Software Requirements Specification for Code Conversion

Version 3

Prepared by

- ❑ احمد معتز عبد الفتاح عبد اللطيف سليم 202301420
- ❑ نجلاء أحمد محمود أحمد 202301608
- ❑ سامح طه شرف هاشم 202301563

Graduate School of Statistical Research

May 2024

Table of Contents

Revision History

Name	Date	Version
Code Conversion	8 March 2024	1
Code conversion	20 April 2024	2
Code conversion	2 May 2024	3

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to detail the requirements for the Code Conversion module. This module is responsible for converting code from one programming language to another while maintaining the functionality and structure of the original code.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- Typical Users, such as students, who want to use Code Conversion for learn different syntax between languages (java , C#, python, etc.)
- Advanced/Professional Users, such as engineers or researchers, who want to use Code Conversion for more learning.
- Programmers who are interested in working on the project by further developing it or fix existing bugs.

1.4 Product Scope

The web application aims to be a user-friendly platform for people at all skill levels, offering syntax conversion between programming languages, easy access to programming tutorials while making sure it is free and easy to use and in someday it can be sponsored and adapted by major companies

1.5 References

Code Conversion ’s website:

https://CodeConversion .org/

Code Conversion ’s GitHub

page: https://github.com/Code

Conversion

IEEE Template for System Requirement Specification Documents:

https://goo.gl/nsUFwy

GNU General Public License version 3:

http://www.gnu.org/licenses/gpl.html

CDDL Common Development and Distribution License:

https://opensource.org/licenses/CDDL-1.0

2. Overall Description

2.1 Product Perspective

System Architecture

The Code Conversion module will be integrated into an existing software system and will serve as a standalone feature within the larger system architecture. The module will interact with other system components through APIs and user interfaces to receive input code and deliver the converted code.

API Integration: The Code Conversion module will provide an Application Programming Interface (API) for seamless integration with other software components, allowing for automated code conversion processes.

User Interface (UI): A graphical User Interface (GUI) will be provided for users to input code snippets or files and retrieve the converted code.

2.2 Product Functions

- Users can choose the desired programming language and the desired control type to view the syntax for it
- Users can choose multiple programming languages and compare between their syntax in a
- specific control type

- Admin will be able to login to perform CRUD operations for tutorials and programming languages.
- Users can view categorized tutorials based on programming languages.

2.3 User Classes and Characteristics

- Typical Users, such as students, who want to use Code Conversion for analyzing networks (Social networks, Social Media networks, Semantic networks etc.)
- Advanced/Professional Users, such as engineers or researchers, who want to use Code Conversion for more demanding Code analysis.
- Programmers who are interested in working on the project by further developing it or fix existing bugs

2.4 Operating Environment

- Windows 2000
- Windows XP
- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Mac OS X
- Linux

2.5 Design and Implementation Constraints

Code Conversion is developed in Python, it uses Flask as the ORM and has been built on top of the Flask Framework, SQL-alchemy , html and JavaScript using Vs Code editor. It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other

2.6 User Documentation

There is a quick start guide available on the website of Code Conversion: https://Code Conversion .org/users/quick-start/

And one for layouts:
https://Code Conversion .org/users/tutorial-layouts/

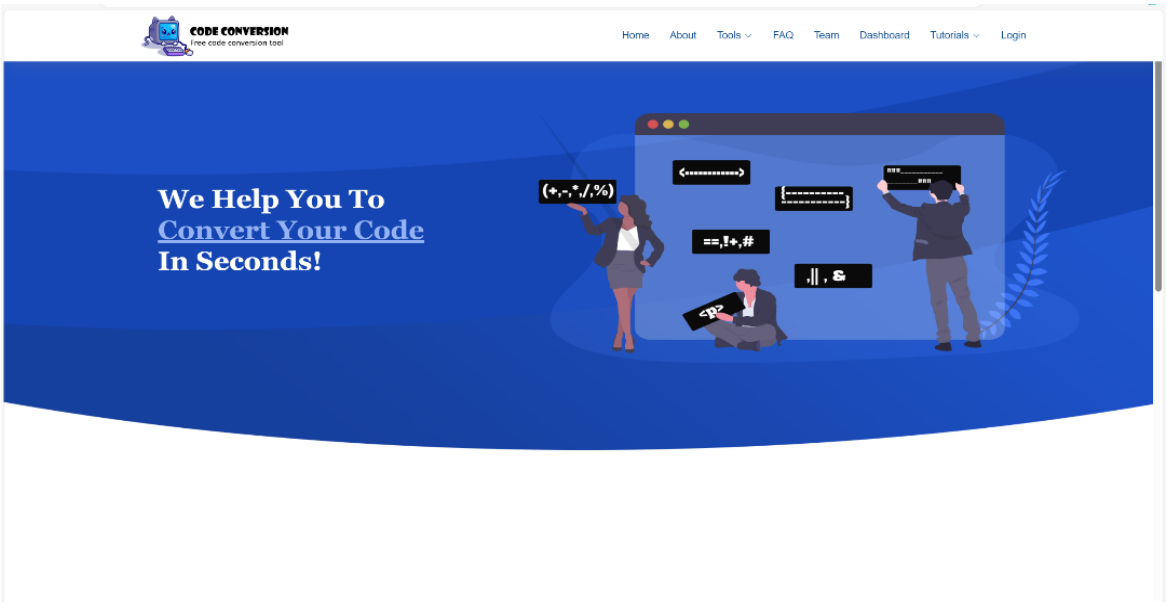
2.7 Assumptions and Dependencies

Code Conversion is developed in Python and therefore requires python interpreter to be installed on the user's system or bundled with the software. The latest stable version of Code Conversion requires python version 3.12 or higher and flask orm with Vscode . This applies to Windows and Linux users. On Mac OS X, interpreter is bundled with the application. Also, we assume that programming Languages syntax will remain the same will not be changed

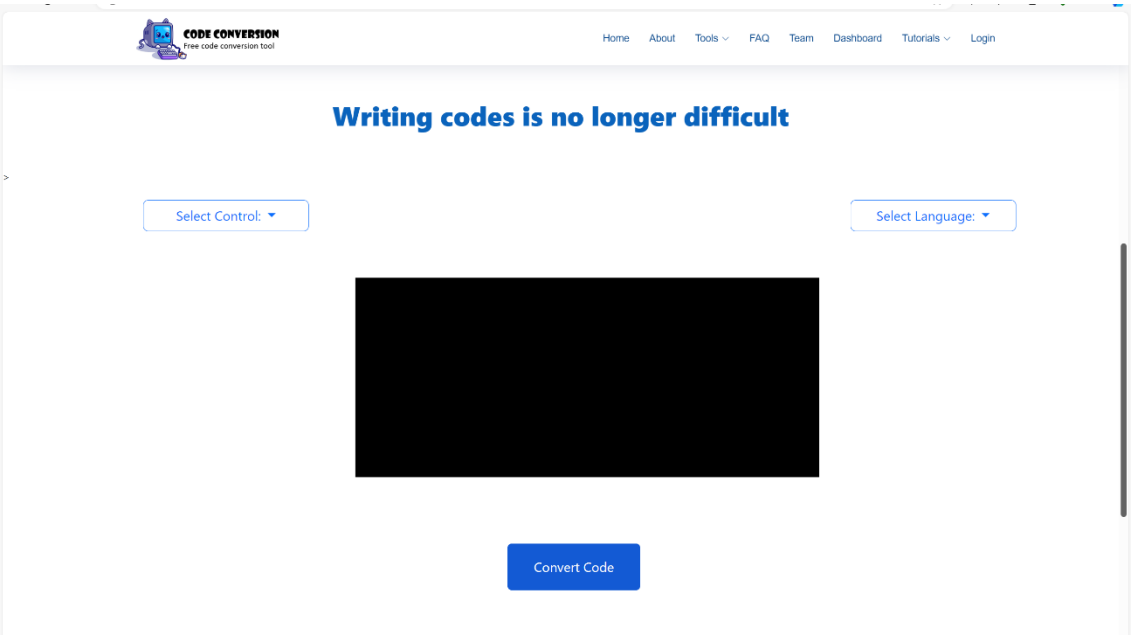
3. External Interface Requirements

3.1 User Interfaces

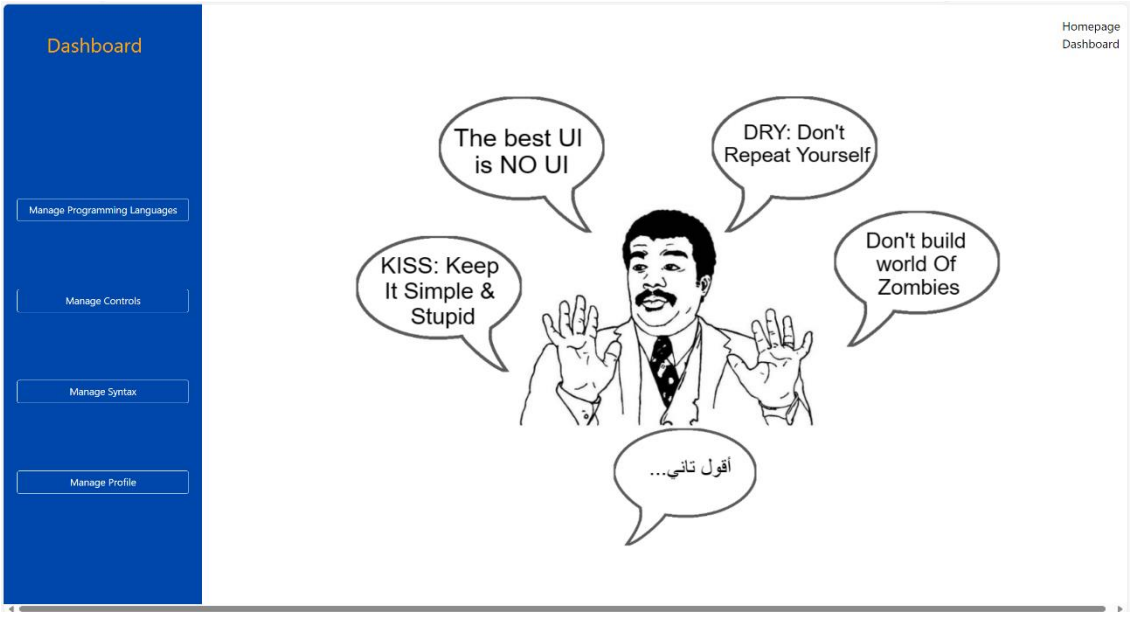
1. Code Conversion's Welcome Screen:



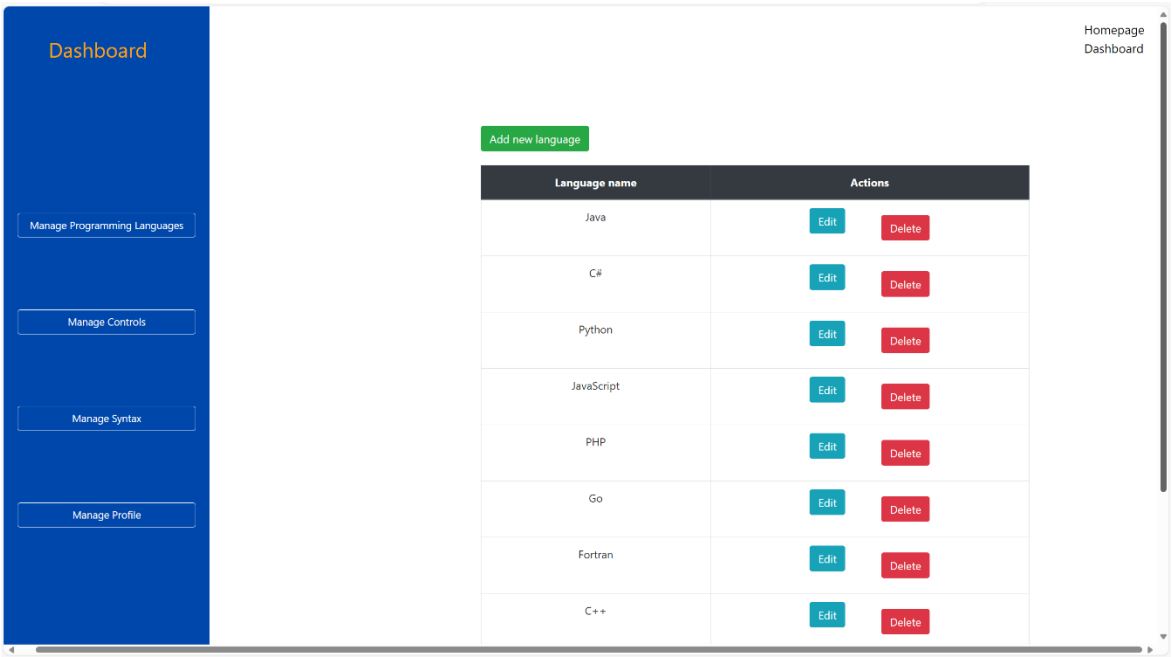
2. Code Conversion's Main Screen:



