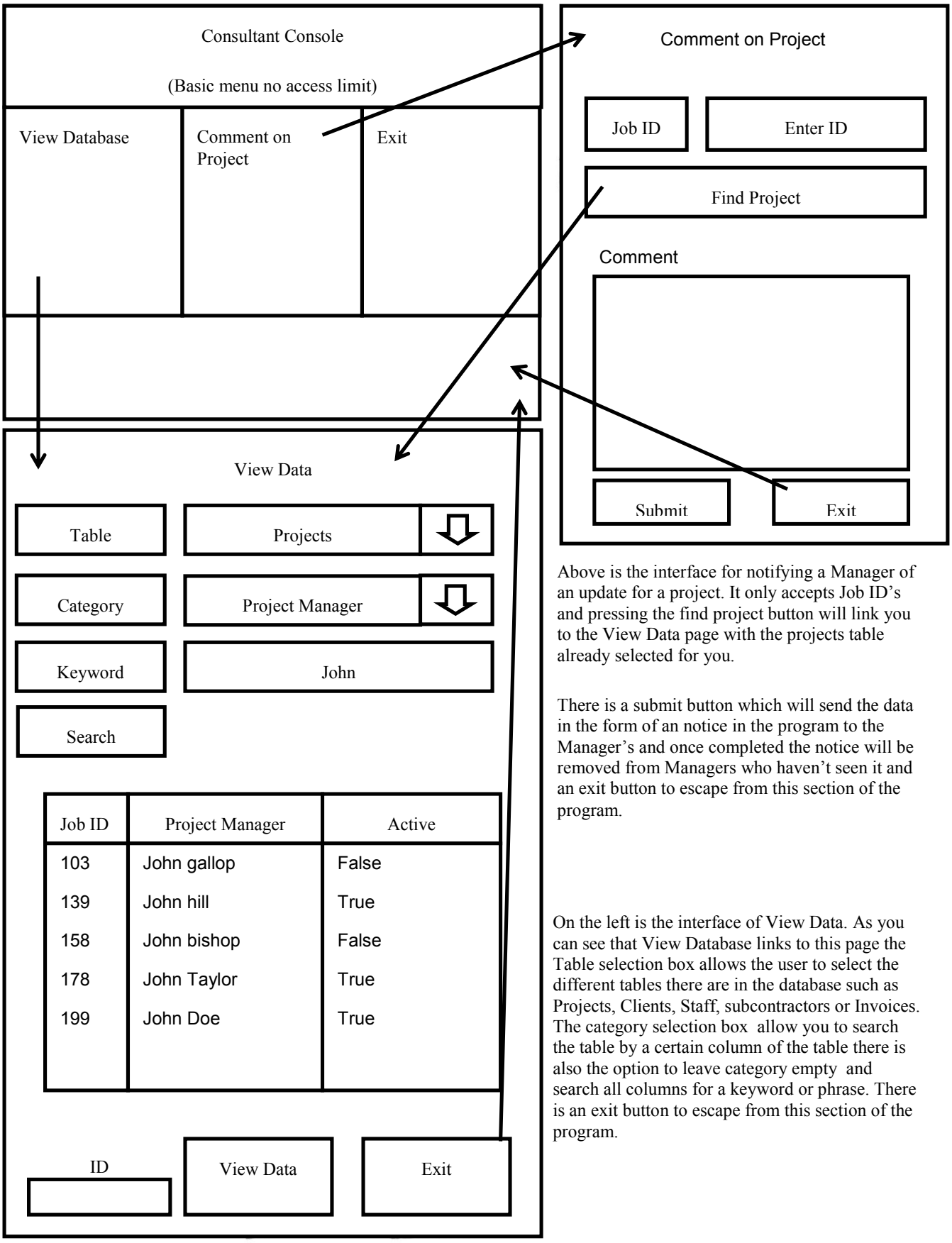
2. User Interface Designs



To the left is the login Page design. There are large buttons in case they are using this on their phone or tablet. It is simplistic so fewer mistakes are made when using it.

Below is the advanced user interface it is only accessed by project managers and above. It has big buttons so it is hard to make a mistake and is easy to use on a tablet or phone

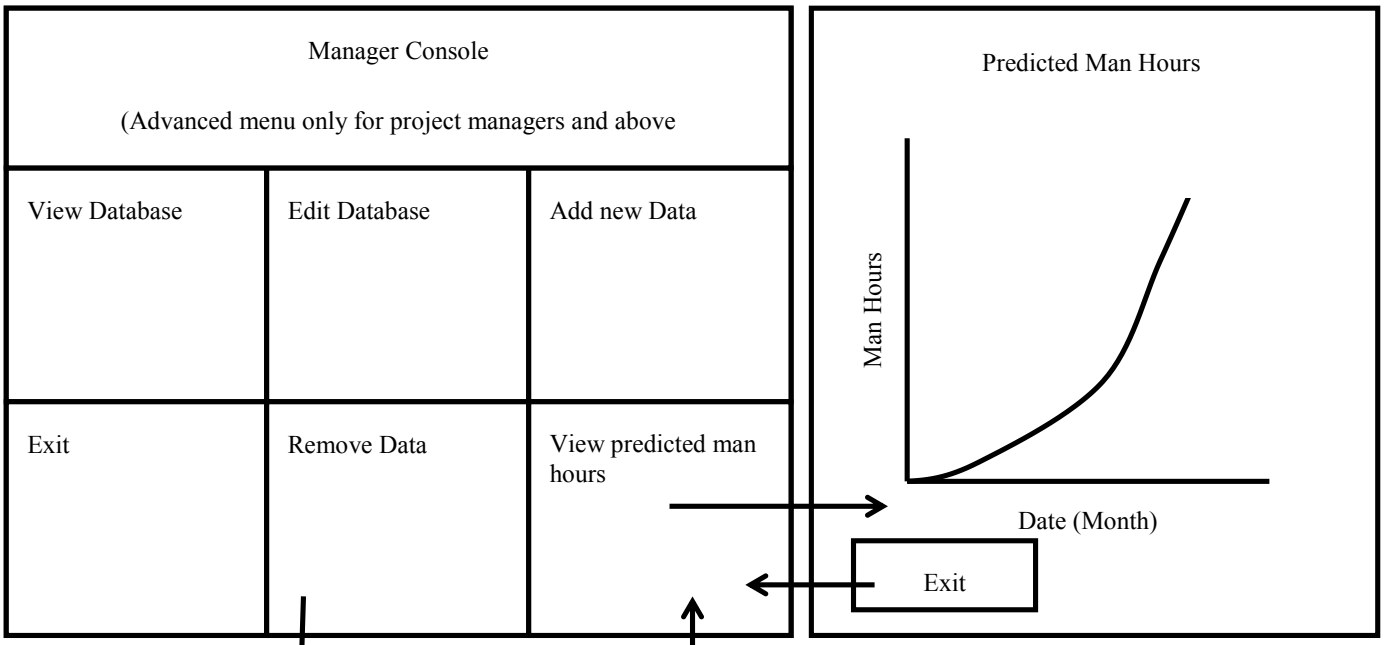
On the left is the simple user interface. It can be accessed by all staff and has 3 large buttons for viewing projects and notifying project managers of updates.

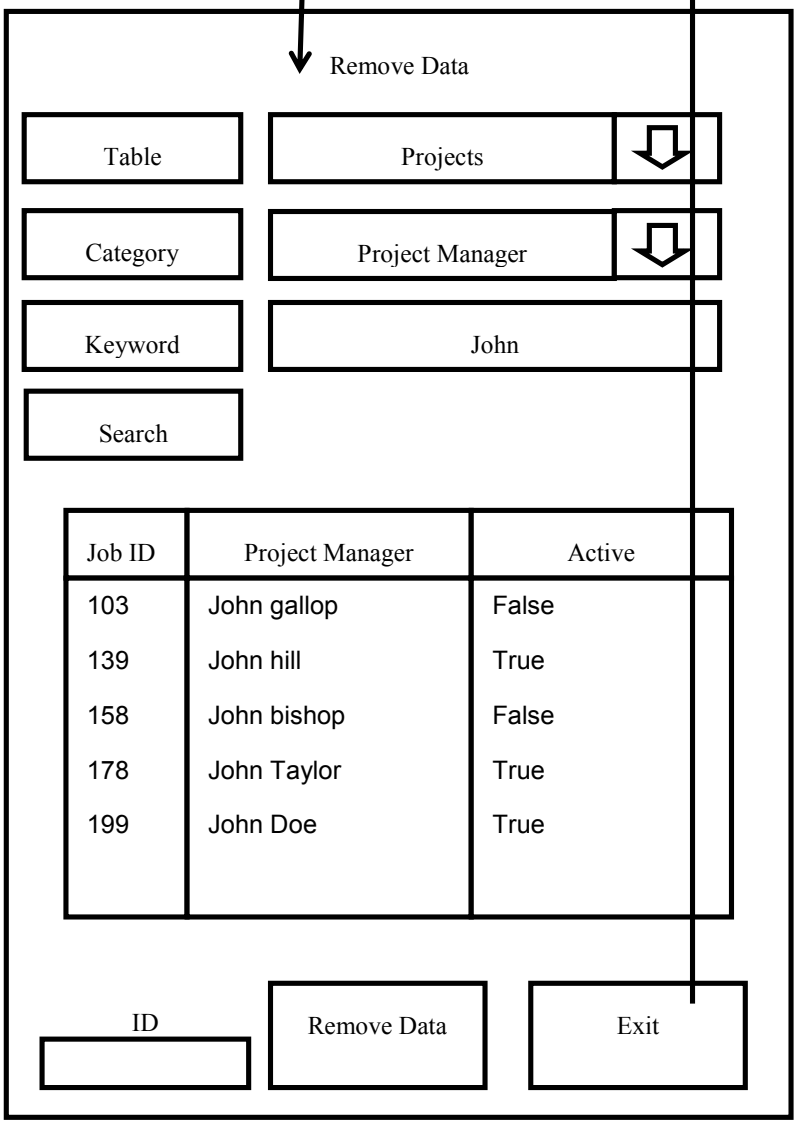


Above is the interface for notifying a Manager of an update for a project. It only accepts Job ID's and pressing the find project button will link you to the View Data page with the projects table already selected for you.

