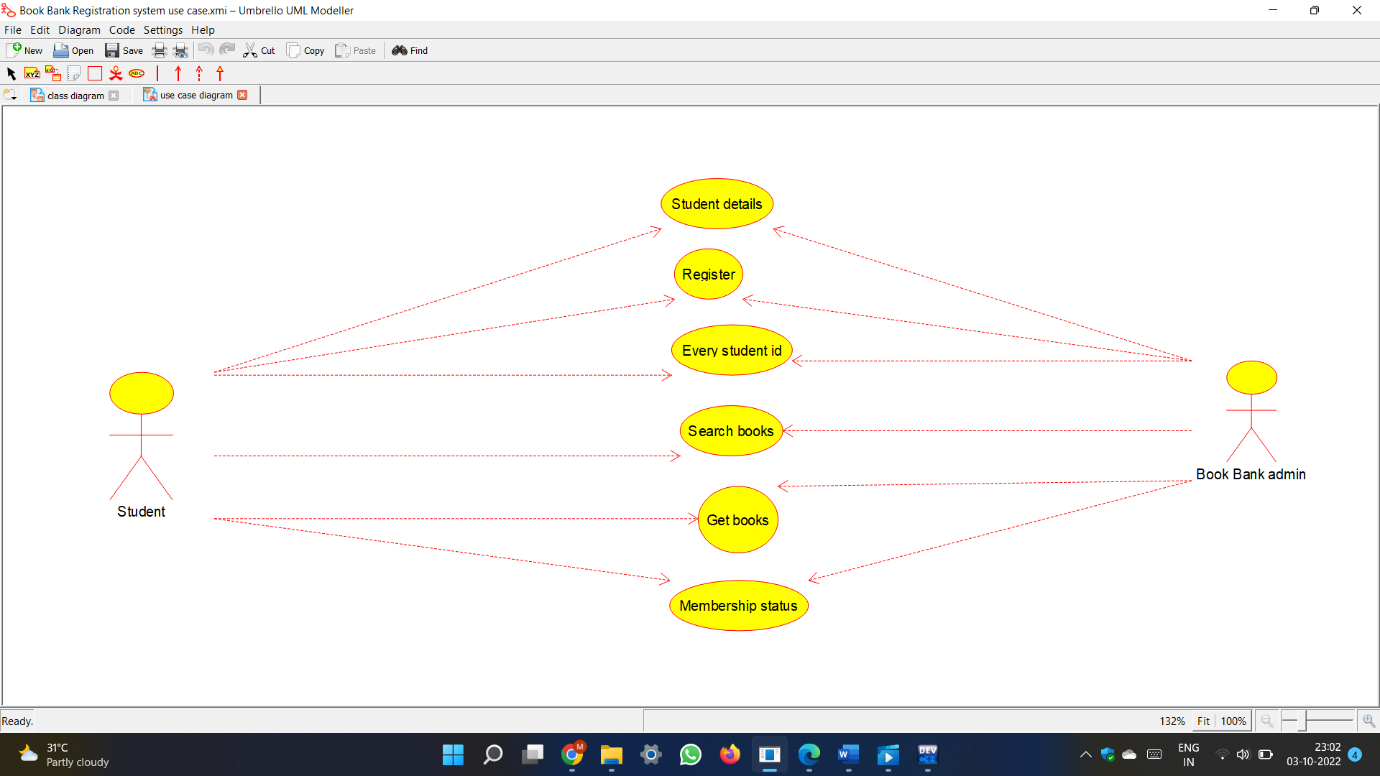
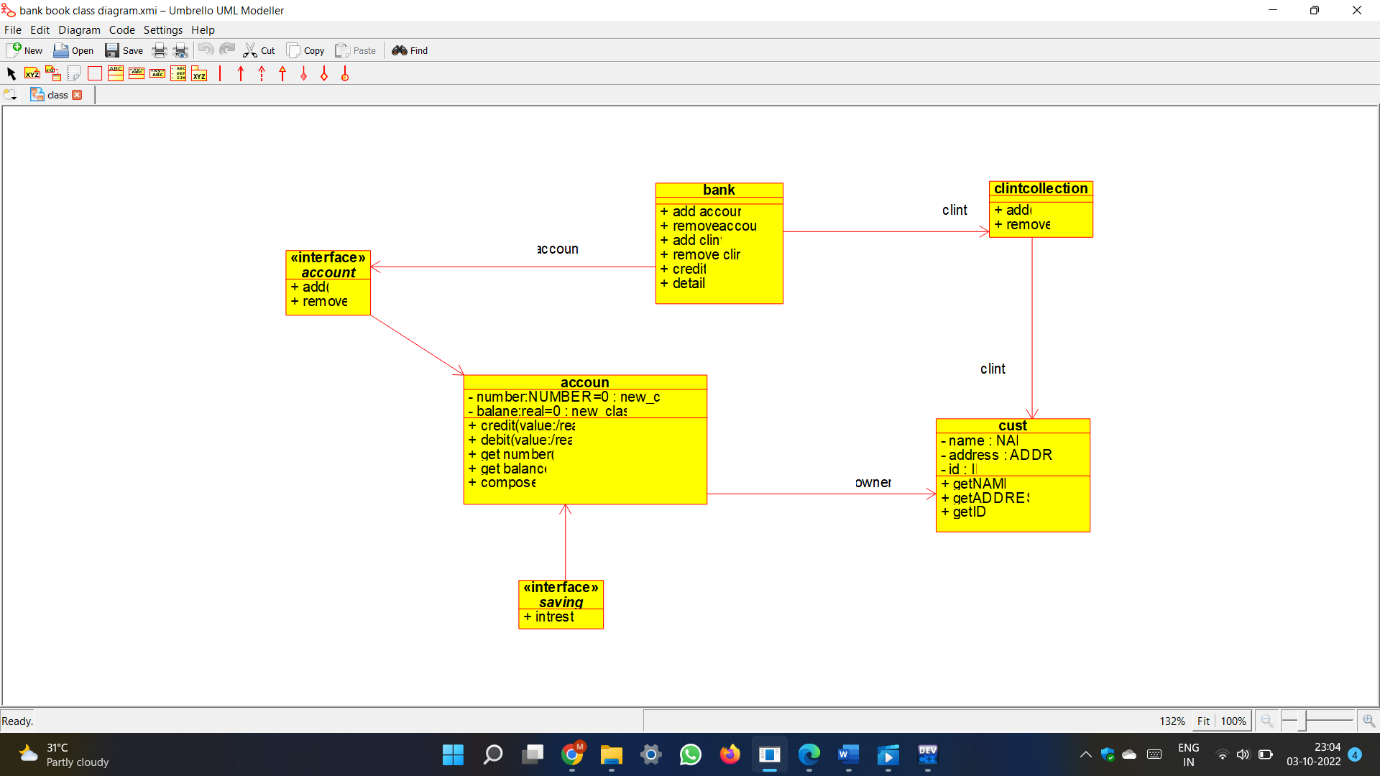
Book Bank Registration