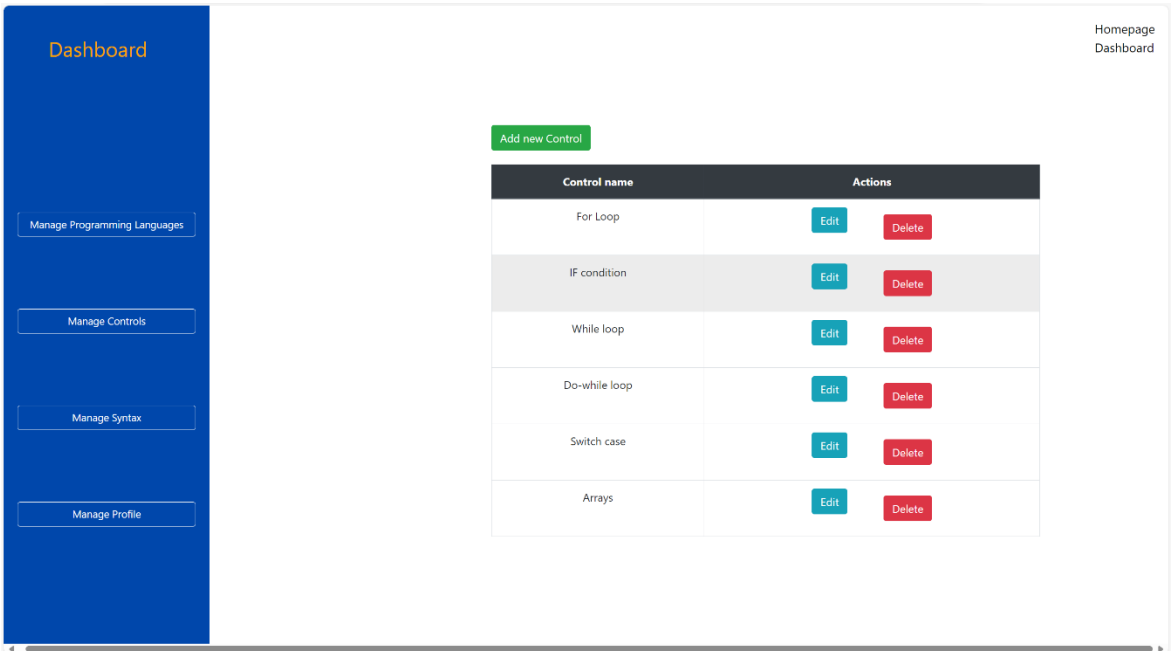
3. Main Dashboard Page



4. Manage Programming Languages page



5. Manage Controls Page



6. **Overview of the Website.**

- Home Page

Menu bar Buttons, from left to right:

- Home:

Takes the user back to the Home Page

- About

Give the user more information about the website vision and scope

- Tools
  - Single Syntax Conversion
    - Takes the user to the single syntax conversion page
  - Multi Syntax Conversion
    - Takes the user to the single syntax conversion page

- FAQ:

Answers the Frequently Asked Questions

- Team

Presents the owner of the site

- Dashboard

Takes the admin to the Main Dashboard page

- Tutorials
    - How to use

      provides brief guide for the website

    - Languages

      Shows submenus of most popular languages

        - High Level Languages

          Take the user to the page shows the details about High Level languages

        - Low Level Languages

          Take the user to the page shows all the details about Low Level languages

        - Programming Skills

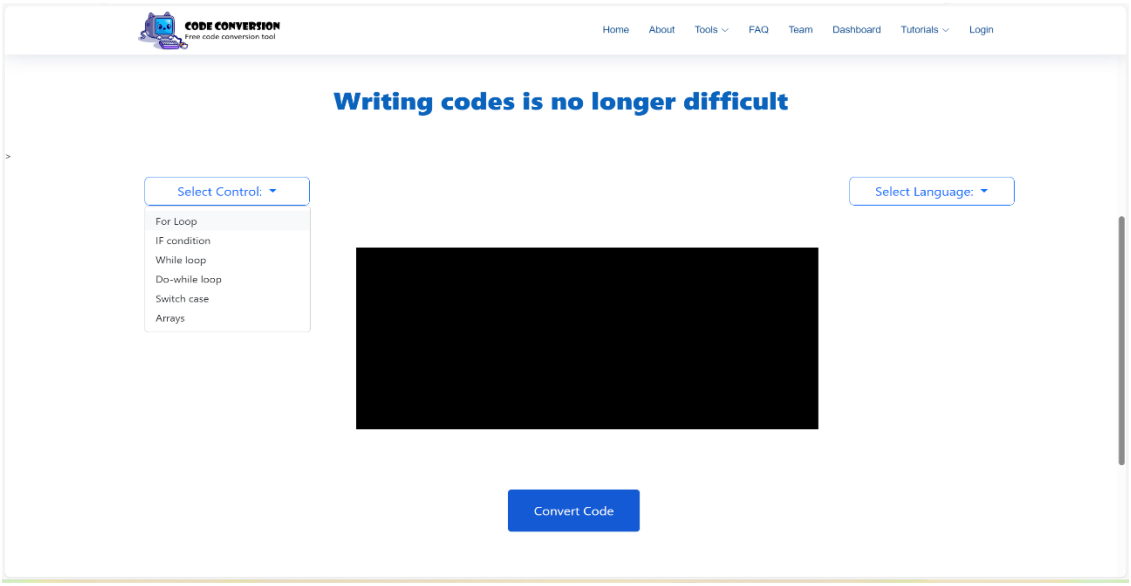
          Take the user to the page shows the skills needed for programming.

- Login

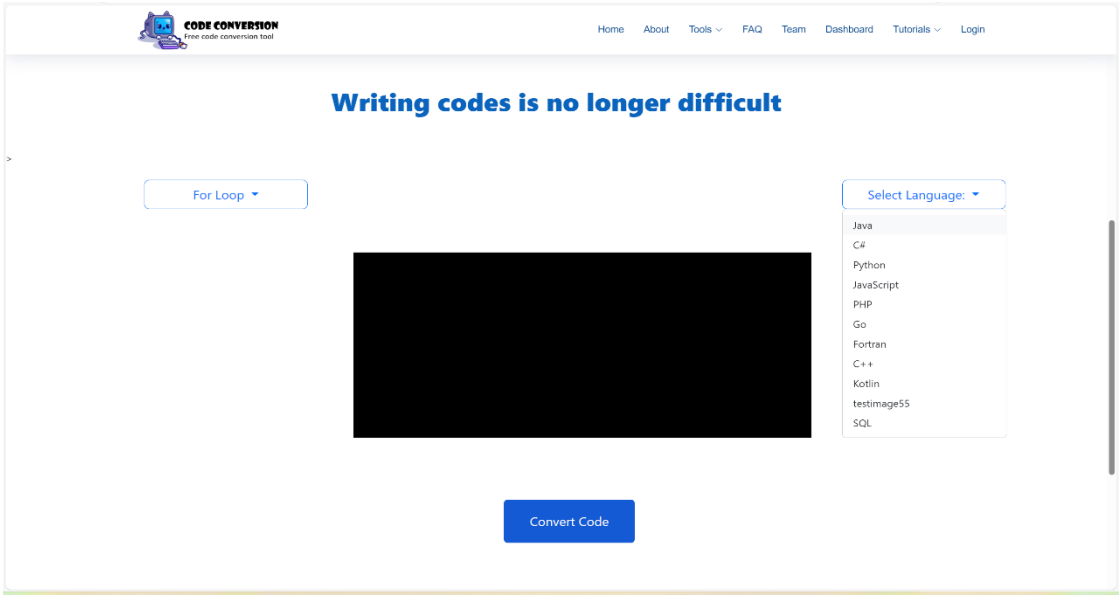
Enable the user to access his account.

At the Middle of the page

Drop Down menu for choosing the Programming language.

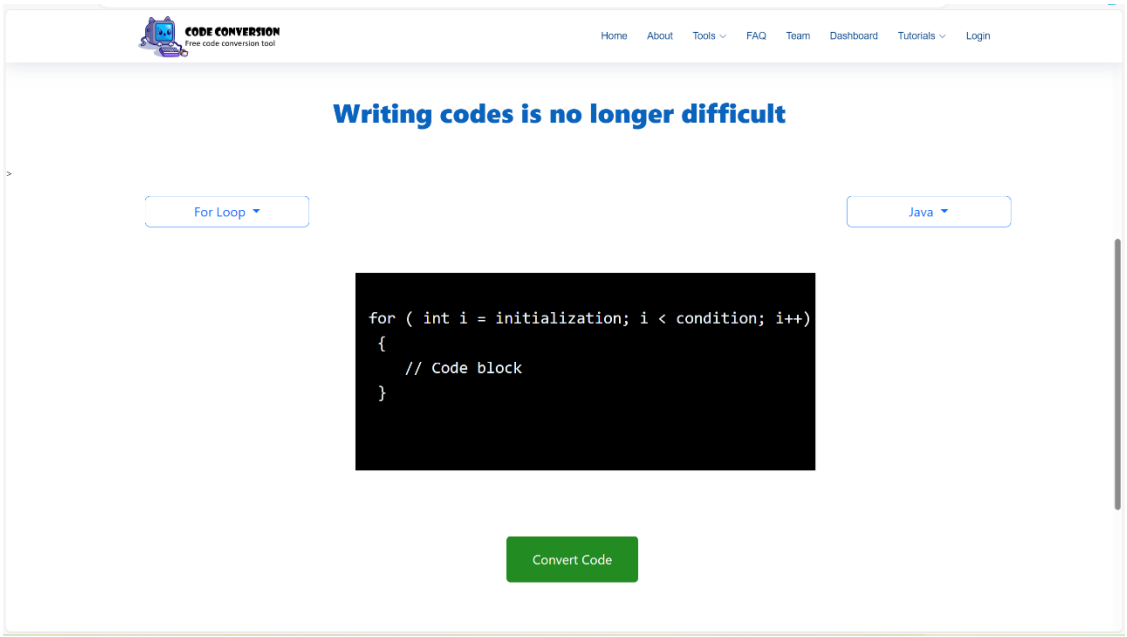


Drop Down menu for choosing the Programming Statement.