There is a submit button which will send the data in the form of an notice in the program to the Manager's and once completed the notice will be removed from Managers who haven't seen it and an exit button to escape from this section of the program.

On the left is the interface of View Data. As you can see that View Database links to this page the Table selection box allows the user to select the different tables there are in the database such as Projects, Clients, Staff, subcontractors or Invoices. The category selection box allow you to search the table by a certain column of the table there is also the option to leave category empty and search all columns for a keyword or phrase. There is an exit button to escape from this section of the program.

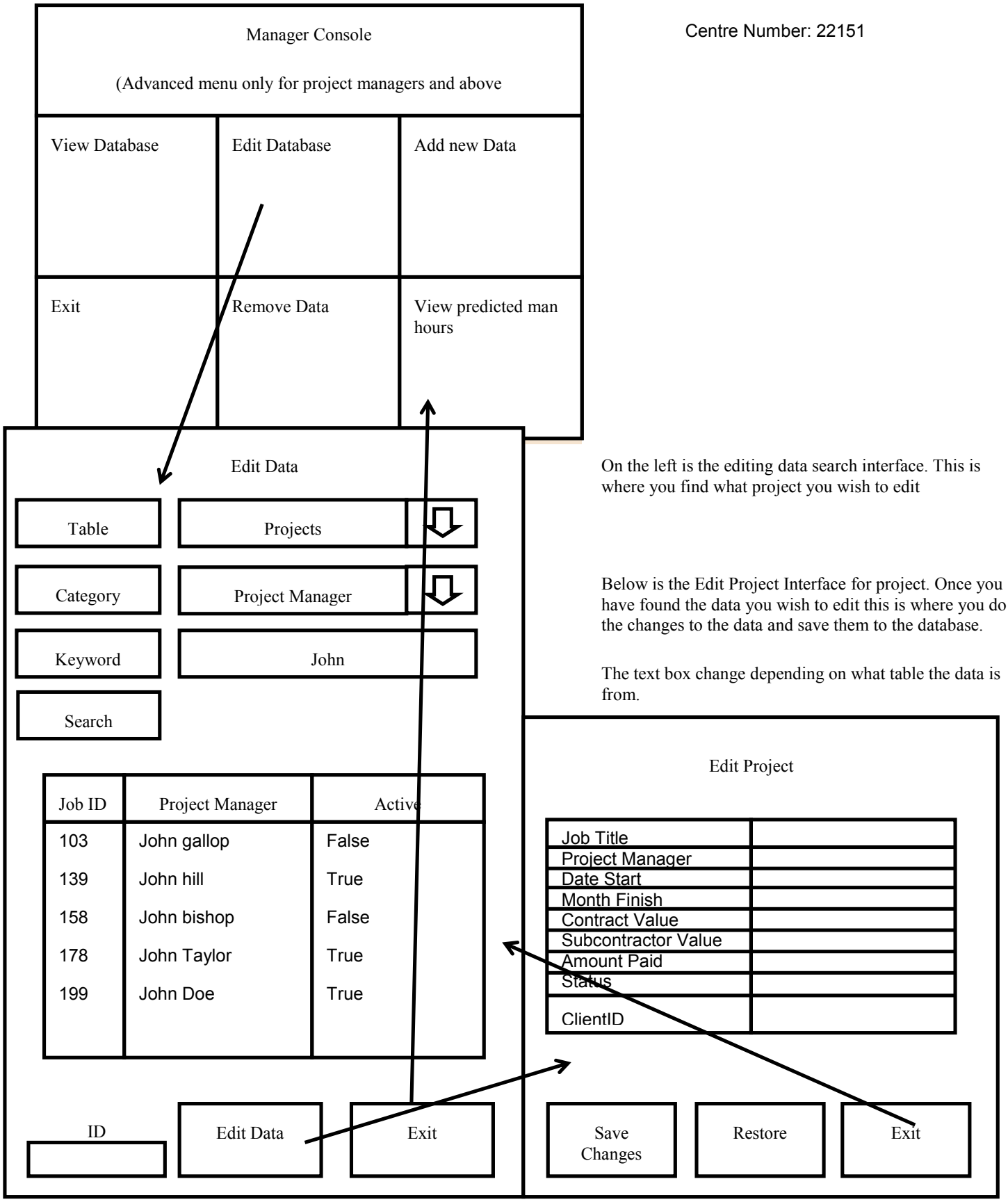




Above is the interface for viewing the work load. It takes the work hours to complete a project and plots them on a graph

The page has an exit button that links to the main menu.

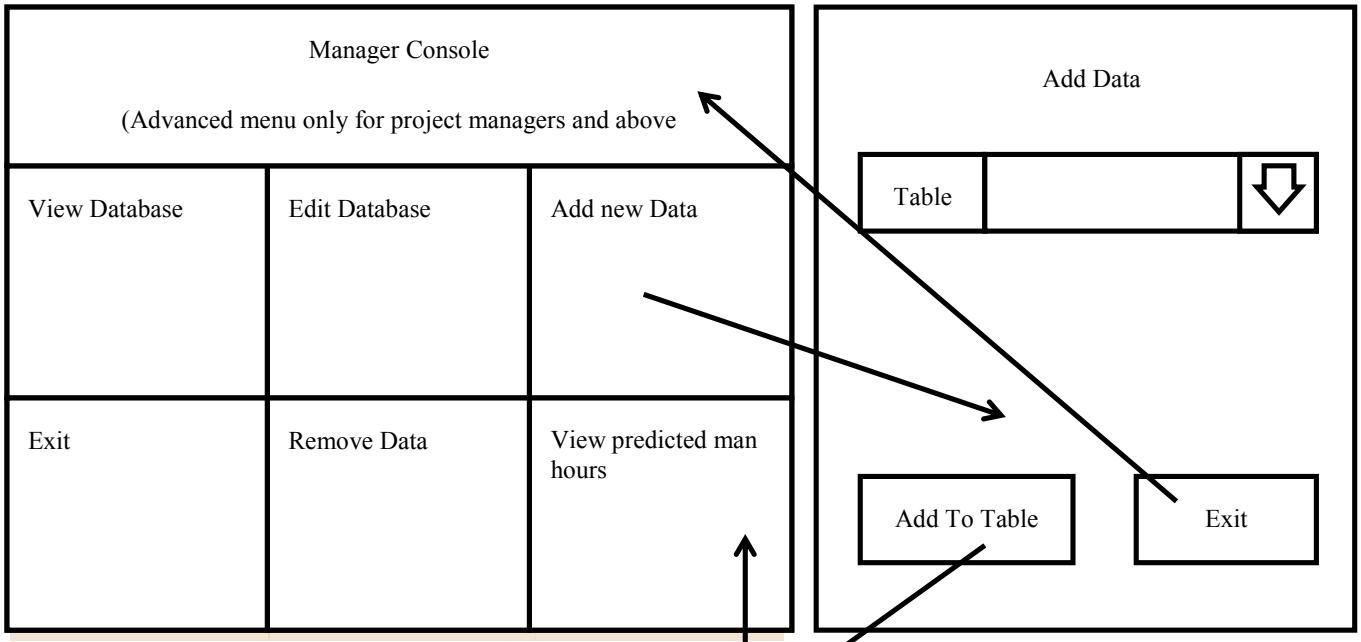
To the left is the interface to remove data. It has a search tool to find the data you wish to delete. You then enter the id of the project you wish to delete into the job id box to remove it. It also has an exit button which takes you back to the main menu.