Use case :



Class diagram :



Code :

#ifndef SHIPPING\_AGENT\_H

#define SHIPPING\_AGENT\_H

#include <string>

/\*\*

\* class shipping\_agent

\*

\*/

class shipping\_agent

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

shipping\_agent ();

/\*\*

\* Empty Destructor

\*/

virtual ~shipping\_agent ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

protected:

// Static Protected attributes

//

// Protected attributes

//

public:

// Protected attribute accessor methods

//

protected:

public:

// Protected attribute accessor methods

//

protected:

private:

// Static Private attributes

//

// Private attributes

//

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

private:

};

#endif // SHIPPING\_AGENT\_H

#ifndef SAVING\_H

#define SAVING\_H

#include <string>

/\*\*

\* class saving

\*

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Abstract Class \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

saving does not have any pure virtual methods, but its author

defined it as an abstract class, so you should not use it directly.

Inherit from it instead and create only objects from the derived classes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

class saving

{

public:

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

virtual void intrest ()

{

}

protected:

public:

// Protected attribute accessor methods

//

protected:

public:

// Protected attribute accessor methods

//

protected:

private:

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

private:

};

#endif // SAVING\_H

#ifndef ACCOUN\_H

#define ACCOUN\_H

#include <string>

/\*\*

\* class accoun

\*

\*/

class accoun

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

accoun ();

/\*\*

\* Empty Destructor

\*/

virtual ~accoun ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void credit\_value\_real\_ ()

{

}

/\*\*

\*/

void debit\_value\_real\_ ()

{

}

/\*\*

\*/

void get\_number\_ ()

{

}

/\*\*

\*/

void get\_balance ()

{

}

/\*\*

\*/

void compose ()

{

}

protected:

// Static Protected attributes

//

// Protected attributes

//

public:

// Protected attribute accessor methods

//

protected:

public:

// Protected attribute accessor methods

//

protected:

private:

// Static Private attributes

//

// Private attributes

//

new\_class\_4 number\_NUMBER\_0;

new\_class\_5 balane\_real\_0;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of number\_NUMBER\_0

\* @param new\_var the new value of number\_NUMBER\_0

\*/

void setNumber\_NUMBER\_0 (new\_class\_4 new\_var) {

number\_NUMBER\_0 = new\_var;

}

/\*\*

\* Get the value of number\_NUMBER\_0

\* @return the value of number\_NUMBER\_0

\*/

new\_class\_4 getNumber\_NUMBER\_0 () {

return number\_NUMBER\_0;