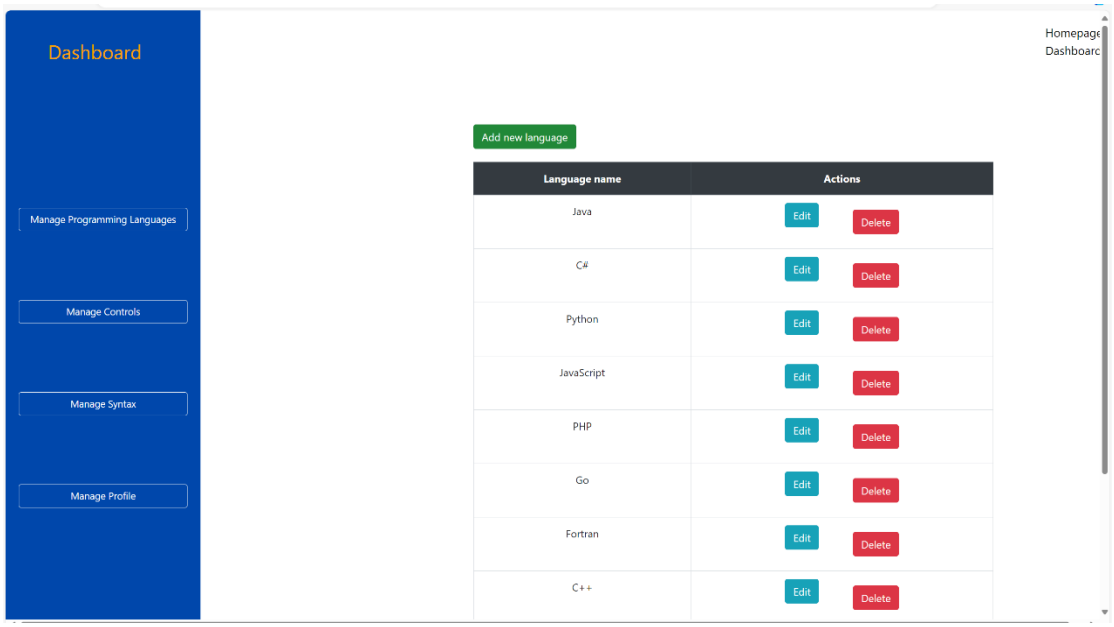


Convert Button:

For showing the statement in the black textbox



- DashBoard- Manage programming language Button:



Add New language:

For adding new Language

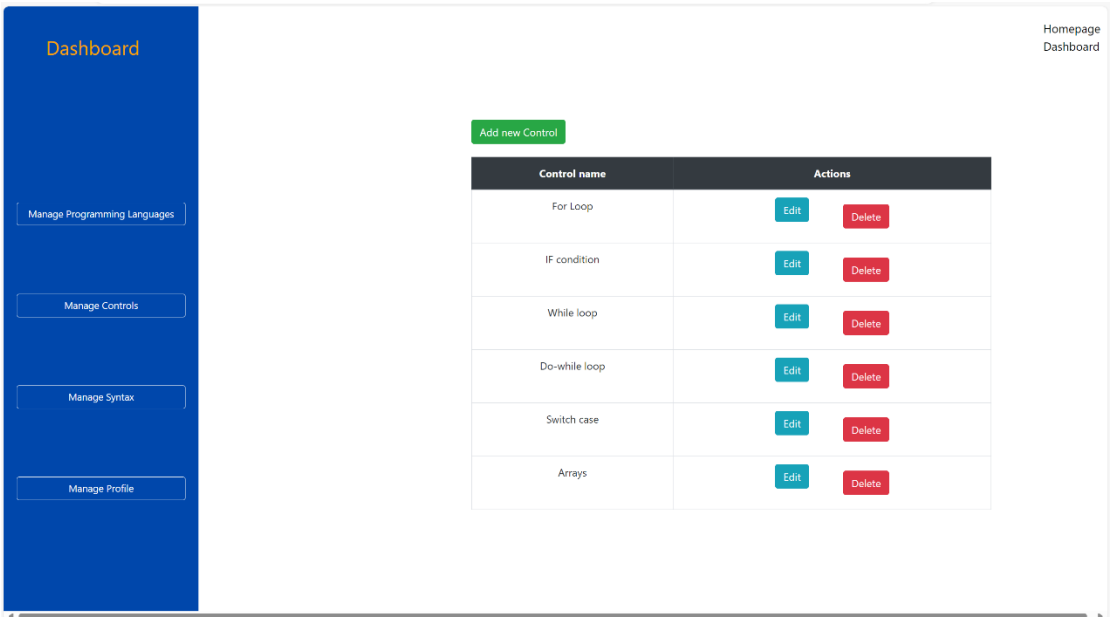
Edit language Button:

For editing language

Delete language Button:

For deleting the language from the list

- DashBoard- Manage Controls Button:



Add new Control

Add New Control:

For adding new control

Edit

Edit Control Button:

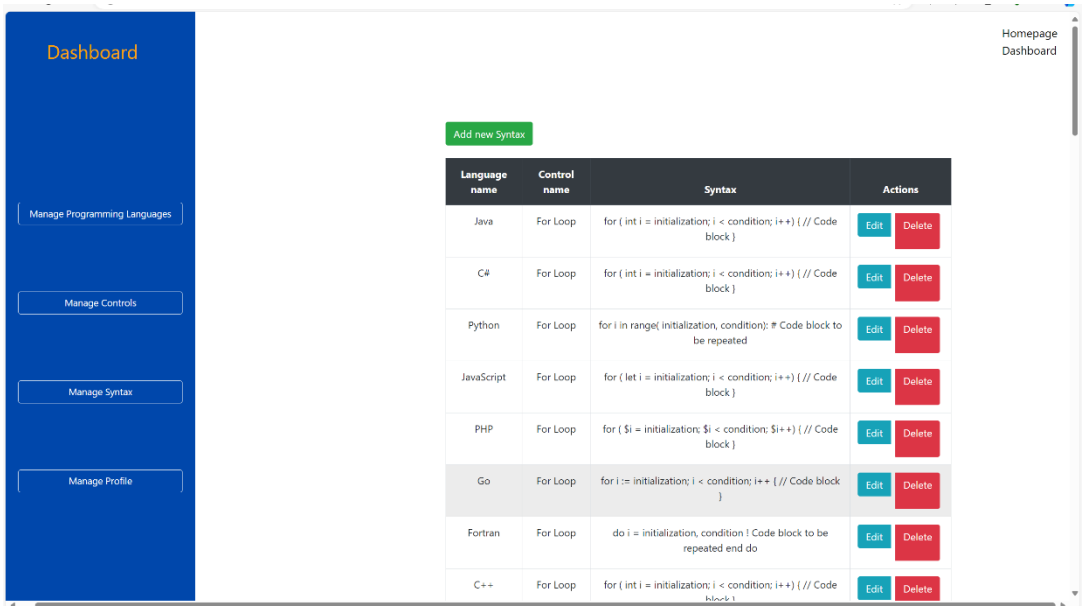
For editing Control

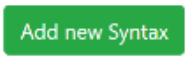
Delete

Delete Control Button:

For deleting the Control from the list

- <u>DashBoard- Manage Syntax Button:</u>





Add New Syntax:

For adding new Syntax

Edit Syntax Button:

For editing the Syntax

Delete Syntax Button:

For deleting the Syntax from the list

3.2 Hardware Interfaces

The minimum hardware requirements of Code Conversion Website are a 500-Megahertz CPU and 128 megabytes of RAM. Also, a compatible graphics card is required.

3.3 Software Interfaces

Python requires to be installed on the system, more specifically flask, virtual environment, vscode, SQL Alchemy in their latest release.

Additional information can be found on section 2.7 of this document.

Code Conversion website can be connected with a PLSQL to add, edit, delete data from the database

3.4 Communications Interfaces

Code Conversion website requires an internet connection to access via any browser.

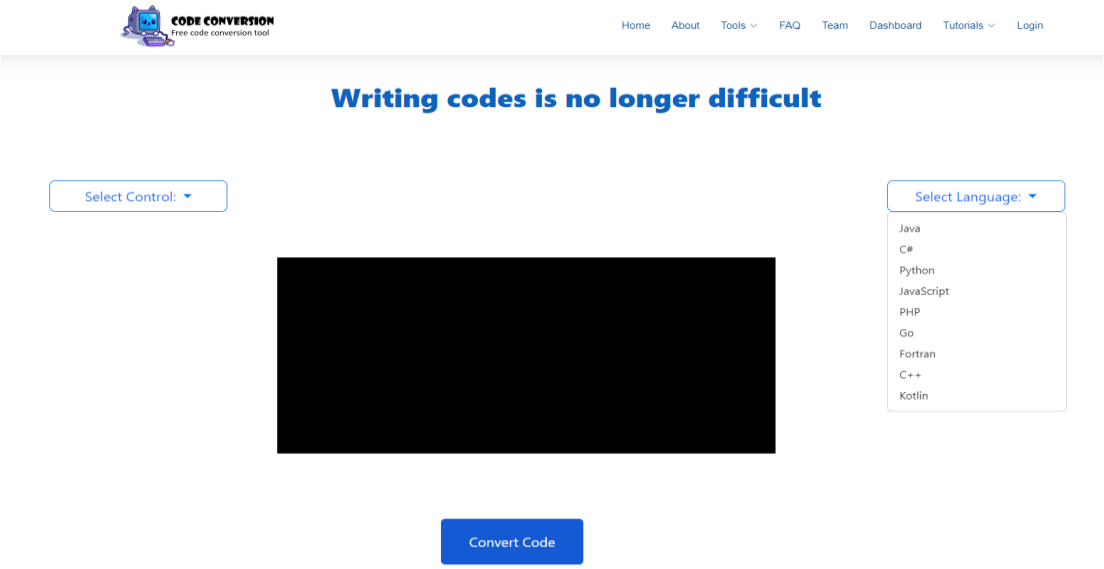
4. System Features

4.1 Syntax conversion

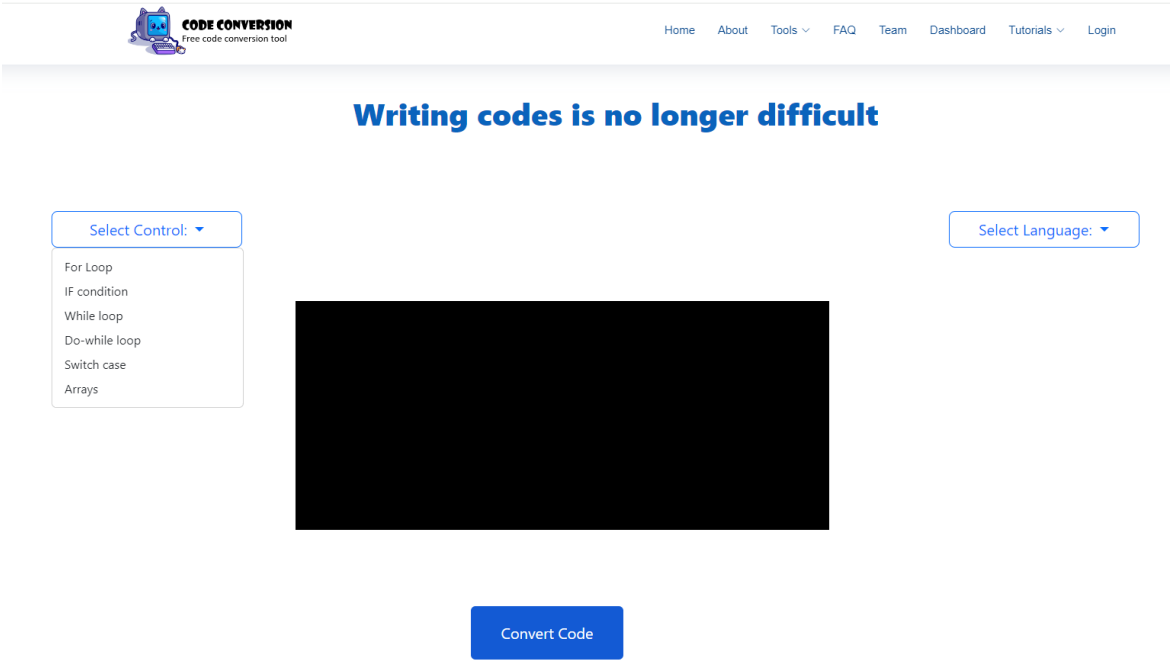
4.1.1 Single syntax convert

Users can directly interact with the dropdown list by clicking on it to choose a programming language to convert from also users can choose from the second dropdown list the control type to view corresponding syntax for that language.