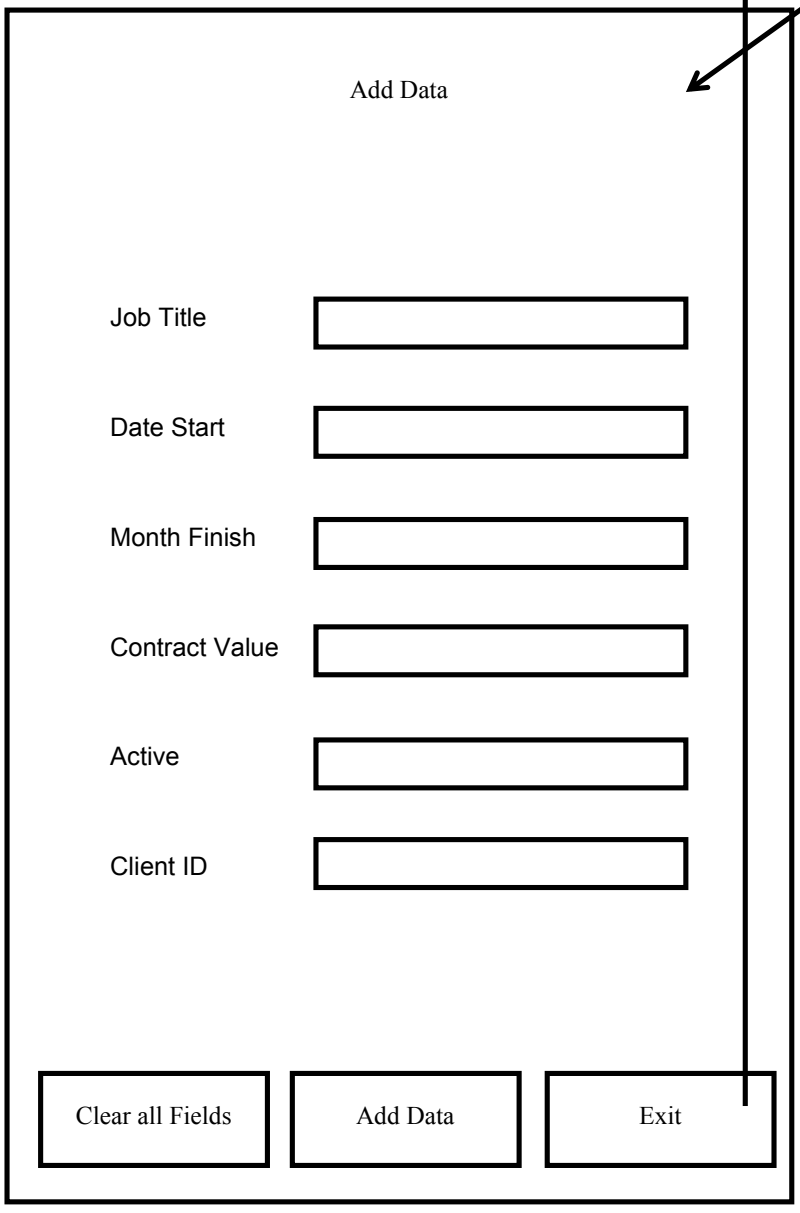


On the left is the editing data search interface. This is where you find what project you wish to edit

Below is the Edit Project Interface for project. Once you have found the data you wish to edit this is where you do the changes to the data and save them to the database.

The text box change depending on what table the data is from.





Above is the interface to select which table you wish to add new data to.

On the left is the Add data Interface.

There are big clear buttons for submission, clear all field, submission, add new project and exit. This is so you don't make a mistake and clear the fields by accident.

3. Hardware Specification

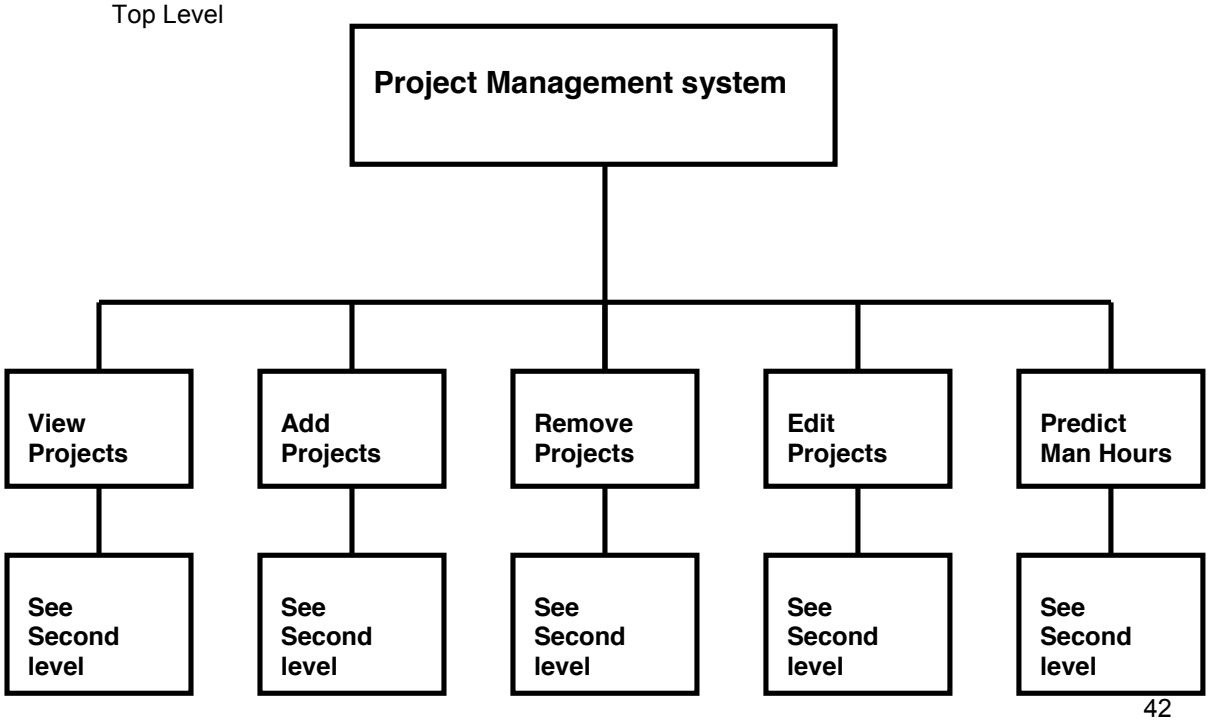
The following is a list of parts required to run the system effectively:

- Keyboard and Mouse to navigate through the system and enter data. My client use these to interact with their current database on a daily basis so I am using them in my system as they are very familiar with them.
- Visual Display Unit to view the contents of the database. The user needs to be able to view the data that is produced from the database e.g. to view past projects or graph predictions. The user would not be able to operate the program effectively without one.
- Approximately 512MB of Main Memory to input data into the database and run other programs effectively, such as the operating system or an Internet browser. Little processing power is needed as the program does not do very many calculations.
- Approximately 3GB of free space on a hard drive to store the program and the data stored in the database, otherwise they would lose the data when they turn off the computer .