}

/\*\*

\* Set the value of balane\_real\_0

\* @param new\_var the new value of balane\_real\_0

\*/

void setBalane\_real\_0 (new\_class\_5 new\_var) {

balane\_real\_0 = new\_var;

}

/\*\*

\* Get the value of balane\_real\_0

\* @return the value of balane\_real\_0

\*/

new\_class\_5 getBalane\_real\_0 () {

return balane\_real\_0;

}

private:

void initAttributes () ;

};

#endif // ACCOUN\_H

#ifndef ADDRESS\_H

#define ADDRESS\_H

#include <string>

/\*\*

\* class ADDRESS

\*

\*/

class ADDRESS

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

ADDRESS ();

/\*\*

\* Empty Destructor

\*/

virtual ~ADDRESS ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

protected:

// Static Protected attributes

//

// Protected attributes

//

public:

// Protected attribute accessor methods

//

protected:

public:

// Protected attribute accessor methods

//

protected:

private:

// Static Private attributes

//

// Private attributes

//

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

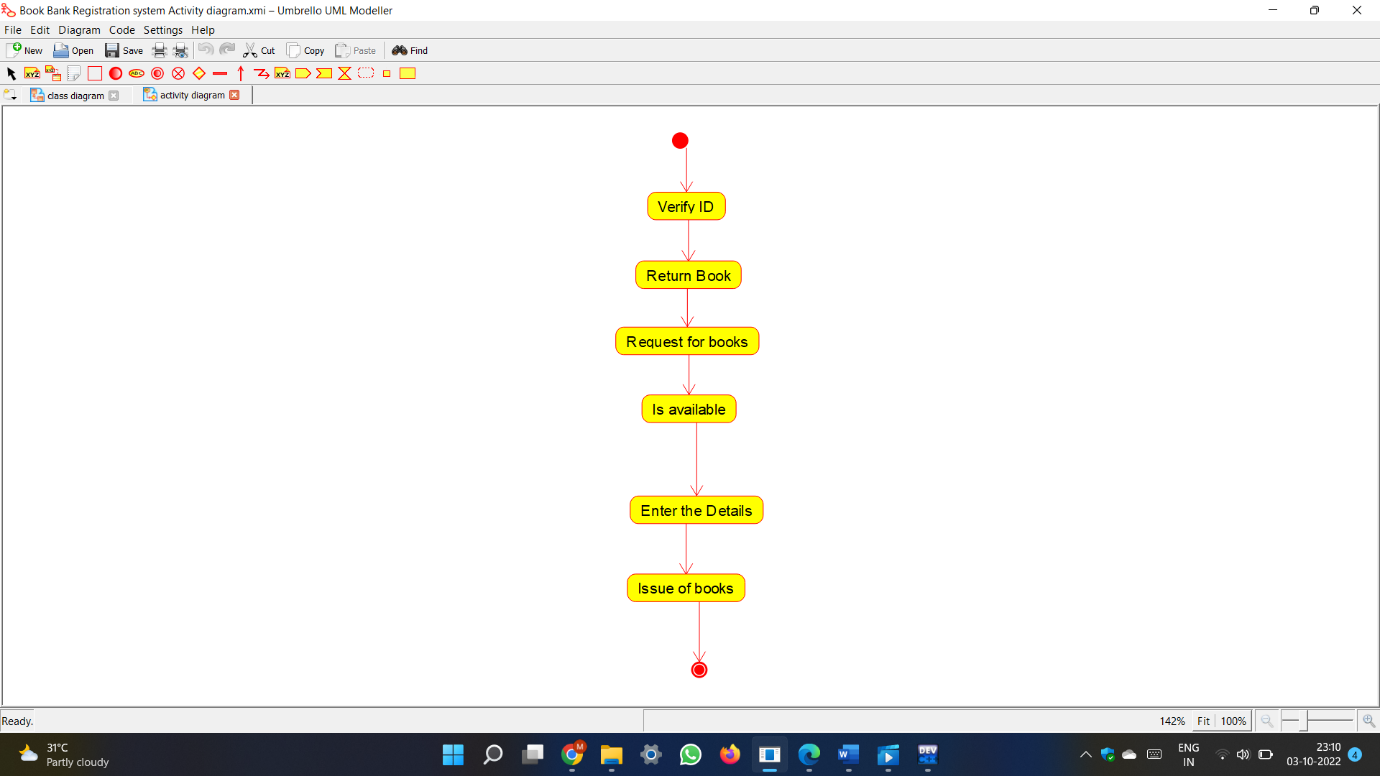
//

private:

};

#endif // ADDRESS\_H

Activity diagram :



State chart diagram :

