Steamware

All Plug-in's in one spot

Problem Statement:

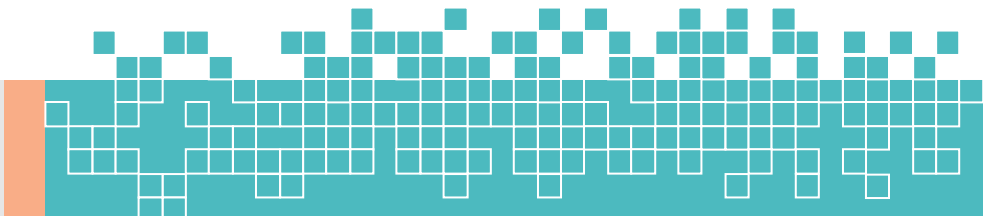
Coding is a part of every programming for the past 2 decades but a common problem every coder face is the bugs and invalidations in a code even when the coding is done at its best. So a small community of people researched to find a way where the coding and implementations can be done easily and more efficiently And this ultimately led to the origin of plug-in's.

These Plug-in's are highly sophisticated code fragments that are implemented into the coding environment using simple and easily recognizable text. Over the years these plugins have evolved and it has also helped programmers to add more features to the current programming language. Every programming language have more than 10 necessary plugins and 500 to 3,000 plugins depending on its popularity among programmers.

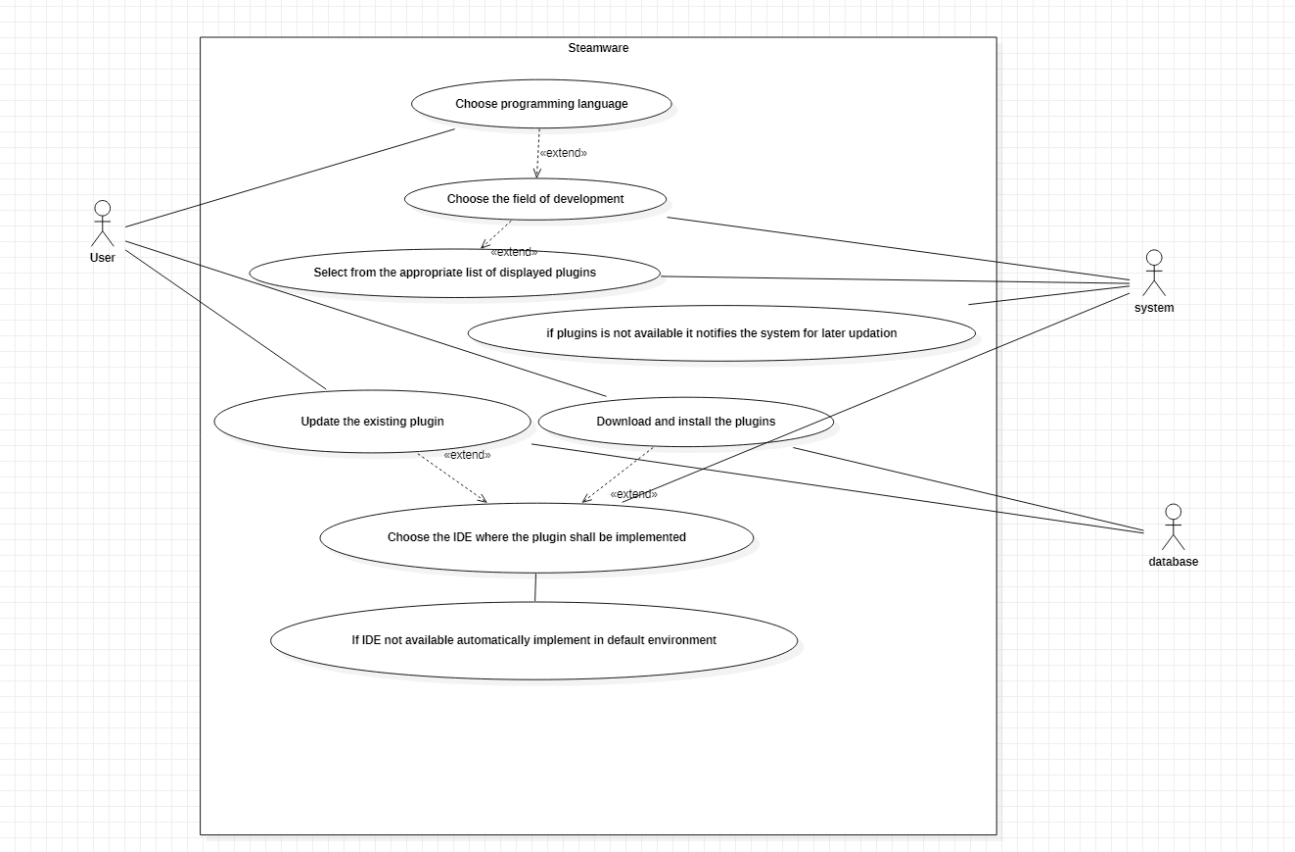
So, if a single programming language has 500 to 1,000 plugins. There are around 700 programming languages and in an average, there are around 5,776 plug-in's and around 300 universal plug-in's .