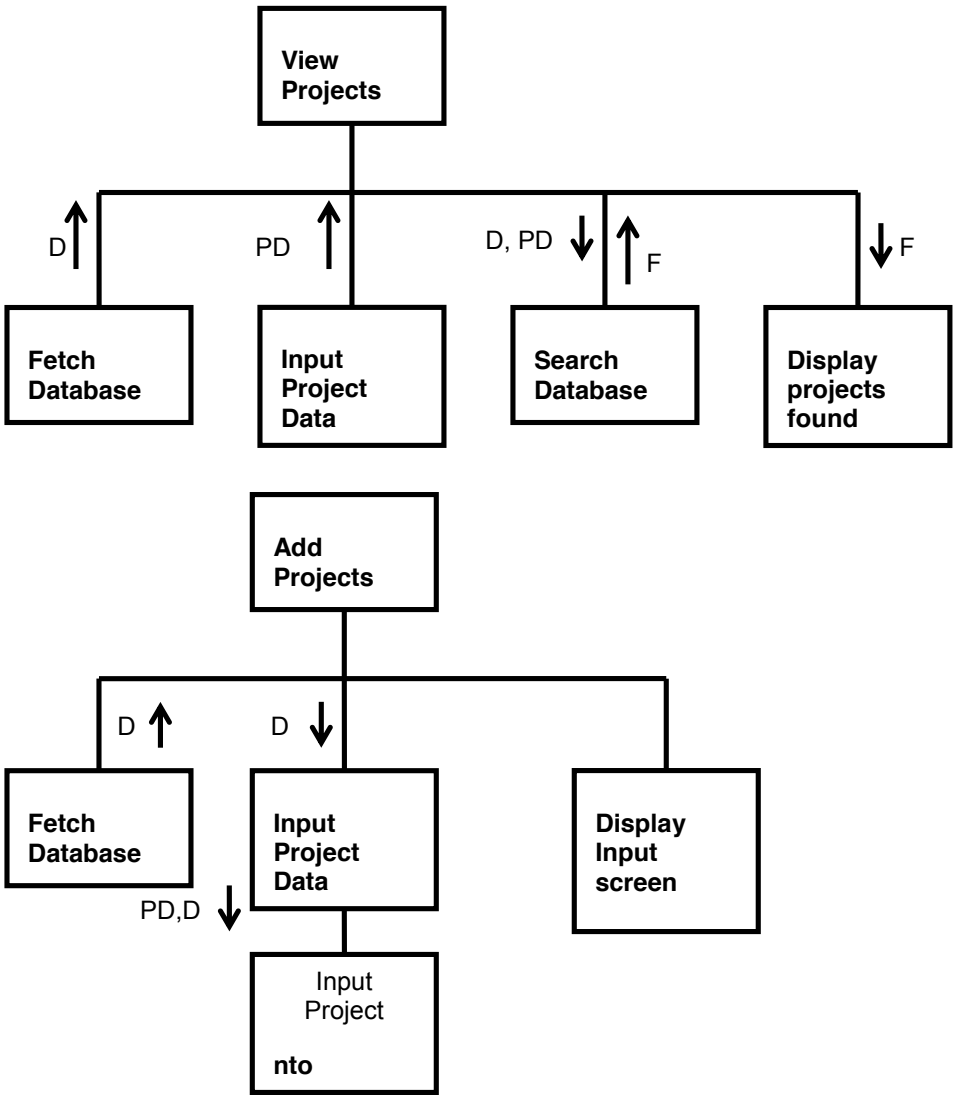
4. Program Structure

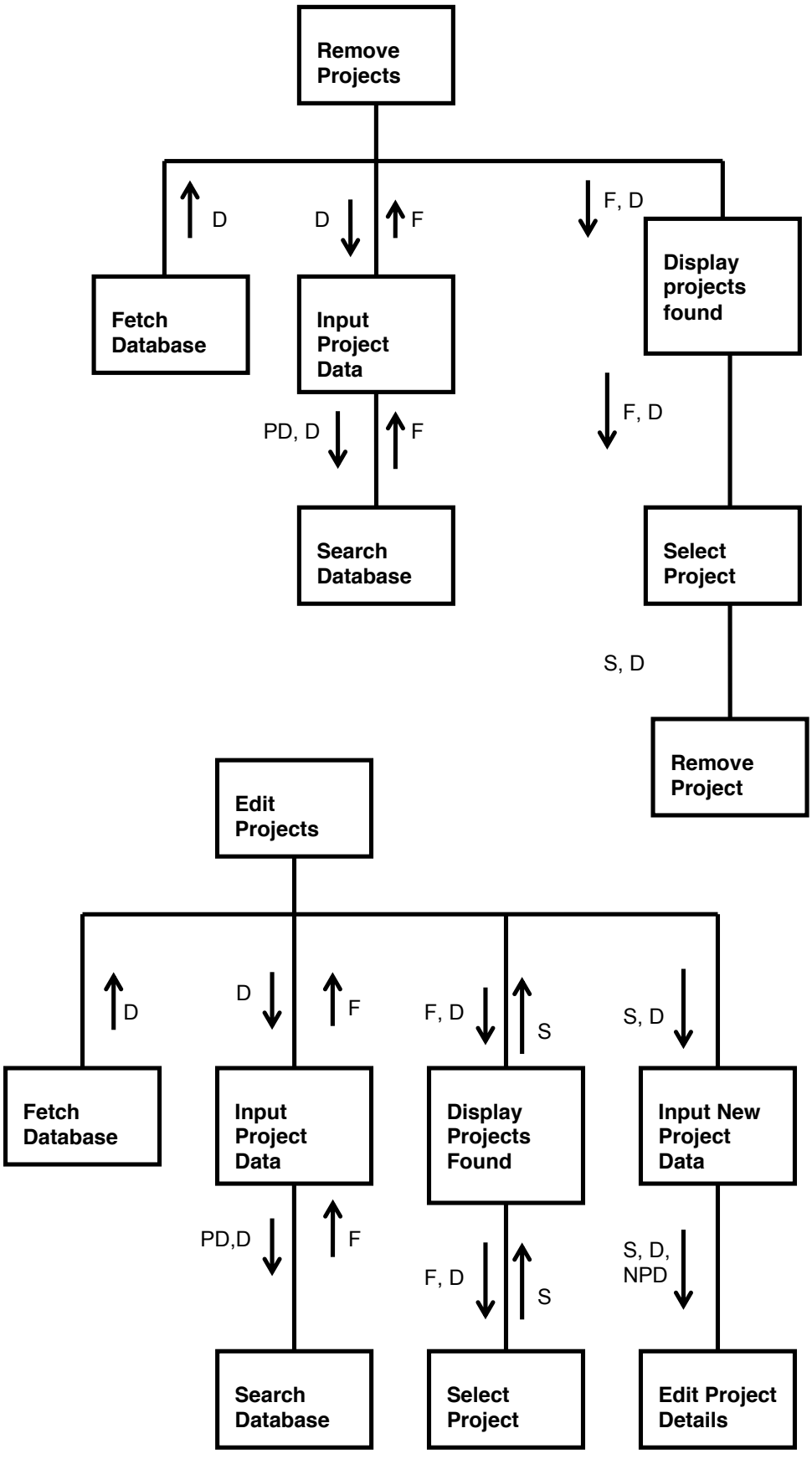
4.1. Topdown Design Structure Charts

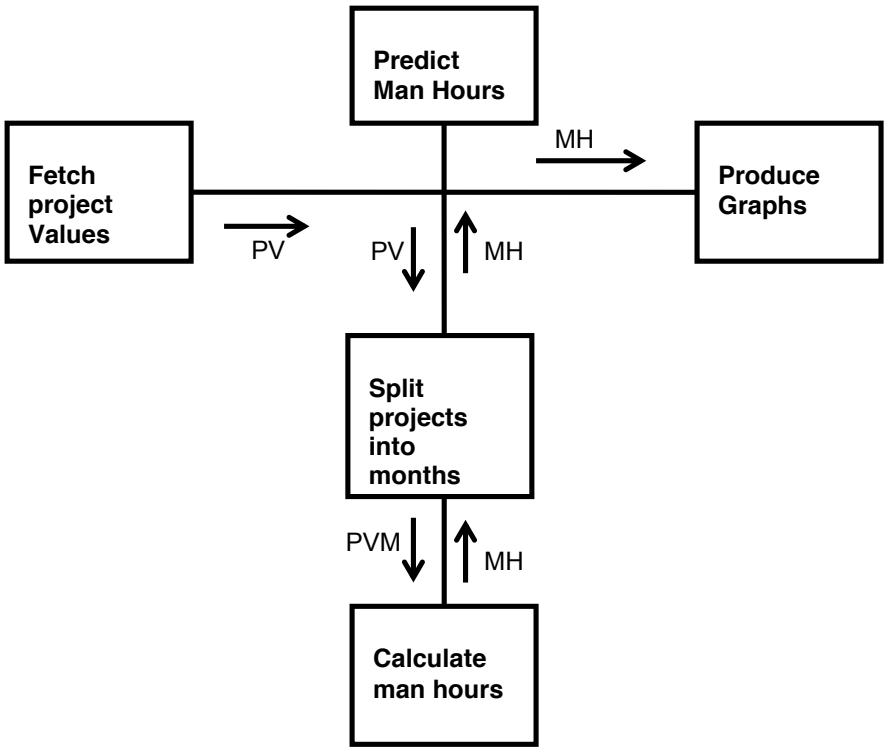
Top Level



Second Level




Remove Projects
D
D
F
F, D
Fetch Database
Input Project Data
Display projects found
PD, D
F
F, D
Search Database
Select Project
S, D
Remove Project
Edit Projects
D
D
F
F, D
S
S, D
Fetch Database
Input Project Data
Display Projects Found
Input New Project Data
PD,D
F
F, D
S
S, D, NPD
Search Database
Select Project
Edit Project Details



4.2. Algorithms In Pseudo-Code For Each Data Transformation Process

Record, constant and variable names (Predefined before functions and main program)	**Pseudo-code**	**comment**
Projects	RECORD Projects is JobTitle: String DateStart: Date DateFinish: Date ContractValue: Integer ClientID: Integer Active: Boolian END RECORD	This record will allow us to interact with the database with a constant set of variables. This allows us to interact with the database knowing that no extra variables will be put into the database and cause an error.

Clients	RECORD Clients is CompanyName: String AddressNumber: Integer AddressStreet: String AddressCity: String AddressCounty: String PostCode: String Email: String Moblie:Integer Office:Integer END RECORD	
Staff	RECORD Staff is FirstName: String SecondName: String Gender: String Title: string DOB: Date Role: String Email: String Mobile: Integer Home: Integer AddressNumber: Integer AddressStreet: String AddressCity: String AddressCounty: String PostCode: String END RECORD	

Subcontractors	RECORD Subcontractors is CompanyName: String AddressNumber: Integer AddressStreet: String AddressCity: String AddressCounty: String PostCode: String Email: String Moblie:Integer Office:Integer END RECORD	
Invoices	RECORD Invoices is InvoiceNo: String InvoiceValue:Integer END RECORD	
Controller	ARRAY Controller: string [5] Controller [1] ← Projects Controller [2] ← Clients Controller [3] ← Staff Controller [4] ← Subcontractors Controller [5] ← Invoices	This keeps all the records in once place thus making it simpler to access and pull out the record that you want.

