



# Di2Code

PROJECT PROPOSAL

# Motivation and Goal

As software engineers, we use Unified Modeling Language (UML) almost constantly.

Class diagram in (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

We generally draw this diagram by hand. Then spend quite some time to convert it to computer-made diagrams using any tool and then spend time to convert that specific diagram into a code using any specified programming language.

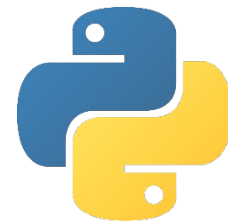
This is where our project comes to rescue, **Our Goal** is to help software engineers save time and effort by doing all the hard work for them. All by capturing a photo of their hand-written class diagram , we will be able to draw it and convert it to code. Sounds magical, right?

## Implementation

- **Preprocessing**: transform image into binary image, noise removal detect edges,
- **Segmentation**: breaking up image into basic component.(arrows, triangles, rectangles,...)
- **Detection** of the component(class names, attributes, methods)
- **Draw** the UML.
- **Convert** the detected UML Class Diagram to classes implemented in code.
- **Implement** web application.

# Implementation Considerations

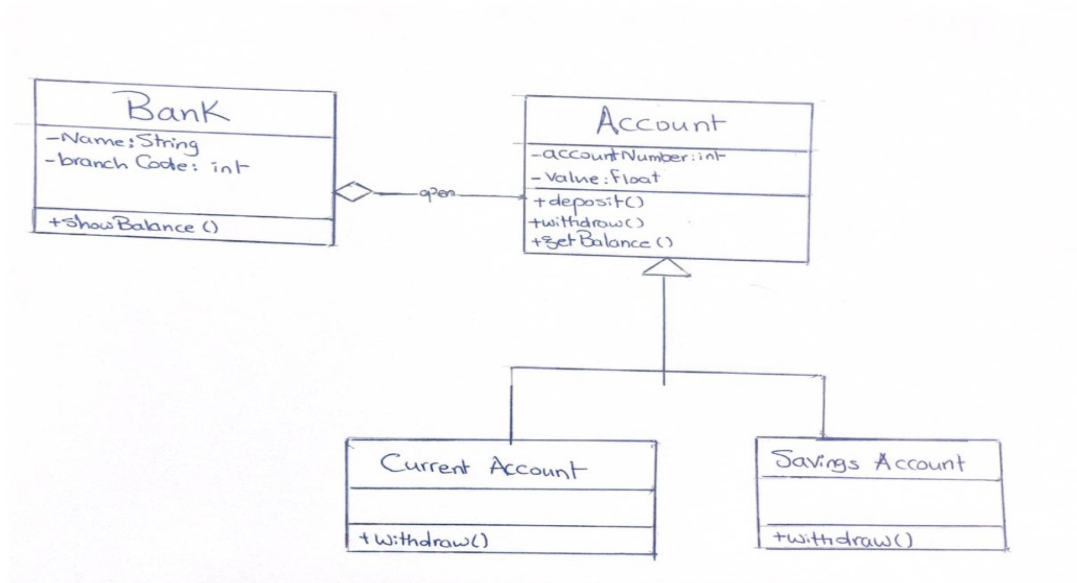
- Project will be implemented using OpenCV and python.
- The output code **could** be in many languages (c++, python, Java,...)
- The GUI will be a web application.



## Input and Output Sample

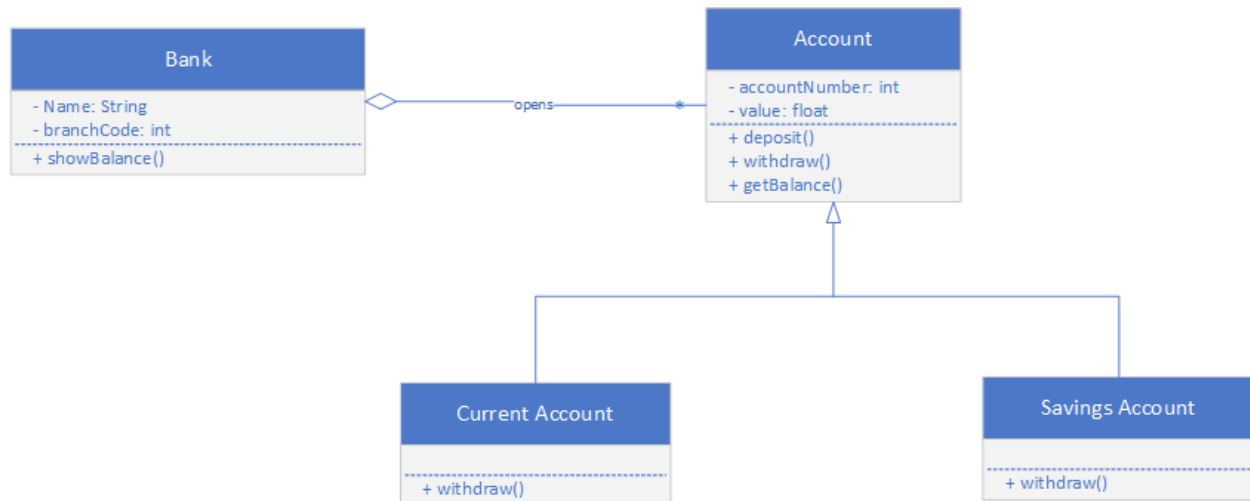
Input:

- Hand-written class diagram image



**Output:**

- Class diagram drawn.



- A file that has the converted code (.py or .cpp or .....).

```

// SavingsAccount class definition
class SavingsAccount : public Account {
public:
    // LIFECYCLE
    /** Default + Overloaded constructor.
    */
    SavingsAccount(int = 0, float = 0.0);

    // Use compiler-generated copy constructor, assignment, and destructor.
    // SavingsAccount(const SavingsAccount&);
    // SavingsAccount& operator=(const SavingsAccount&);
    // ~SavingsAccount();

    // OPERATIONS
    /** Overriding function that withdraws amount from SavingsAccount.
    *
    * @param aAmount The amount to be withdrawn.
    *
    * @return void
    */
    void Withdraw(float aAmount = 0);
};

// end class SavingsAccount
#endif
  
```

```

}class Bank {
public:
    // LIFECYCLE
    /** Default + Overloaded constructor.
    */
    Bank(const string& = "", int = 0, Account* = NULL);

    // Use compiler-generated copy constructor, assignment, and destructor.
    // Bank(const Bank&);
    // Bank& operator=(const Bank&);
    // ~Bank();

    // OPERATIONS
    /** function that shows balance.
    *
    * @param void
    *
    * @return void
    */
    void ShowBalance()const;

    // ACCESS
    // setters
    void SetName(const string& = "");
    void SetBranchCode(int = 0);
    void SetAccount(Account* = NULL);
  
```

## Team Members

Each team member will be responsible for one the main parts of the project.

**Team member names:**

<b>Dalia Ayman Ameen</b>	<b>1600521</b>
--------------------------	----------------

<b>Doaa Yehia Sayed Ahmed</b>	<b>1600525</b>
-------------------------------	----------------

<b>Zeinab Ismail Hefny</b>	<b>1600613</b>
----------------------------	----------------

<b>Sara Raafat Muhammed</b>	<b>1600623</b>
-----------------------------	----------------