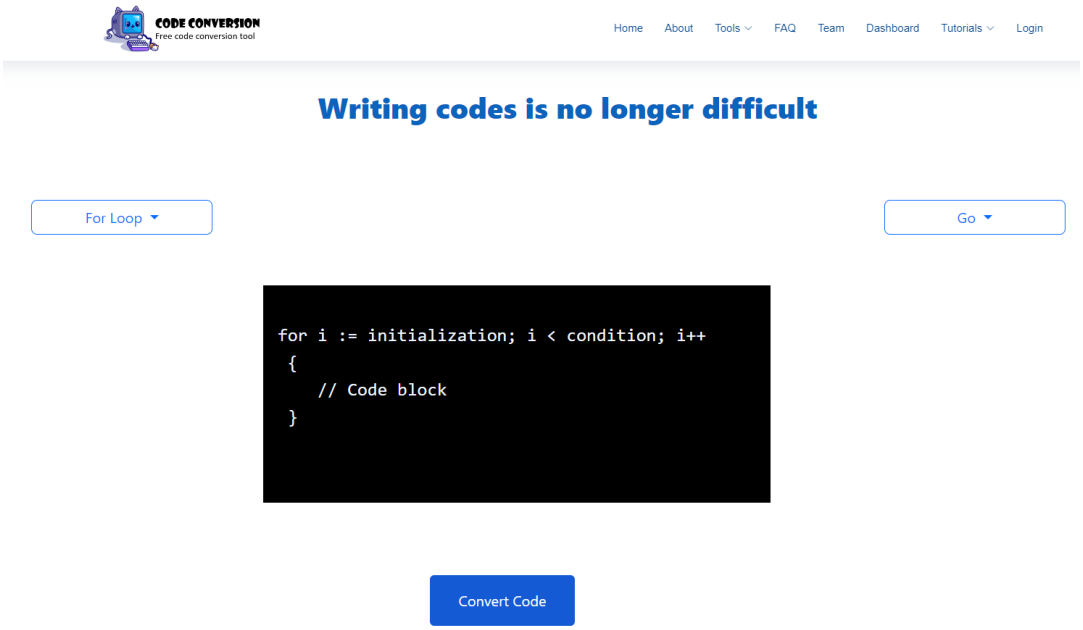
4.1.1.1 Choosing programming language:



4.1.1.2 Choosing control type:



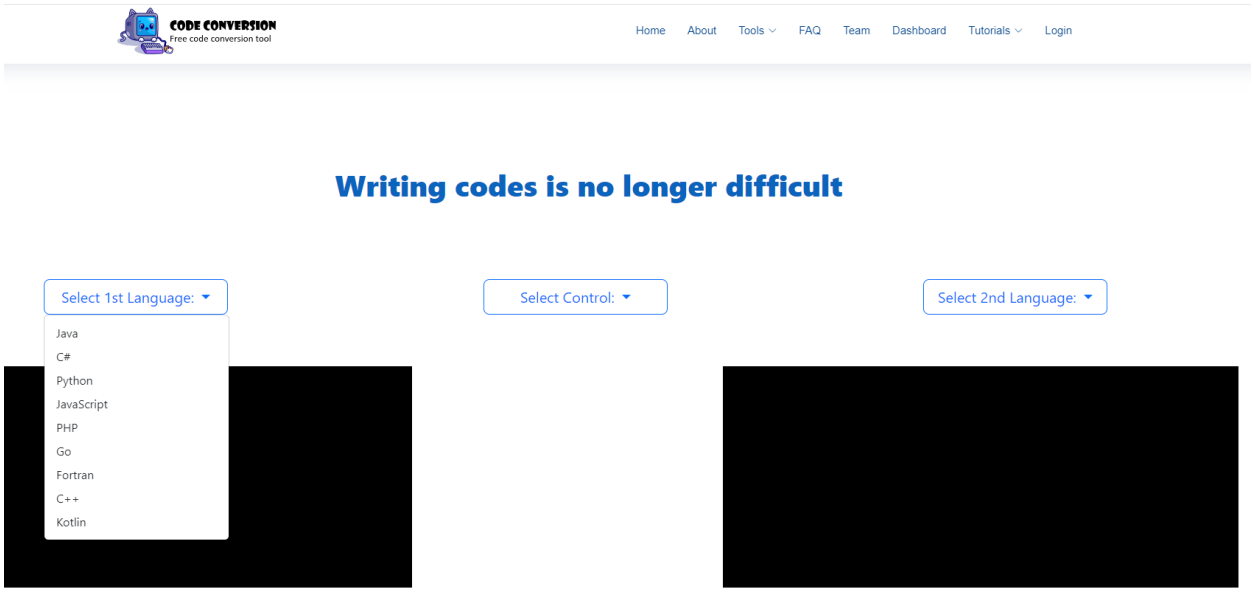
4.1.1.3 Final output after conversion



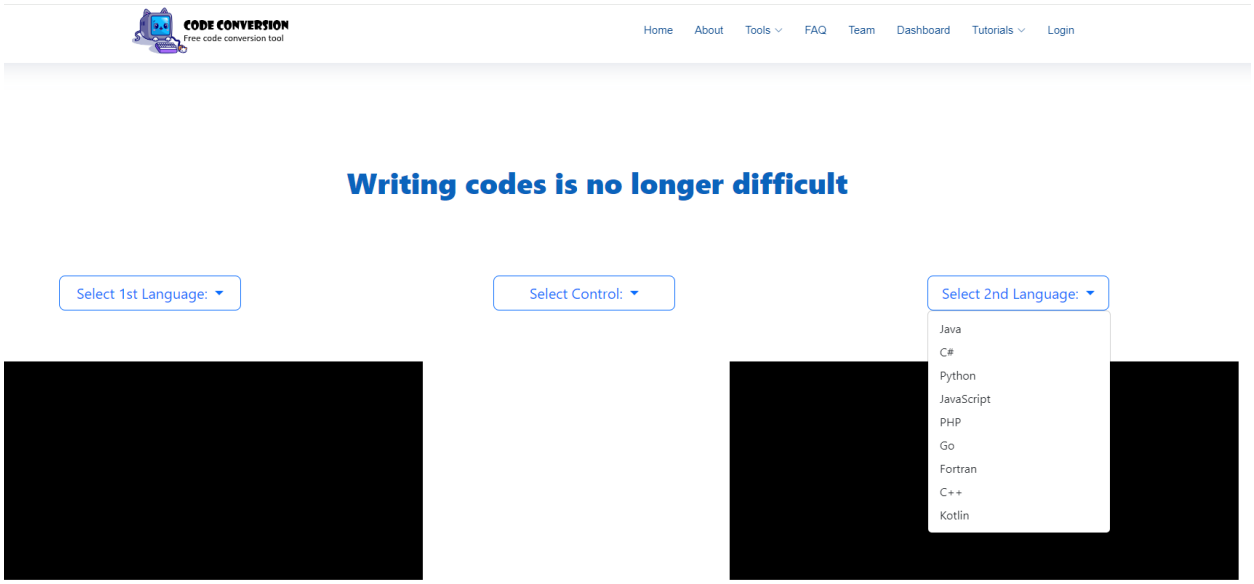
4.1.2 Multi syntax convert

Users can directly interact with the first dropdown list by clicking on it to choose first programming language to convert from also users can choose from the second dropdown list to choose second programming language to convert to and then users will select desired control type to view compare between both languages in that specific control type

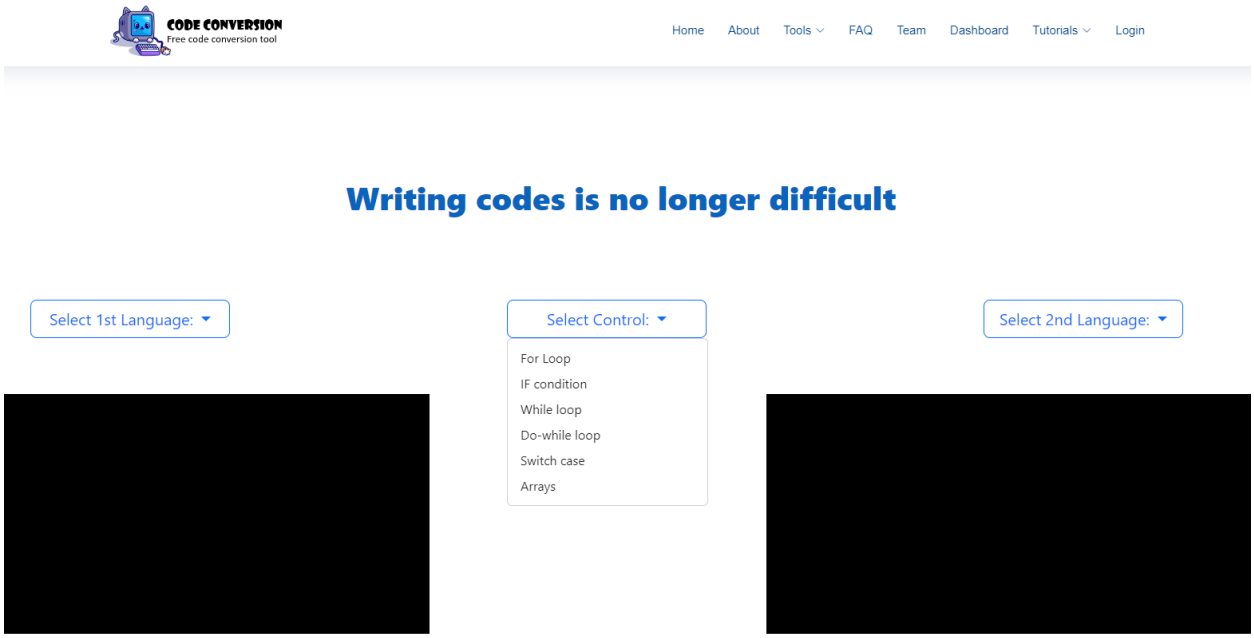
4.1.2.1 Choosing first programming language:



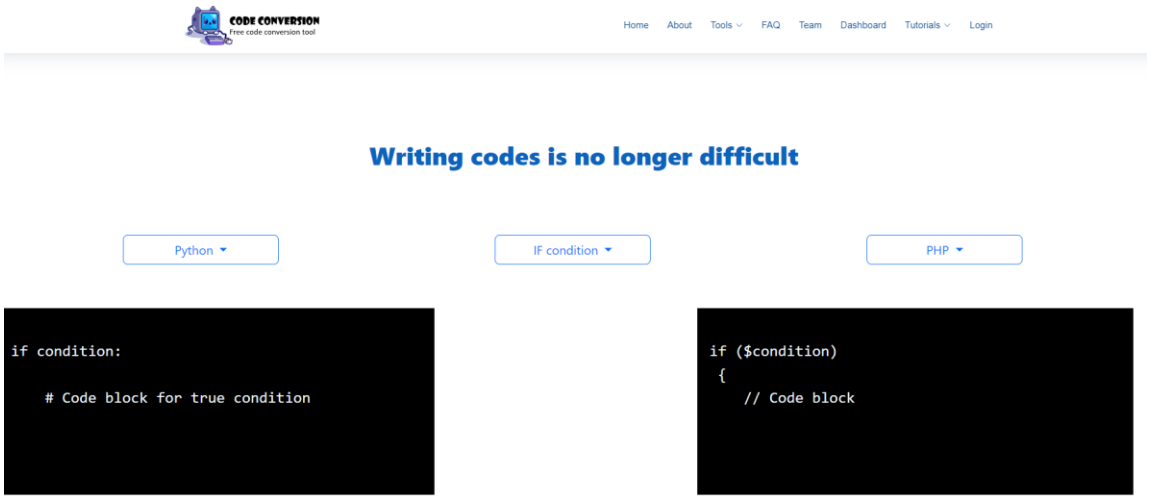
4.1.2.2 Choosing second programming language:



4.1.2.3 Choosing control type:



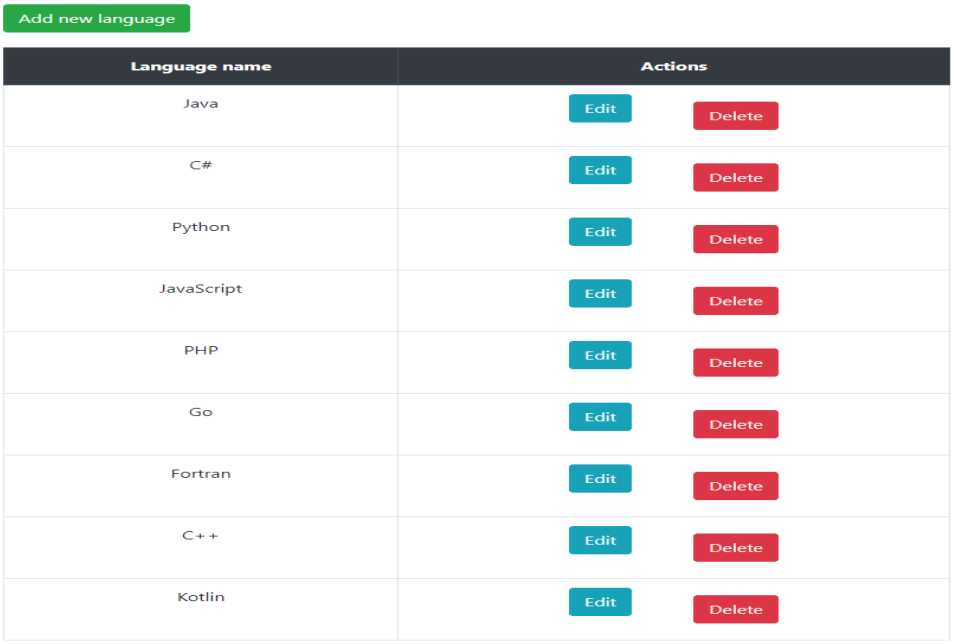
4.1.2.4 Final output after conversion:



4.2 Admin dashboard

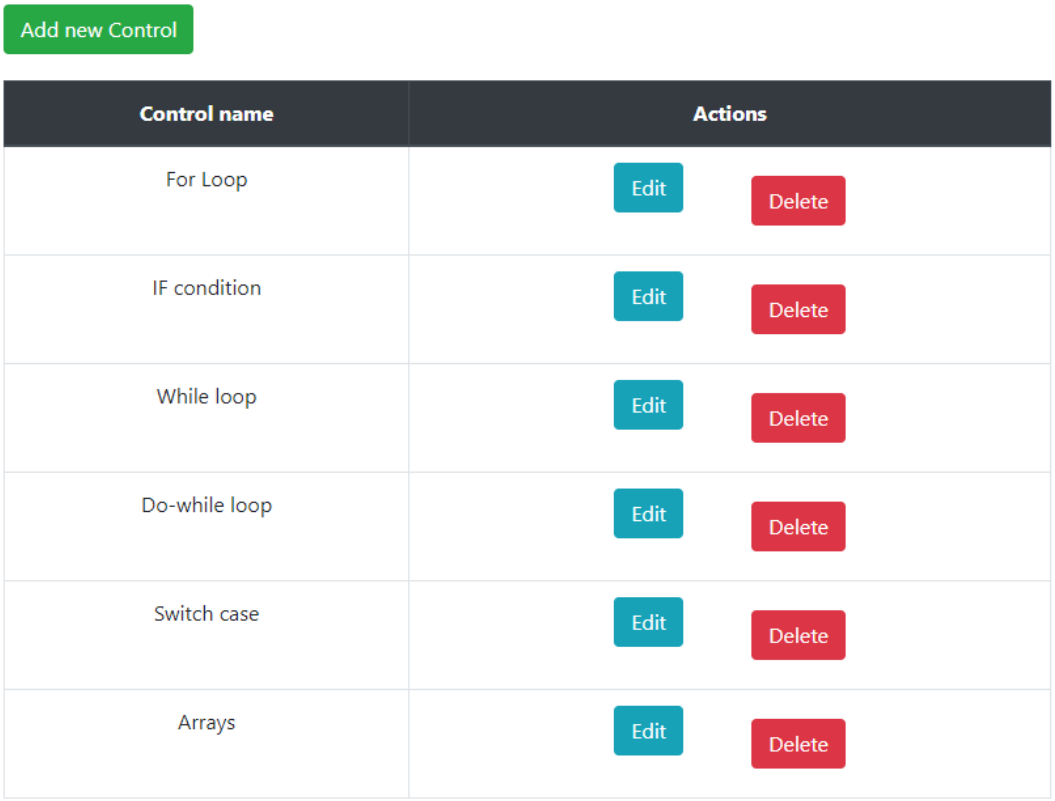
4.2.1 programming language management:

Data of the programming languages are presented in tables where the admin can have full control on performing crud operation on any programming language



4.2.2 control type management:

Data of the control types are presented in tables where the admin can have full control on performing crud operation on any control type



4.2.3 syntax management

Data of the programming languages syntaxes are presented in tables where the admin can have full control on performing crud operation on any syntax by choosing appropriate programming languages and control types

Add new Syntax

Language name	Control name	Syntax	Actions
Java	For Loop	for (int i = initialization; i < condition; i++) { // Code block }	Edit Delete
C#	For Loop	for (int i = initialization; i < condition; i++) { // Code block }	Edit Delete
Python	For Loop	for i in range(initialization, condition): # Code block to be repeated	Edit Delete
JavaScript	For Loop	for (let i = initialization; i < condition; i++) { // Code block }	Edit Delete
PHP	For Loop	for ($i = initialization; $i < condition; $i++) { // Code block }	Edit Delete
Go	For Loop	for i := initialization; i < condition; i++ { // Code block }	Edit Delete
Fortran	For Loop	do i = initialization, condition ! Code block to be repeated end do	Edit Delete
C++	For Loop	for (int i = initialization; i < condition; i++) { // Code block }	Edit Delete

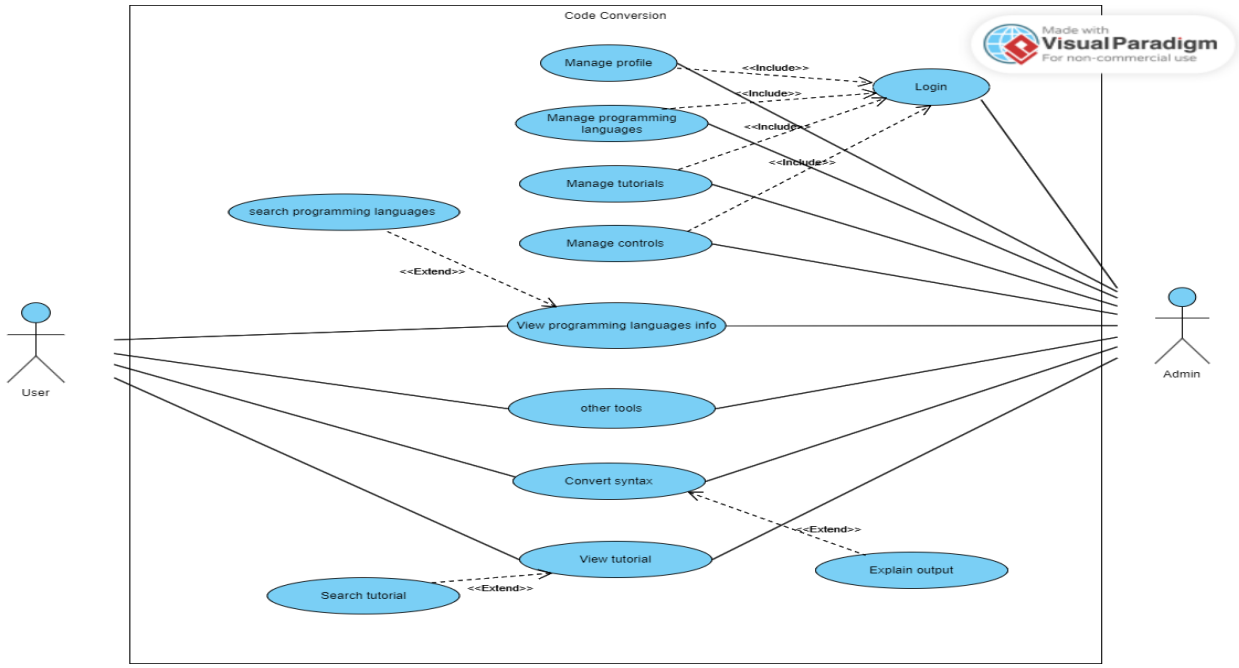
4.3 Modeling

4.3.1 Use Case diagrams:

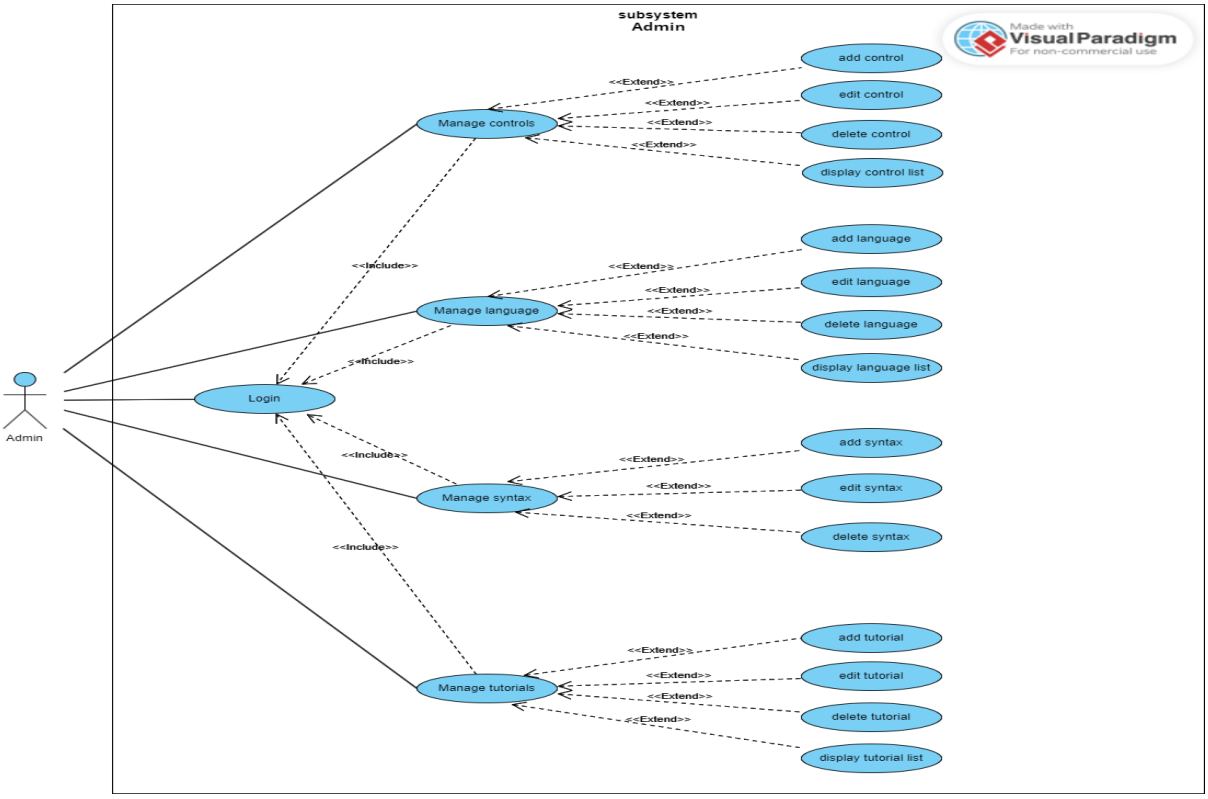
4.3.1.1 Definitions, acronyms, and abbreviations:

Actors	Shown in the diagram as stick figures with a name underneath. They represent elements that will be directly interacting with the system.
Use Cases	Oval shapes that have their names in the center. These represent direct functionality within the system that must be implemented.
Interactions	Lines that connect the actors with the different Use Cases. These show that there is some form of direct interaction between the actor and that specific functionality.
Includes	Dotted lines labeled "<<include>>" that connect two use cases and have an arrow pointing towards one. This means that the use case without the arrow calls on the functionality of the use case with the arrow.
Extends	Dotted lines labeled "<<extend>>" that connect two use cases and have an arrow pointing towards one. This means that the use case without the arrow takes all of the functionality of the use case with the arrow and adds extra functionality.
The System Boundary	The large rectangle that contains the Use Cases. Everything within the rectangle is what the system is responsible for implementing
Use Case Template	Describes the basic functionality and features of each use case and the can be found in the pages following the use case diagram.