Functions	Pseudo-code	Description

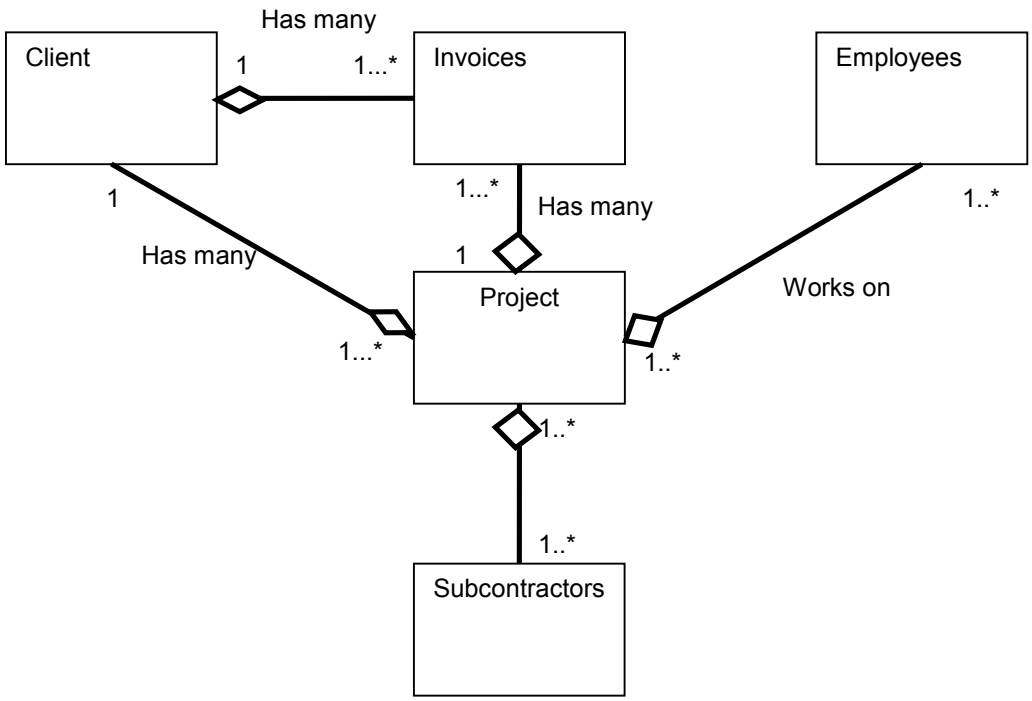
AddData	FUNCTION AddData(Controller:ARRAY)
	choice: String
	Selection: Integer
	Selection ← 0
	WHILE Selection = 0 DO
	Output “please select the table to extend”
	INPUT choice
	IF choice = “Projects” THEN
	Selection ← 1
	ELSE IF choice = “Clients” THEN
	Selection ← 2
	ELSE IF choice = “Staff” THEN
	Selection ← 3
	ELSE IF choice = “Subcontractors” THEN
	Selection ← 4
	ELSE IF choice = “Invoices” THEN
	Selection ← 5
	ELSE IF choice = “Exit” THEN
	Selection ← 6
	ELSE
	OUTPUT “Invalid Choice please select Projects ,Clients ,Staff , Subcontractors ,Invoices or Exit to exit”
	END IF
	END WHILE
	CONNECT to ProjectManagement Database
		Selects a data template from the Controller array
		
		Then uses that template to add new data to the existing data base

```
        IF Selection = 1 THEN
                OUTPUT  “please enter JobTilte, DateStart, DateFinish, ContractValue and Active in the fields below in the order shown”
                INPUT Controller [1].JobTitle
                INPUT Controller [1].DateStart
                INPUT Controller [1].DateFinish
                INPUT Controller [1].ContractValue
                INPUT Controller [1].Active
        END IF


        IF Selection = 2 V 4 THEN
                OUTPUT “Please enter  CompanyName, HouseNumber, street, city, county, postcode, mobile, office in the order given”
                INPUT Controller [2].CompanyName
                INPUT Controller[2].AddressNumber
                INPUT Controller[2].AddressStreet
                INPUT Controller[2].AddressCity
                INPUT Controller[2].AddressCounty
                INPUT Controller[2].Postcode
                INPUT Controller[2].Mobile
                INPUT Controller[2].Office
                IF Selection =2 THEN
                else
                END IF
        END iF

        IF Selection 3 THEN
                OUTPUT “Please enter  First Name, Second Name, Gender, Title, DOB, Role, Email, Mobile, Home,  Number ,Street,  AddressCity, County and PostCode in the order given”
                INPUT  Controller[3].FirstName
                INPUT  Controller[3].SecondName
                INPUT  Controller[3].Gender
                INPUT  Controller[3].Title
                INPUT  Controller[3].DOB
                INPUT  Controller[3].Role
                INPUT  Controller[3].Email
                INPUT  Controller[3].Mobile
                INPUT  Controller[3].Home
                INPUT  Controller[3].AddressNumber
                INPUT  Controller[3].AddressStreet
                INPUT  Controller[3].AddressCity
                INPUT  Controller[3].AddressCount
                INPUT  Controller[3].PostCode
        END IF

        IF Selection 5 THEN
                OUTPUT”please enter the Invoice number and the value of the invoice in the order given”
                INPUT Controller[5].InvoiceNumber
                INPUT Controller[5].InvoiceValue
        END IF
```

Remove Data	FUNCTION RemoveData()
	
	CONNECT to ProjectManagement Database
	choice: String
	Selection: Integer
	ID: integer
	Selection ← 0
	WHILE Selection = 0 DO
	Output “please select the table to extend”
	INPUT choice
	IF choice = “Projects” THEN
	Selection ← 1
	OUTPUT “Please enter the ID of the project you wish to delete”
	INPUT ID
	ELSE IF choice = “Clients” THEN
	Selection ← 2
	OUTPUT “Please enter the ID of the Client you wish to delete”
	INPUT ID
	ELSE IF choice = “Staff” THEN
	Selection ← 3
	OUTPUT “Please enter the ID of the Staff you wish to delete”
	INPUT ID
	ELSE IF choice = “Subcontractors” THEN
	Selection ← 4
	OUTPUT “Please enter the ID of the Subcontractor you wish to delete”
	INPUT ID
	ELSE IF choice = “Invoices” THEN
	Selection ← 5
	OUTPUT “Please enter the ID of the Invoice you wish to delete”
	INPUT ID
	ELSE IF choice = “Exit” THEN
	Selection ← 6
	ELSE
	OUTPUT “Invalid Choice please select Projects ,Clients ,Staff , Subcontractors , Invoices or Exit to exit”
	END IF
	END WHILE	

EditData	FUNCTION EditData(Controller:ARRAY) RemoveData() AddData(Controller)	Uses the remove function to remove the incorrect data and then calls the add function to add in the correct data

