Exercise 1

1.
Code Completion , suggests methods, classes, variables.

Syntax Highlighting improves readability with color-coded keywords.

Debugging Tools , set breakpoints and step through code.

Project Management , easily manage multiple files and dependencies.

Error Detection , real-time error and warning highlighting.

Integrated Compiler/Interpreter , compile/run code directly in the IDE.

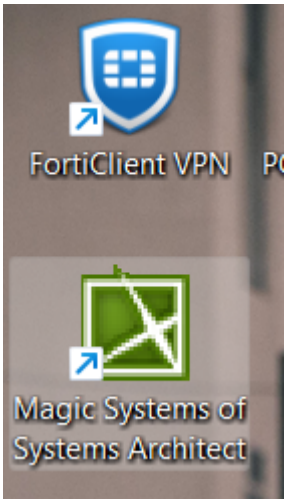
Version Control Integration

Refactoring Support , rename variables, extract methods safely.

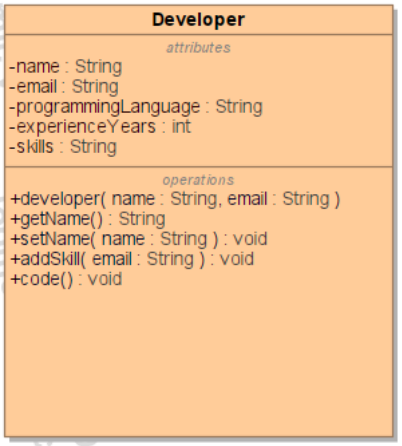
Plugin Support , extend functionality with tools

Built-in Terminal , execute shell commands within the IDE.

2.
installing magic systems of systems architect



5. class developer



6.

Software Design: The process of defining how the system will be built (architecture, components, interfaces).

Software Analysis: Focuses on understanding what the system should do (requirements, use cases, user stories).

Analysis = Problem understanding.

Design = Solution structuring.

7. Why software design is necessary

Ensures structure, reusability, scalability.

Helps manage complexity and reduce bugs.

Enables easier testing and maintenance.

Without design: Code becomes hard to maintain, error-prone, unscalable.

8. interdependent.

In a banking app, the UI design affects the component design (transferService), which in turn affects database design (transaction table) and architecture.
If the UI changes to support bulk transfers, all other designs may need changes too.