4.3.1.2 Use Case diagram 1:



4.3.1.3 Use Case diagram 2:



4.3.1.4 Use Cases Event Flow:

Use case name	Manage Profile
Participating Actors	Admin
Entry condition	admin wants to edit his profile
Exit condition	profile is updated successfully
Event flow:	1. Admin visits profile page 2. Admin make changes to profile and save it 3. Profile is update successfully

Use case name	Manage Programming languages
Participating Actors	Admin
Entry condition	admin wants to add, edit or delete programming languages

Exit condition	changes are successfully saved
Event flow:	1. Admin visits programming languages management page 2. Admin make changes and save it 3. Changes are successfully saved

Use case name	Manage tutorials
Participating Actors	Admin
Entry condition	admin wants to add, edit or delete tutorials
Exit condition	changes are successfully saved
Event flow:	1. Admin visits tutorials management page 2. Admin make changes and save it 3. Changes are successfully saved

Use case name	Manage controls
Participating Actors	Admin
Entry condition	admin wants to add, edit or delete controls
Exit condition	changes are successfully saved
Event flow:	1. Admin visits controls management page 2. Admin make changes and save it 3. Changes are successfully saved

Use case name	login
Participating Actors	Admin
Entry condition	admin wants to login to website
Exit condition	login was successful
Event flow:	1. Admin press login button 2. Admin writes username and password 3. Login was successful

Use case name	View programming languages info

Participating Actors	Admin, User
Entry condition	actors want to know about languages background and usage scenarios
Exit condition	information is successfully displayed
Event flow:	1. actors visits programming languages details page 2. actors choose desired language to display its details 3. Details are successfully displayed

Use case name	search programming languages
Participating Actors	Admin, User
Entry condition	actors want to search for specific language details
Exit condition	details are successfully displayed
Event flow:	1. Actors visit languages details page 2. Actors search for specific language 3. Details are successfully displayed

Use case name	convert syntax
Participating Actors	Admin, User
Entry condition	actors want to view a control syntax for a specific programming language or compare between multiple languages
Exit condition	syntax is successfully displayed
Event flow:	1. Actors select desired control from dropdown menus 2. Actors select desired language from dropdown menus to convert 3. Results are displayed successfully

Use case name	view tutorial
Participating Actors	Admin, User
Entry condition	actors want to view tutorial page categories
Exit condition	tutorial categories page is displayed successfully
Event flow:	1. Actors visit tutorial page 2. Tutorial is successfully displayed

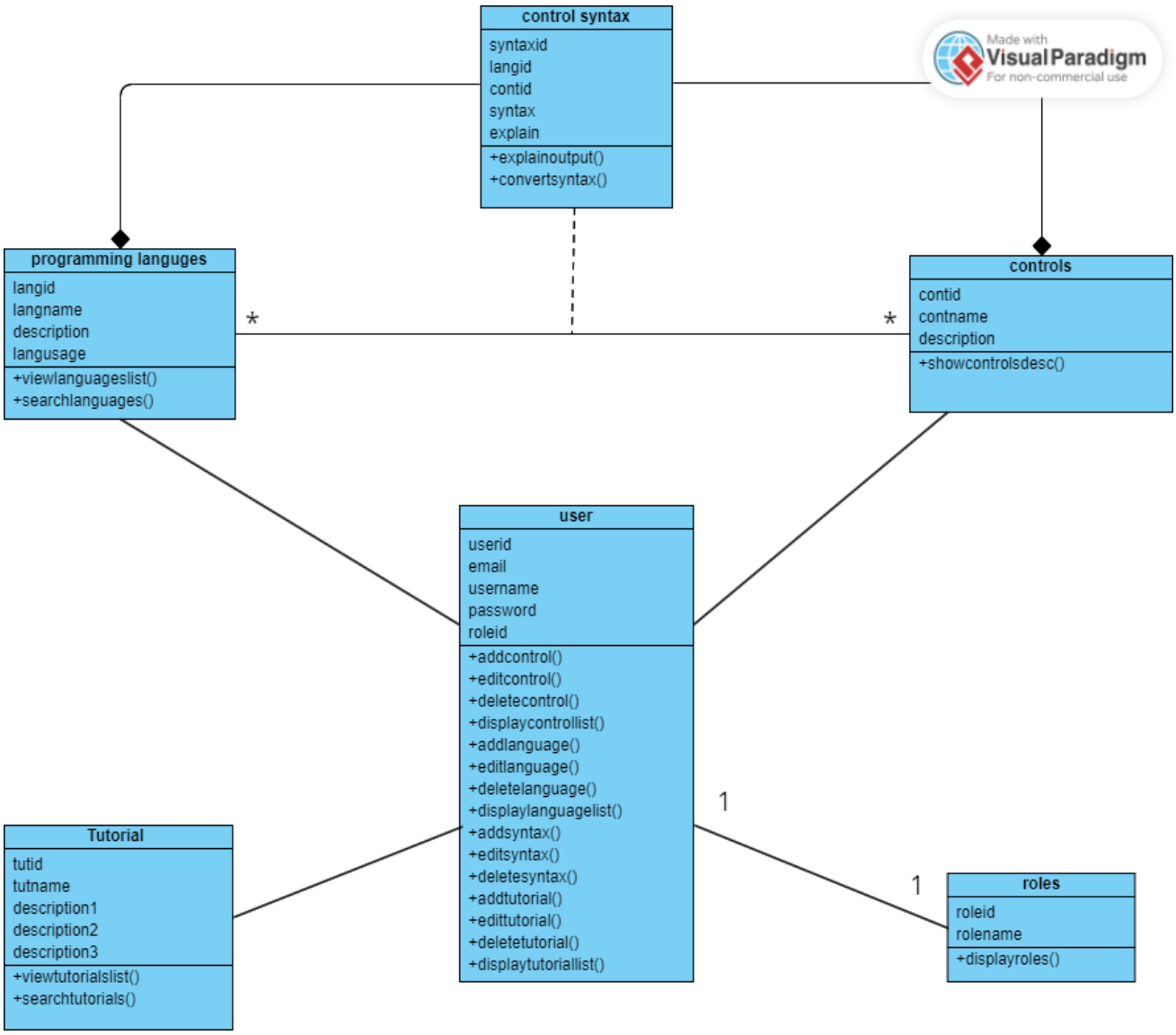
Use case name	search tutorial
Participating Actors	Admin, User
Entry condition	actors want to search for specific tutorial
Exit condition	desired tutorial is successfully displayed
Event flow:	1. Actors visit tutorial page 2. Actors search for specific tutorial 3. Tutorial is successfully displayed

4.3.2 Class diagrams:

4.3.2.1 Definitions, acronyms, and abbreviations:

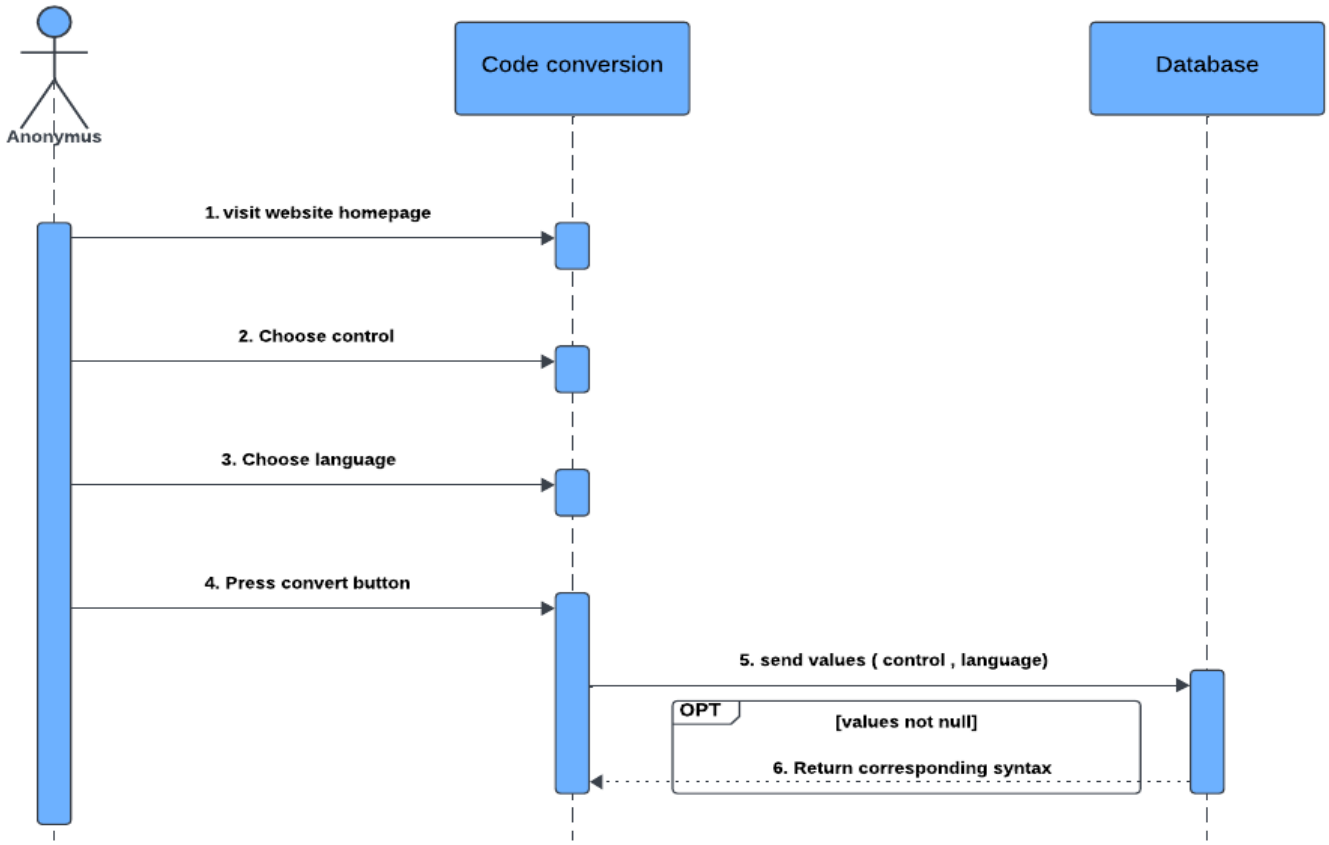
Classes	Rectangles in the diagram that are split into three parts. The top section is the name of the class, the middle section is the list of variables that are stored in the class and the bottom section is the list of functions in the class. These rectangles represent objects within the system.
Variables	These have a name followed by a semicolon and then a type. The type denotes what kind of data can be stored in the variable.
Functions	These have a name followed by a list of any variable that the function receives in-between the parenthesis "()". After that there is a semicolon and any variables that the function may return, if none it will be void.
Generalizations	Shown using a line from one object to the other with an unfilled triangle on one end. The object without the triangle inherits the functionality and variables from the object that has the triangle pointing towards it.
Aggregations	Lines that have an unfilled diamond on one end. This means the object with the diamond contains the object(s) without the diamond. This may have numbers on the ends (multiplicities).
Associations	Lines connecting two classes that can have a name beside it, may point in one direction, and may have numbers at the ends (multiplicities). These designate some relationship between the objects. Arrows are simply there to assist you in recognizing which direction the name of the association is read.
Multiplicities	Numbers that may be on the ends of Aggregations and Associations. They state how many of the one object can be