4.3. Object diagrams



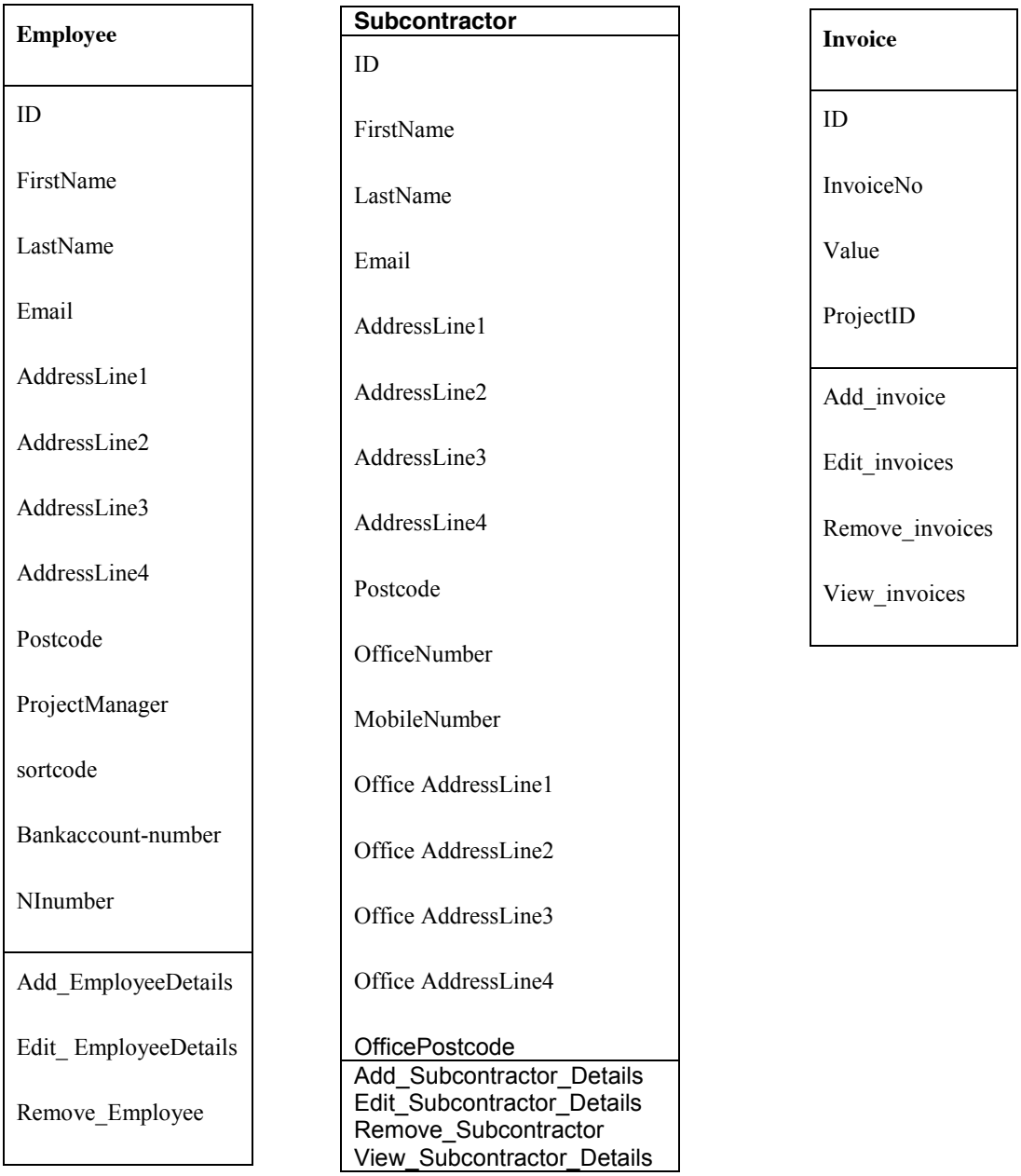
4.4. Class definitions

Key:

Label
Attribute
behaviours

Project
CientID JobNumber JobTitle DateStart MonthFinishing ContractValue NetInvoiceValue SubcontractorsValue AmountPaid Status
Add_ProjectDetails Add_ProjectManager Edit_ProjectDetails Edit_ProjectManager View_ProjectInfo Remove _Project

Client
ID FirstName LastName Email AddressLine1 AddressLine2 AddressLine3 AddressLine4 Postcode OfficeNumber MobileNumber Office AddressLine1 Office AddressLine2 Office AddressLine3 Office AddressLine4 OfficePostcode
Add_PersonalClientInfo Add_OfficeInformation Edit_PersonalClientInfo Edit_OfficeInfo View_allinfo Remove_Client


Employee
ID
FirstName
LastName
Email
AddressLine1
AddressLine2
AddressLine3
AddressLine4
Postcode
ProjectManager
sortcode
Bankaccount-number
NInumber
Add_EmployeeDetails
Edit_ EmployeeDetails
Remove_Employee
Subcontractor
ID
FirstName
LastName
Email
AddressLine1
AddressLine2
AddressLine3
AddressLine4
Postcode
OfficeNumber
MobileNumber
Office AddressLine1
Office AddressLine2
Office AddressLine3
Office AddressLine4
OfficePostcode
Add_Subcontractor_Details
Edit_Subcontractor_Details
Remove_Subcontractor
View_Subcontractor_Details
Invoice
ID
InvoiceNo
Value
ProjectID
Add_invoice
Edit_invoices
Remove_invoices
View_invoices

5. Prototyping

5.1. Consideration Of Impact On Design And Development

An area that I am going to prototype is the entering, editing and deletion of data from the database. If I didn't do a prototype for this and it failed then my system would be useless to Element Energy.

I am also prototyping the graphs in python as this is one of the main features of the program. This will be a high process based operation. I am going to prototype it to make sure that the clients system can handle the work load and if so how much it affects other programs running at the same time.

Also, it would be useful to make a prototype of the user interface and have the client use it for a while on different devices and screen resolutions to make sure that they can navigate through the different pages with ease. This will also give them the chance to give feedback on the project, and will also allow them to give extra information and clarity on what they want the system to do. This will also ensure that I have not missed anything that they wish the program to do.

6. Definition of Data Requirements

6.1. Identification Of All Data Input Items

- Work hours
- Job title
- Project Manager
- Date Start
- Month Finishing
- Contract Value
- Subcontractors value
- Amount Paid
- Status
- Job ID
- Invoice numbers
- Username
- Password
- Client Name
- Company Name

- Address
- Office Number
- Mobile
- Staff ID
- Role
- Prefix
- Gender
- Date Of Birth
- SubcontractorID

6.2. Identification of all output items

- Work hours
- Job title
- Project Manager
- Date Start
- Month Finishing
- Contract Value
- Subcontractors value
- Amount Paid
- Status
- Job ID
- Invoice numbers
- Username
- Password
- Client Name
- Company Name

- Address
- Office Number
- Mobile
- Staff ID
- Role
- Prefix
- Gender
- Date Of Birth
- SubcontractorID

6.3. Explanation of how data output items are generated

Output	How it is generated
Project data	Input by user
Client data	Input by user
Invoice data	Input by user
Staff data	Input by user
Subcontractor data	Input by user
Graphs	Project values pulled from Project data and spilt into months and then the man hours for each month is calculated and plotted on a graph with man hours against months.

6.4. Data Dictionary

Name	Data Type	Length	Validation	Example Data	Comment
Job Number	Integer	8 bytes	There is no value of same value	1000	This is produced by the system so there is a very low chance that there could be a problem.
Project manager	string	26 characters	Must have a input and be a existing manager	John Doe	I'm using a Selection box to prevent existing Managers not being chosen

Job Title	String	40 characters	Must give two inputs that match each other	Yorkshire Turbines	This is prone to errors as there is no way to check if this is the real name of the project
Date Start	Date	DD/MM/YYYY	It need to be real day of the year and also has to be entered	10/11/2012	There may be complications with the client and this date may change so the start date needs to be adjustable.
Date Finish	Date	DD/MM/YYYY	It is not before Date Start	15/5/13	This is hard to evaluate as some project workers could fall ill during the time of the project. Then the project would take longer therefore have a later dead line so this needs to be adjustable.
Active	Boolean	TRUE/FALSE	Has to be TRUE or FALSE	True	This is important as it keeps track of what jobs are finished and about to begin.
Value of Project	Float	4 bytes	Must have a Float value	10000	This is the total value of the project to the client
Invoice Number	Integer	8 bytes	Must be a unique integer value	200	This is the number that is assigned to all invoices so they can be referred too

Invoice Value	Float	4bytes	Must be a Float value	100	This is the value of a single invoice
Subcontractor value	Float	4bytes	Must be a Float value	1000	This is the value of the subcontracts that the company has hired
JobID	Integer	8bytes	Must be a unique integer value	234	This identifies all the different jobs in the system
Amount paid	Float	4bytes	Must be a Float value	11000	This is the amount that the client has given the company as of this moment for a certain project
Username	String	16 characters	Must be over 8 characters and not already exist	John_gumble	This is the unique name for an employee so they can use the system
Password	String	32 characters	Must be over 8 characters	Security5	This is the password to allow the user to log in
ClientID	Integer	8 bytes	Must be a unique integer value	234	This identifies all the different Clients in the system
Company Name	String	32 characters	Must exist	John Lewis	Stores the name of the company
House Number/Name	string	32 characters	Must exist	32	Stores the house number of where the client or employee lives

street	string	32 characters	Must exist	Coles road	Stores the street of where the client or employee lives
City	string	32 characters	Must exist	Cambridge	Stores the city of where the client or employee lives
county	string	32 characters	Must exist	Cambridgeshire	Stores the county of where the client or employee lives
postcode	string	7 characters	Must exist	CB246BS	Stores the postcode of where the client or employee lives
Email	string	256 characters	Must exist	gary@jtresponse.co.uk	Stores the email of the client or the employee
mobile	integer	4 bytes	Must exist	07123123123	Stores a mobile number of a client or a employee
office	string	32 characters	Must exist	+33 (1) 123123123	Stores a office phone number of a client
Staff ID	Integer	8 bytes	Must be a unique integer value	234	It is a unique number for each member of staff
First name	string	32 characters	none	Fred	This stores the first name of a staff member
Second name	string	32 characters	none	Hill	This stores the second name of a staff member

role	string	32 characters	Must exist	consultant	This stores the role that a staff member plays in the company
Title	String	4 characters	Must exist	miss	This stores the prefix of a member of staff
DOB	date	8 characters	Must exist	10/11/12	Stores the date of birth of a staff member
gender	string	6 characters	Must be male or female	female	Stores the staffs gender
Subcontractor ID	Integer	8bytes	Must be a unique integer value	234	It is a unique number for each subcontractor
Invoice ID	Integer	8bytes	Must be a unique integer value	234	It is a unique number for each invoice
Invoice No	string	32 characters	Must be unique	32a	This is a number used for each invoice based on the job number