The next question is that where are these plugins and how can we add them to our compilers and code editors ?

All these plugins are distributed among the program dev. Web Sites and other geeky websites like GitHub, Stackoverflow, SourceForge, Gitea and many more .
So for a user to implement a program he need a proper plugin and finding the plugin in the web is going to waste a lot of time. A proper and universal solution should be implemented for this so that the overall coding and debugging part of a software can be made more efficiently at a short period of time.



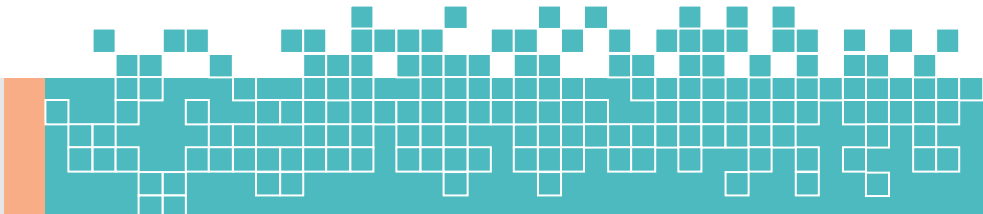
Use Case Diagram:



Software Requirements Specifications

STEAMWARE

USE CASE NAME	Use case name is Steamware. It fetches numerous libraries related to multiple programming languages and imports it to your working environment
BRIEF DESCRIPTION	Steamware is an open-source software which gives user access to thousands of libraries and plugins from available websites and databases depending on the work of field you decide to work on. It makes coding environment easy and user friendly. It accumulates data from common and universal websites like Github where software updation is constant. It also helps new coders to gather information and data regarding their snippet.



FLOW OF EVENTS

→ Open the application

→ Select your programming language

→ Select your field of work

→ Select from the appropriate list of displayed plugins

→ Download and install the plugins
→ Choose the IDE where the plugins shall be implemented.

BASIC FLOW

This use case starts when the use wishes to perform data analysis

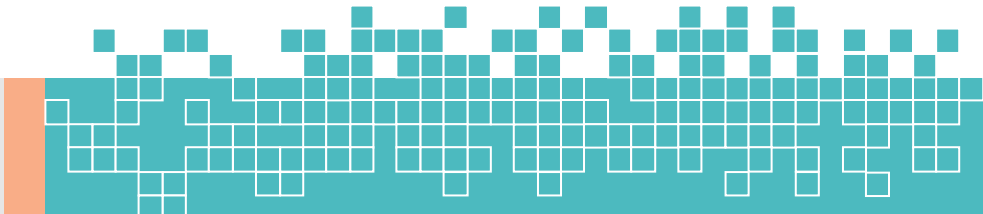
1. Select Python from the programming languages

2. Selects data analytics from the field of work

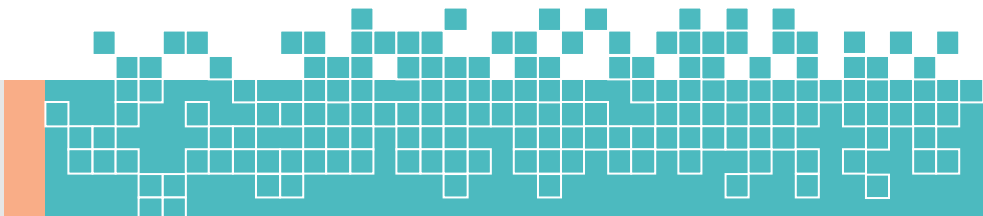
3. From the displayed plugins, matplot, pandas, numpy and seaborn are selected.

4. These libraries are installed from their respective websites

5. These are imported to VSCode with default import python lines



ALTERNATIVE FLOW	**Unavailable plugins/libraries** if in the Basic Flow the actor doesn't find the required library, then a request a pushed to the system to update their database with the necessary plugins. Once the plugin is available, it is notified to the user. **Unavailable IDE** If in the Basic Flow the system doesn't find the required IDE for the plugin to be implemented, then plugin is automatically pushed to default environment. (i.e.., Windows-Notepad)
PRE-CONDITION	IDE, OS, basic coding knowledge
POST CONDITION	If the use case was successful, the actor is now working with all the required plugins and a stable IDE.



EXTENSION POINTS

There are few extension points that can be used:

1) Plugins can also be imported from GitHub

2) Additional information for required plugins can be accessed from Stack Overflow

3) An option can be given to download a free IDE before adding the IDE to install the required plugins.