4.3.2.2 Class diagram:

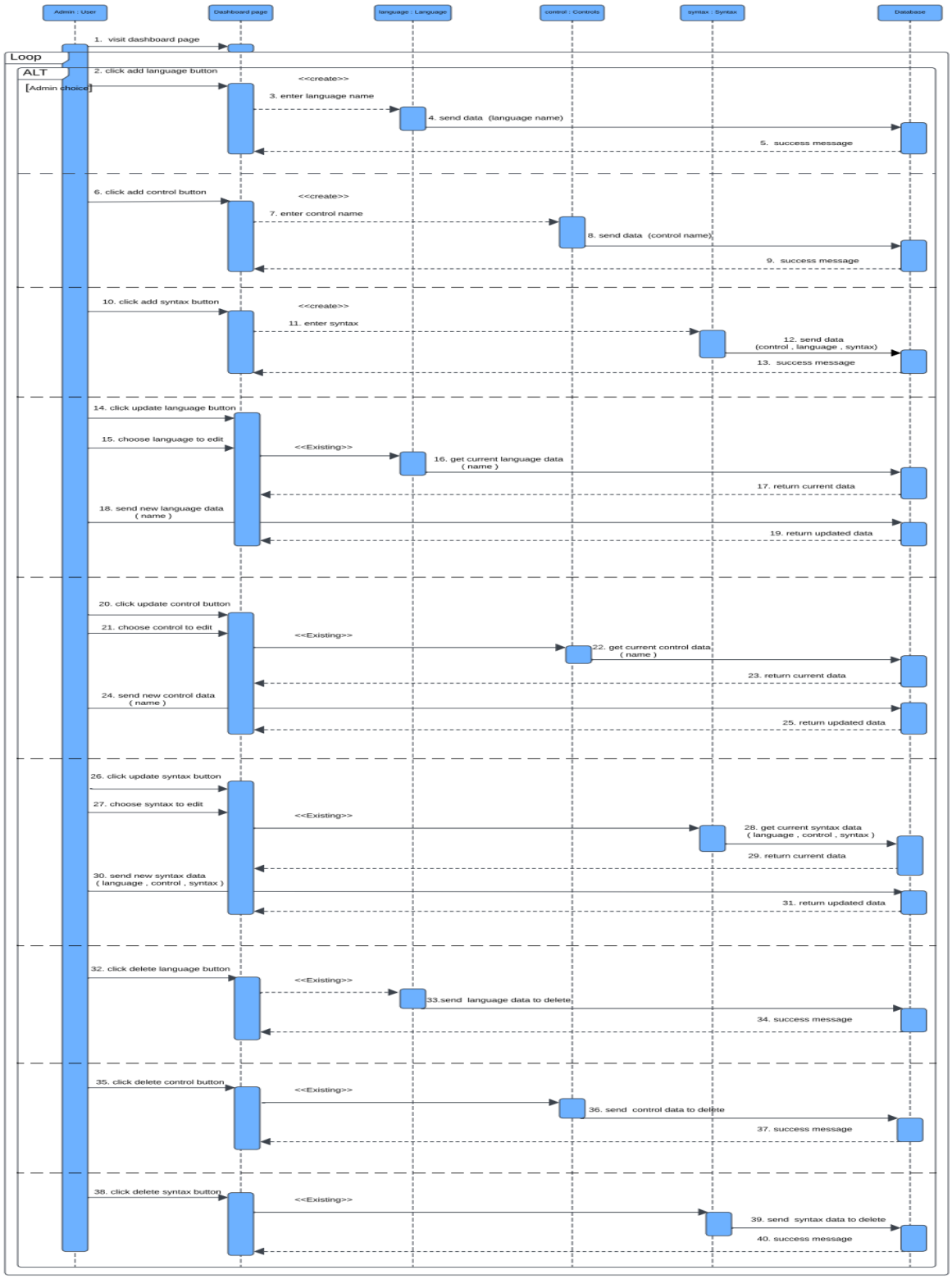


4.3.3 Sequence diagram:

4.3.3.1 Sequence diagram 1



4.3.3.2 Sequence diagram 2


Admin : User
Dashboard page
language : Language
control : Controls
syntax : Syntax
Database
Loop
ALT
[Admin choice]
1. visit dashboard page
2. click add language button
<<create>>
3. enter language name
4. send data (language name)
5. success message
6. click add control button
<<create>>
7. enter control name
8. send data (control name)
9. success message
10. click add syntax button
<<create>>
11. enter syntax
12. send data
(control , language , syntax)
13. success message
14. click update language button
15. choose language to edit
<<Existing>>
16. get current language data
(name)
17. return current data
18. send new language data
(name)
19. return updated data
20. click update control button
21. choose control to edit
<<Existing>>
22. get current control data
(name)
23. return current data
24. send new control data
(name)
25. return updated data
26. click update syntax button
27. choose syntax to edit
<<Existing>>
28. get current syntax data
(language , control , syntax)
29. return current data
30. send new syntax data
(language , control , syntax)
31. return updated data
32. click delete language button
<<Existing>>
33.send language data to delete
34. success message
35. click delete control button
<<Existing>>
36. send control data to delete
37. success message
38. click delete syntax button
<<Existing>>
39. send syntax data to delete
40. success message

5. Nonfunctional Requirements

5.1 Performance Requirements:

Code Conversion website requires a system with at least a stable browser with JavaScript enabled and 4 GB of RAM any graphics card available will be enough. Performance depends also on browser cache and hard disk speed.

5.2 Security Requirements:

Code Conversion website does not have any security requirements and thus any type of user can use it without any additional privileges.

5.3 Authentication System:

Login and register will not be implemented for normal users as any type of user can use the website as guest / anonymous without the need to authenticate himself

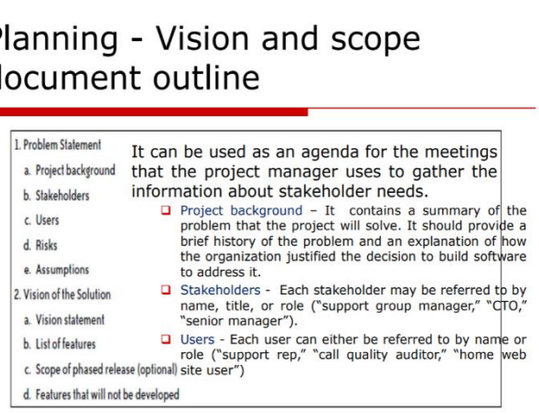
6. Conclusion

Converting code type of website system that aims to help anyone at any level of expertise in seconds for free with no cost at all so that everyone anywhere can start learning and enhancing his mindset and the way of thinking by exposing himself to the world of coding and programming in general

Vision and Scope Document

Code conversion

Lecture 2 Slide 9



1-Problem Statement

A- Project background:

Our web app is a friendly helper for people learning to code. It's free and easy to use with few clicks you can get started you can convert code from one language to another, find basic programming tutorials. This project began not more than a PowerPoint document and turned into a real web app that everyone can use for free we made this because many tools that do the same thing cost money with limited features and are not easy for everyone as they require some technical knowledge. Our goal is to help people at any skill level to get started and learn to code easily.

B- Stakeholders:

- Business owners
- Developers
- Product managers
- Team members
- Users
- Admins

C- Users:

they are anyone who will use the application whether a student, a developer or community

D- Risks:

- Accurately converting syntax between programming languages.
- Tutorials may be hard to follow for beginners
- Team members leaving the job

E- Assumptions:

- Programming Languages syntax will remain the same will not be changed any time

2- Vision of the solution

A- Vision statement:

The web application aims to be a user-friendly platform for people at all skill levels, offering syntax conversion between programming languages, easy access to programming tutorials while making sure it is free and easy to use and in someday it can be sponsored and adapted by major companies

B- List of features:

- Single syntax convert:

Users can choose the desired programming language and the desired control type to view the syntax for it

- Multi syntax convert:

Users can choose multiple programming languages and compare between their syntax in a specific control type

- Admin dashboard:

Admin will be able to login to perform CRUD operations for tutorials and programming languages.

- Programming tutorials:

Users can view categorized tutorials based on programming languages.

C- Features that will not be developed:

- Register / login will not be required for users, they can use website as guests
- Tutorials will not be in video format, will be available as text

Make a mitigation plan

The team can take any or all of these actions to mitigate a risk:

Alter the project plan - The project schedule can be adjusted to help reduce the risk.

Add additional tasks - There are certain actions that can be added to the schedule to help avoid risks. For example, if there is a high probability that a critical team member will leave the organization, cross-training tasks can be assigned to other people.

Sample risk plan

Risk plan for project	Call center application project			
Assessment team members	Mike, Barbara, Quentin, Jill, Sophie, Dean, Kyle			
Risk	**Prob.**	**Impact**	**Priority**	**Actions**
Senior management will move call center offshore, which will require an internationalization feature to be built	3	5	15	1. Mike will add a requirements task to the schedule for Quentin to begin investigating internationalization requirements. 2. If the call cener is moved, Mike will call a team meeting to review the schedule and Barbara will inform the rest of senior management of the potential delay.
Jim will be pulled off of this project for Royalty Archive project bug fixes	4	3	12	1. Assign Kyle to work with Jill on the initial programming tasks to make sure he is cross-trained. 2. If Jill is pulled off, she will spend 10% of her time reviewing this project with Kyle.
Reporting feature will be needed	2	4	8	If this happens, Mike will work with Sophie and Kyle to reestimate the programming tasks.
Additional time will be needed to gather requirements from potential users at Boston client	5	1	5	None
Will need to support tie-in to support additional database vendors	1	3	3	None