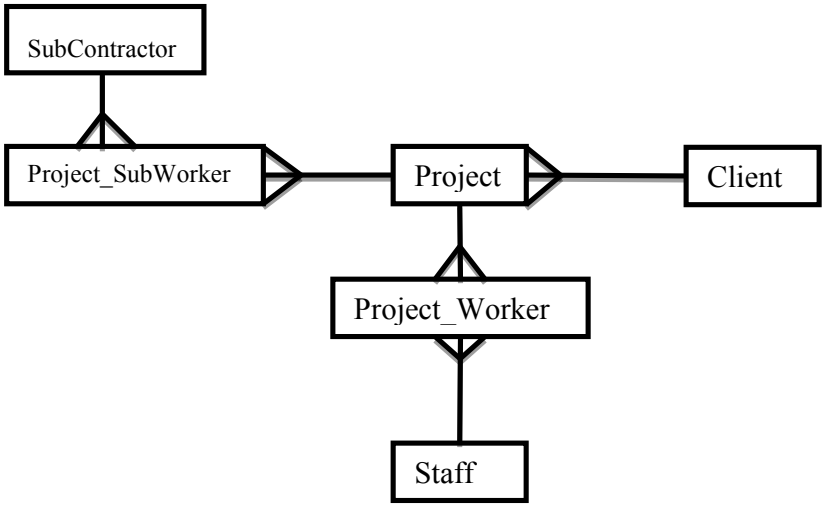
6.5. Identification of appropriate storage media

An appropriate storage media would be a hard drive. The database will need to be kept in long term storage as the data will be needed to be accessed more than once. To back up the system an external hard drive or a flash memory stick will the best as you will be able to fit all the system files on this device and it is portable so you can keep it away from the system kept at the office for extra security

7. Database Design

7.1. Normalisation

7.1.1. ER Diagrams



Entity descriptions-

SubContractor (ID, Company Name, Address Number, Address Street, Address City, Address County, Postcode, Email, Mobile, Office Number)

Staff(ID, First Name, Second Name, Address Number, Address Street, Address City, Address County, Postcode, Email , Mobile, Home, Role, DOB , Title , Gender)

Client (ID, Company Name, Address Number, Address Street, Address City, Address County, Postcode, Email, Mobile, Office Number)

Project_worker (Project worker ID *JobID* , *Staff ID,)*

Project(ID, Job title, Date Start , Date Finish, Product Description, Invoice Numbers, ,Active, Value Of Project)

Project_SubWorker (SubWorkerID ,*JobID,SubcontractorID,* Subcontractor value)

7.1.2. UNF to 3NF

Key-	**Primary Key=*** **Foreign Key=$** **Composite Key =&**	
UNf	**1NF**	**2NF (Attributes Dependant on Job ID)**
JobID*	**Repeating**	Job ID*$
JobTitle	JobID&$	Active
Date Start	Client ID&	Subcontractor ID
Date Finish	Client Company Name	Subcontractor Company Name
Active	Client Address Number	Subcontractor Address Number
Value Of Project	Client Address Street	Subcontractor Address Street
Product Description	Client Address City	Subcontractor Address City
Client ID	Client Address County	Subcontractor Address County
Invoice Numbers	Client Postcode	Subcontractor Postcode
Client Company Name	Client Email	Subcontractor Email
Client Address Number	Client Mobile	Subcontractor Mobile
Client Address Street	Client Office Number	Subcontractor Office Number
Client Address City	Subcontractor ID	Staff ID
Client Address County	Subcontractor Company Name	Staff First Name
Client Postcode	Subcontractor Address Number	Staff Second Name
Client Email	Subcontractor Address Street	Staff Email
Client Mobile	Subcontractor Address City	Staff Mobile
Client Office Number	Subcontractor Address County	Staff home
Subcontractor ID	Subcontractor Postcode	Staff Address Number
Subcontractor Company Name	Subcontractor Mobile	Staff Address Street
Subcontractor Address Number	Subcontractor Office Number	Staff Address City
Subcontractor Address Street	Subcontractor Mobile	Staff Address County
Subcontractor Address City	Subcontractor Email	Staff Postcode
Subcontractor Address County	Staff ID	Staff Job Title
Subcontractor Postcode	Staff First Name	Staff DOB
Subcontractor Email	Staff Second Name	Staff Role
Subcontractor Mobile	Staff Email	Staff Gender
Subcontractor Office Number	Staff Mobile	
Subcontractor Value	Staff home	
Staff ID	Staff Address Number	
Staff First Name	Staff Address Street	
Staff Second Name	Staff Address City	
Staff Email	Staff Address County	
Staff Mobile	Staff Postcode	
Staff home	Staff Job Title	
Staff Address Number	Staff DOB	
Staff Address Street	Staff Role	
Staff Address City	Staff Gender	
Staff Address County		
Staff Postcode	**Non-repeating**	
Staff Job Title	JobID*	
Staff DOB	JobTitle	
Staff Role	Date Start	
Staff Gender	Date Finish	
	Product Description	
	Value Of Project	
	Invoice Numbers	
	Subcontractor Value	
	Active	

Key-	Primary Key=* Foreign Key=$ Composite Key =&	
2NF (attributes depending on Client ID)	**3NF (Group Staff)**	**3NF(group Client)**
Client ID*	Staff ID*	Client ID*
Client Company Name	Staff First Name	Client Company Name
Client Address Number	Staff Second Name	Client Address Number
Client Address Street	Staff Email	Client Address Street
Client Address City	Staff Mobile	Client Address City
Client Address County	Staff home	Client Address County
Client Postcode	Staff Address Number	Client Postcode
Client Email	Staff Address Street	Client Email
Client Mobile	Staff Address City	Client Mobile
Client Office Number	Staff Address County	Client Office Number
	Staff Postcode	
	Staff Job Title	
	Staff DOB	
	Staff Role	
	Staff Gender	

Key- Primary Key=* Foreign Key=$ Composite Key =&		
3NF(Group subcontractor)	**3NF(group Project_worker)**	**3NF(group Project)**
Subcontractor ID*	Project worker ID*	JobID*
Subcontractor Company Name	Job ID$	JobTitle
Subcontractor Address Number	Staff ID $	Date Start
Subcontractor Address Street		Date Finish
Subcontractor Address City		Product Description
Subcontractor Address County		Client ID$
Subcontractor Postcode		Invoice Numbers
Subcontractor Email		Active
Subcontractor Mobile		Value Of Project
Subcontractor Office Number		

3NF(Project SubWorker)
SubWorkerID*
Job ID$
Subcontractor ID $
Subcontractor value

7.2. SQL Queries

- **Add New Project**

‘’’INSERT INTO Projects(JobTitle, DateStart, MonthFinishing, ContractValue, Active, ClientID)VALUES ("{0}","{1}","{2}","{3}","{4}","{5}")'".format (details[0],details[1],details[2],details[3],details[4],Client)

- **Add New Client**

‘’’INSERT INTO Client (CompanyName, AddressNumber, AddressStreet, AddressCity, AddressCounty, PostCode, Email, Mobile, Office) VALUES ("{0}","{1}","{2}","{3}","{4}","{5}","{6}","{7}","{8}")'".format (details[0],details[1],details[2],details[3],details[4],details[5],details[6],details[7],details[8])

- **Add New Staff**

'"INSERT INTO Staff (FirstName, SecondName, Email, Mobile, Home, Role, Title, DOB, Gender, AddressNumber, AddressStreet, AddressCity, AddressCounty, PostCode) VALUES ("{0}","{1}","{2}","{3}","{4}","{5}","{6}","{7}","{8}","{9}","{10}","{11}","{12}","{13}") '".format (details[0],details[1],details[2],details[3],details[5],details[4],details[5],details[6],details[7],details[8],details[9],details[10],details[11],details[12],details[13])

- **Add New Subcontractor**

'"INSERT INTO SubContractors (CompanyName, AddressNumber, AddressStreet, AddressCity, AddressCounty, PostCode, Email, Mobile, Office)VALUES ("{0}","{1}","{2}","{3}","{4}","{5}","{6}","{7}","{8}")'".format (details[0],details[1],details[2],details[3],details[5],details[4],details[5],details[6],details[7],details[8])

- **Add New Invoice**

'"INSERT INTO Invoices(InvoiceNo,InvoiceValue) VALUES ("{0}","{1}")'".format(details[0],details[1])

- **Search Database**

```
ID+="%"
SQL='" Select * from "{0}" where "{1}" like "{2}" '".format(Location,Atrribute,ID)
```

- **Remove Data**

  ''' Delete * from "{0}" where "{1}" = {2}'''.format(Location,Atrribute,ID)

- **Upadate Data**

  '''UPDATE "{0}" SET "{1}" = "{2}" WHERE "{3}" = "{4}"""".format
  (info[0],info[1],info[2],info[3],info[4])

8. Security and Integrity of the System and Data

8.1. Security and integrity of data

The data stored in the database reflects how the company is doing and what is happening to their business. It needs to be password protected to keep this information safe within the company. There will be access restrictions and encrypted backups to prevent the loss of any important data and to keep the data secure

There will be validation on the information that they enter into a project to make sure that it is all correct and that the projects that they are adding to the database do not conflict with each other.