Risk Mitigation plan

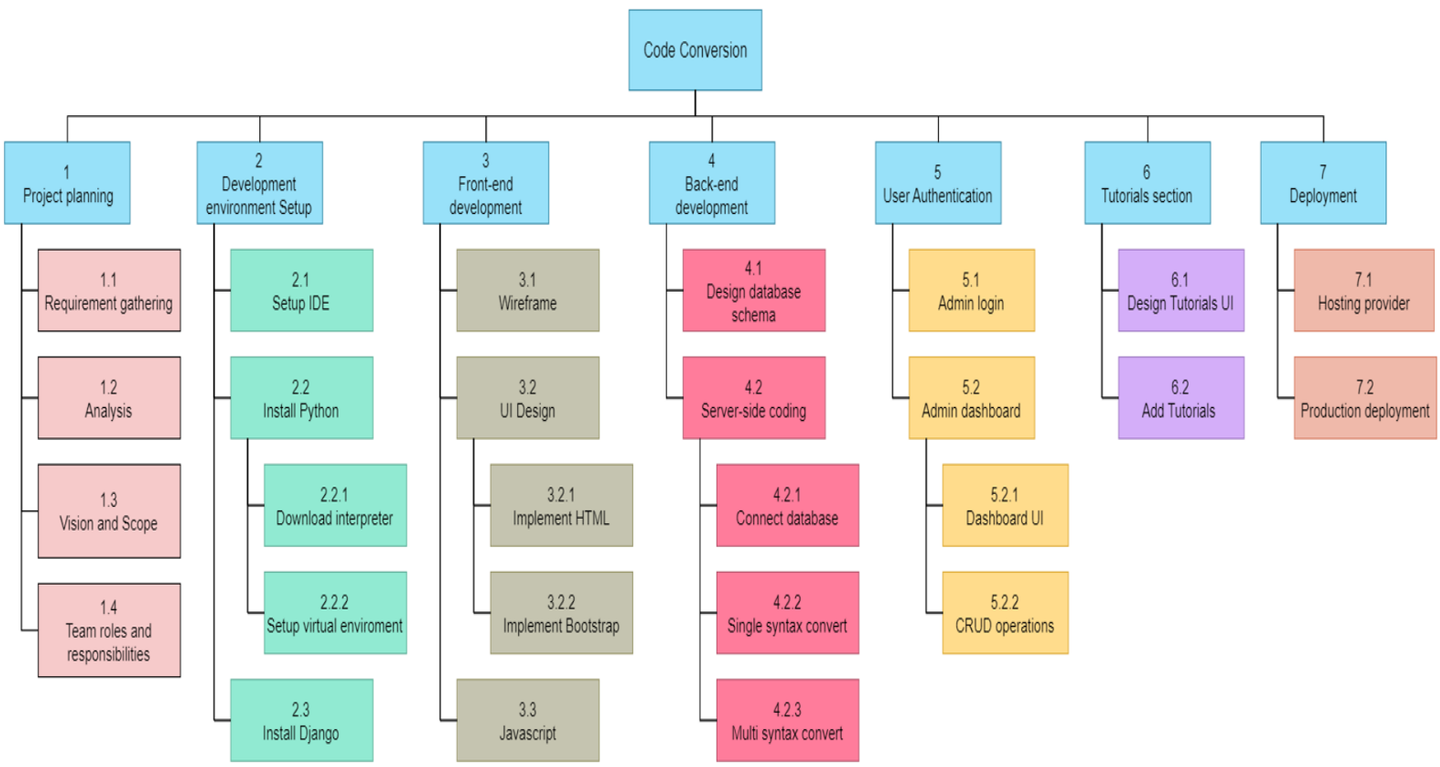
Lecture 2 Slide 16

Risk Plan for Project	Code Conversion Web Application			
Assessment Team Members	Ahmed Moataz, Naglaa Ahmed, Sameh Taha			
Risks	**Prob**	**Impact**	**Priority**	**Actions**
Developer may pass away during developing phase of the project.	3	5	15	-Good documentation -Assistant developer
Developer may get a better job offer.	5	5	25	-Good documentation -Assistant developer
Stakeholder may not complete the payment.	3	6	18	-Payment should be in instalments
The project exceeded the Database storage.	4	4	16	-Good planning -Scalability feature (horizontal up)
Web site performance is very slow.	2	3	6	-Always Monitoring -Support team
some programming languages don't have enough tutorials	1	3	3	- Using the popular and wide spread languages and not out-dated languages -Adding Articles as tutorials
Hacking the Web Site	2	7	14	-Daily backups -Good security team -Latest technology

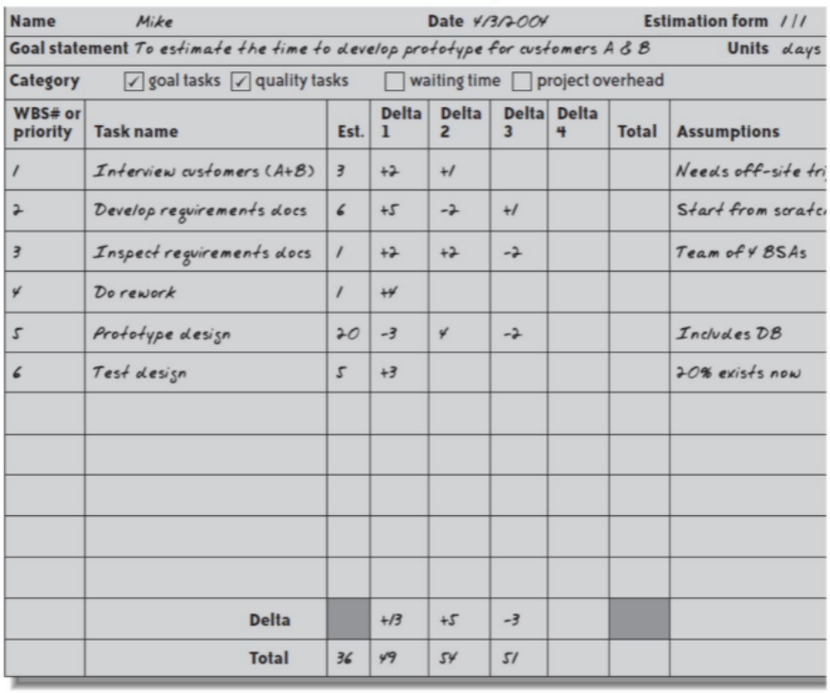
Elements of a Successful Estimate

- A sound estimate starts with a work breakdown structure (WBS).
- A WBS is a list of tasks that, if completed, will produce the final product.
- There are many ways to decompose a project into tasks. The project can be broken down by feature, by project phase (requirements tasks, design tasks, programming tasks, QA tasks, etc.), or by some combination of the two.
- Ideally, the WBS should reflect the way previous projects have been developed.
- A useful rule of thumb is that any project can be broken down into between 10 and 20 tasks. for small projects (like writing a simple calculator program), the tasks are small ("Build the arithmetic object that adds, multiplies, or divides two numbers")
- Once the WBS is created, the team must create an estimate of the effort required to perform each task. The most accurate estimates are those that rely on prior experience.

Work breakdown structure



Filled-in estimation form



Name	Mike			Date 4/3/2004			Estimation form 1/1
Goal statement	To estimate the time to develop prototype for customers A & B						Units days
Category	☑ goal tasks ☑ quality tasks ☐ waiting time ☐ project overhead						
WBS# or priority	Task name	Est.	Delta 1	Delta 2	Delta 3	Delta 4	Total	Assumptions
1	Interview customers (A+B)	3	+2	+1				Needs off-site tri
2	Develop requirements docs	6	+5	-2	+1			Start from scratc
3	Inspect requirements docs	1	+2	+2	-2			Team of 4 BSAs
4	Do rework	1	+4					
5	Prototype design	20	-3	4	-2			Includes DB
6	Test design	5	+3					20% exists now
	Delta		+13	+5	-3			
	Total	36	49	54	51			

lecture 3 slide 15

Name	Ahmed Moataz		**Date** 24/3/2024			**Estimation form** /1/
Goal statement	to estimate the time to develop prototype for code conversion				**Unit** days	
Category	☐ Goal Tasks	☐ Quality Tasks	☐ Waiting time	☐ Project overhead		
priority	Task name	est	Naglaa	Sameh	Total	Assumptions
1	Requirment gathering	3	+2	+1		
2	Single syntax convert	4	+1	-1		
3	Multi syntax covnert	3	+1	+2		
4	Admin dashboard	5	-2	+2		
5	User authentication	5	+2	+1		
6	Programming tutorials	2	+1	-1		
	Delta		+5	+4		
	Total	22	27	31		

Name Naglaa Ahmed	**Date** 24/3/2024	**Estimation form** /2/
Goal statement to estimate the time to develop prototype for code conversion		**Unit** days
Category ☐ Goal Tasks ☐ Quality Tasks ☐ Waiting time ☐ Project overhead		

priority	Task name	est	Ahmed	Sameh	Total	Assumptions
1	Requirment gathering	5	-2	-1		
2	Single syntax convert	5	-1	-2		
3	Multi syntax covnert	4	-1	+1		
4	Admin dashboard	3	+2	+4		
5	User authentication	7	-2	-1		
6	Programming tutorials	3	-1	-2		
	Delta		-5	-1		
	Total	27	22	21		

Name Sameh Taha	**Date** 24/3/2024	**Estimation form** /3/
Goal statement to estimate the time to develop prototype for code conversion		**Unit** days
Category ☐ Goal Tasks ☐ Quality Tasks ☐ Waiting time ☐ Project overhead		

priority	Task name	est	Ahmed	Naglaa	Total	Assumptions
1	Requirment gathering	4	-1	+1		
2	Single syntax convert	3	+1	+2		
3	Multi syntax covnert	5	-2	-1		
4	Admin dashboard	7	-2	-4		
5	User authentication	6	-1	+1		
6	Programming tutorials	1	+1	+2		
	Delta		-4	1		
	Total	26	22	23		

Basic course of event - Assemble tasks

the project manager should create a spreadsheet that lists the final estimates that each person came up with. The preadsheet should indicate the best case and worst-case scenarios, and it should indicate any place that further discussion will be required. Any task with an especially wide discrepancy should be marked for further discussion.

Summarized results of estimation

Goal statement To estimate the time to develop prototype for customers A & B

Estimators Mike, Quentin, Jill, Sophie **Units** days

Shaded items must be discussed

WBS# or priority	Task name	M.	Q.	J.	S.	Best-case	Worst-case	Avg.-hi & lo	Notes
1	Interview customers (A+B)	6	4	3	3	3	6	3.5	
2	Develop requirements docs	5	10	2	5	2	10	5	Discrepancy between Q. and J.
3	Inspect requirements docs	7	5	6	5	5	7	5.5	
4	Do rework	8	7	9	7	7	9	7.5	
5	Prototype design	28	23	31	25	23	31	26.5	
6	Test design	9	7	6	6	6	9	6.5	
	Total	63	56	57	51	46	72	54.5	

Then, Review results

Lecture 3 slide 18

Goal statement	to estimate the time to develop prototype for code conversion			**Unit** days			
Estimators	Ahmed Moataz, Naglaa Ahmed, Sameh Taha						
priority	Task name	A	N	S	Best-case	Worst-case	AVG	Notes
1	Requirment gathering	3	5	4	3	5	4	
2	Single syntax convert	4	5	3	3	5	4	
3	Multi syntax covnert	3	4	5	3	5	4	
4	Admin dashboard	5	3	7	3	7	5	
5	User authentication	5	7	6	5	7	6	
6	Programming tutorials	2	3	1	1	3	2	
	Total	22	27	26	18	32	25	

Cocomo Calculation

Lecture 5 slide 25

Intermediate - Classification of Cost Drivers and their attributes (15) – Cont.

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME	..	..	1.00	1.11	1.30	1.66
STOR	..	..	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

Cost Drivers	RATINGS					
	Very low	Low	Nominal	High	Very high	Extra high
Personnel Attributes						
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90	..	..
LEXP	1.14	1.07	1.00	0.95	..	..
Project Attributes						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	1.04	1.10	..

Effort Adjustment Factors (EAF)

Intermediate COCOMO Calculation

Choose Model: Organic ○ Semi Detached ○ Embedded ○ KLOC

Hardware attributes

Run-time performance constraints:

Memory constraints:

The volatility of the virtual machine environment:

Required turnabout time:

Product attributes

Required software reliability extent:

Size of the application database:

The complexity of the product:

Personnel attributes

Analyst capability:

Software engineering capability:

Applications experience:

Virtual machine experience:

Programming language experience:

Project attributes

Use of software tools:

Application of software engineering methods:

Required development schedule:

Calculate

Result:

KLOC=

EAF=

Effort=

Development Time=

persons required=

Productivity=

Basic COCOMO Calculation

KLOC

Project Type:

- Organic
- Semi Detached
- Embedded

Calculate

KLOC =

Effort =

Development Time =

Persons Required =

Productivity =