The main menu is split into two sections, the managers menu and the consultant menu. To access these menus' in a secure manner I am going to add a login page to separate the consultants from the project managers. To do this I will use usernames and have project managers linked to the manager menu and the consultants linked to the consultant menu. The reason for this is that the managers menu has the ability to remove and add data into the database and if used improperly and by too many people at once could end up with incorrect data in the database. To prevent data being incorrect I have also added a function to the manager's interface so that only one person can edit the database at one time. .

Upon entering data into the system I have given the database referential integrity which allows the user to update project, clients and any other tables in the database and not worry about updating other tables. If one project is removed then all data pointing to that table is also removed. If you update a project then all data pointing to that project would also update. This means that there will be less errors in the system and means more consistent data.

8.2. System security

The restrictions that I will place are imperative to ensure that the data stays consistent and does not become corrupted.

Restricted function	**Reason and how**
Add New Data	I have restricted this, so If a team of people are given a new project and more than one of them attempt to add the new project or client into the database, only the project manager will be able to add new data into the system to avoid duplication. I have also made it so that when you're adding a project to the system that if it cannot have the same title as another project in the database.
Edit Data	I have this function restricted to the project managers because if many people try to edit the project the end result could be that they have incorrect data in their database. To aid the restriction I already have in place, I also have a function that prevents more than one person editing a project at the same time this will prevent the data from becoming corrupted.
Remove Data	I have this function restricted to project managers because if a project is removed from the database you can't get it back. This is a great responsibility so only the project managers can remove a project. To aid the restriction I already have in place I also have a function that prevents more than one person being able to remove a project at any time
Show Project Statistics	This function is restricted as it is a management tool to view how many man hours are required to complete the projects in the next month or quarter.

9. Validation

Data	Validation	Why it is necessary
JobID	Checked against other JobID'S if there are no JobID's with the same value then the JobID is valid **Range Validation**	This is necessary because the JobID is the primary key for the project table. This means that it would be very difficult to be able to sort through the projects if there was two or more JobID's that were the same.
Job Title	Must be below 50 characters **Range Validation**	This is necessary to prevent the database becoming too big. If they were allowed to enter large project names then you could end up using unnecessary memory on the harddrive. It would also make the program larger and cause restrictions on the ram.
Project Manager	Must be on the list of Project Managers **Range Validation**	This is necessary to prevent people who are not project managers being put in charge of a project.
Date Start	Must be a date in the future **Range Validation**	This prevents them entering in wrong data because they would not be able to take on a project that has already started. Although if they do forget to enter in a new project I have added a function for the Office Managers to use which will allow a date to be entered which is in the past.
Month finishing	Must be after date start **Range Validation**	This is been put in as it is impossible to finish before you start

Subcontractors value	Must be a float **Numeric Validation**	This prevents letters being entered into the subcontractors value
Status	Boolean **Range Validation**	There is no need for it not to be Boolean as status only has two options active or inactive so representing status as Boolean is the best way to make sure that invalid data is not entered into the system
Contact Value	Must be a float **Numeric Validation**	This prevents letters being entered into the contract value Also must state what currency that they are using since Element Energy work with people outside of the UK
Invoice number	Must not match any other invoice numbers **Range Validation**	This prevents two invoice numbers being the same which would be bad for the company. If they send off two invoices with the same number they would have to return the payment they receive and send a second invoice to correct their mistake.

10. Testing

10.1. Outline plan

10.1.1. Identification and explanation of suitable test strategies

Test Series	Purpose of test series	Testing Strategy	Strategy Rationale
1	Validation of input data performed corrected	Bottom-up Testing	Each component will be tested as it becomes available
2	To check the fetching of data is preformed	Bottom-up Testing	Each component will be tested as it becomes available
3	To check the reading of fetched data	Bottom-up Testing	Each component will be tested as it becomes available
4	To check the submission of data	Bottom-up Testing	Each component will be tested as it becomes available
5	To check the UI	Functional Testing	This type of testing ignores the internal parts and focus on the output is as per requirement or not
6	To check the GUI	Bottom-up Testing	Each component will be tested as it becomes available
7	To check the ability to break into the system	Unit testing	Testing of individual software components or modules.

10.2. Detailed plan

Test Series and number	Purpose	Description	Test Data	Test Data Type (Normal/Erroneous/Extreme)	Expected Result	Actual result	Evidence in appendix
1.1	Validate the JobID	This will check that there is no JobID's that are the same	1 1.5 -2 "abc"	Normal Erroneous Erroneous Erroneous	Accepted Error Error Error		
1.2	Validate Job title	This will check that there is no other job with the same title It will also check there are letters in the name	1 1.5 -2 "abc" "abc2"	Erroneous Erroneous Erroneous Normal Normal	Error Error Error Accepted Accepted		
1.3	Validate Project Manager	This will check that that project manager exists	1 2 Paul stevens Gary black	Erroneous Erroneous Normal Normal	Error Error Error Accepted		
1.4	Validate Date Start	This will check the date start is in the future and exists	10/11/12 10 December -10/11/12 10/11/96	Normal Erroneous Erroneous Erroneous Normal	Accepted Error Error Error Error		

1.5	Validate Date Finish	This will check the date start is before it and date finish exists	15/11/12 10 December -10/11/12 10/11/96	Normal Erroneous Erroneous Erroneous Normal	Accepted Error Error Error Error		
1.6	Validate Active	Checks that today's date is before the date start and if the date start is before the today date then the project managers are notified Also makes sure the it has a Boolean value	True False Hello 1 0 2 -1	Normal Normal Erroneous Normal Normal Erroneous Erroneous	Accepted Accepted Error Accepted Accepted Error Error		
1.7	Validate Value of Project	Checks that it is a float	10000.50 9086 123.1233 Hello True False	Normal Normal Normal Erroneous Erroneous Erroneous	Accepted Accepted Error Error Error Error		
1.8	Validate Invoice Numbers	Checks they aren't the same	10000.50#2 9086#2 100#5	Erroneous Normal Normal	Error Accepted Accepted		

			123.1233 Hello True False	Erroneous Erroneous Erroneous Erroneous	Error Error Error Error		
1.9	Validate Invoice Values	Checks that they are a float	10000.50 9086 123.1233 Hello True False	Normal Normal Normal Erroneous Erroneous Erroneous	Accepted Accepted Error Error Error Error		
1.10	Validate Subcontractor value	Checks that it is a float	10000.50 9086 123.1233 Hello True False	Normal Normal Normal Erroneous Erroneous Erroneous	Accepted Accepted Error Error Error Error		
1.11	Validate Amount paid	Checks that it is a float	10000.50 9086 123.1233 Hello True False	Normal Normal Normal Erroneous Erroneous Erroneous	Accepted Accepted Error Error Error Error		
1.12	Validate Username	Checks that it is longer than 8 characters	Bob_giller Dean_taylor	Normal Normal	Accepted Accepted		

		and that there is only one of it	Jim_diamond 123 Bob Elvis True	Normal Erroneous Normal Normal Errouneous	Accepted Error Error Error Error		
1.13	Validate Password	Checks that it is longer than 8 characters	Password Security afs76hui898 98io98 billabong	Normal Normal Normal Erroneous Normal	Error Error Accepted Error Accepted		
2.1	Check data is received from database	This checks that there is data been taken from the database and being shown via the system	Database	Project details Client details Staff details Subcontractor details Invoice details	respective details appear on screen for user to read		

3	Check able to read data	Checks that the data is in the correct format for the system to be able to read	Database	Project details Client details Staff details Subcontractor details Invoice details	Respective details able to manipulated by the program to produce new values such as work hours		
4	Checks that the data was submitted to the database	It checks the data submitted by calling it back from the database	Database	Project details Client details Staff details Subcontractor details Invoice details	Respective details found in database and confirmation on screen		
5.1	Checks the UI works	Checked manual by checking all paths and functions work correctly	User input via command line	Links numbered input Functions called from program	User able to work their way around the UI and use the full functionality of the program		
5.2	Checks usability of UI	The user needs to be able to work the program and the interface not being overly complicated	User input via command line	Links numbered input Functions called from program	User able to navigate the main menu and sub menus with ease		

6.1	Checks GUI	Checked manual by checking all paths and functions work correctly	User input via button clicking	Links via buttons Functions called from program	User able to work their way around the GUI and use the full functionality of the program		
6.2	Checks usability of GUI	The user needs to be able to work the program and the interface not being overly complicated	User input via command line	Links via buttons Functions called from program	User able to navigate the main menu and sub menus with ease		
7.1	Insertion of SQL	The user must not be able to insert their own SQL statements into the program via the GUI or UI	User input via GUI and UI	Deletion statements Update statements Add statements	Program output that this is invalid data		
7.2	Consultant login	The consultants must not be able to access the Project Manager menu	Consultant logins	False usernames and passwords Consultant usernames and passwords Project Manager usernames and passwords	False names and passwords will not access any menus All consultant names and passwords only access consultant menu Project Managers able to access the project Manager menu		