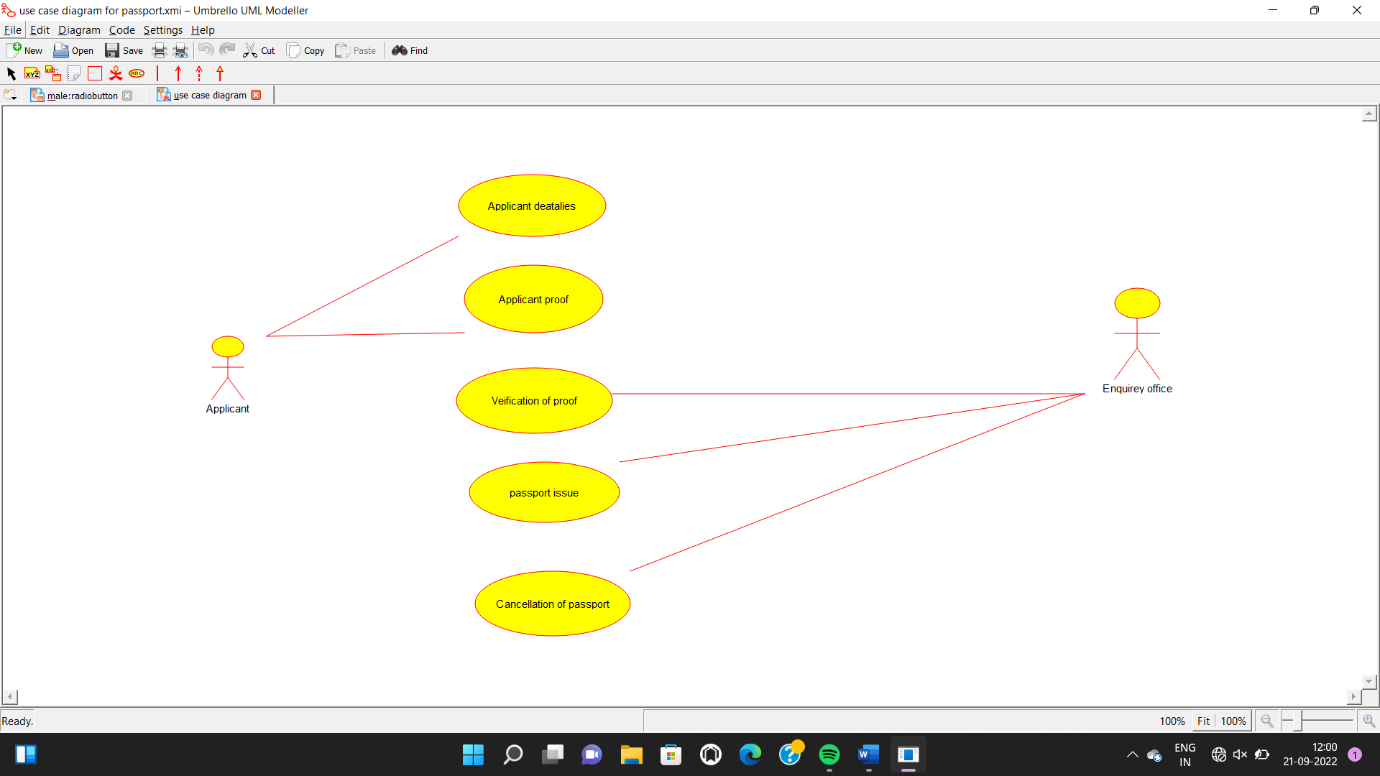
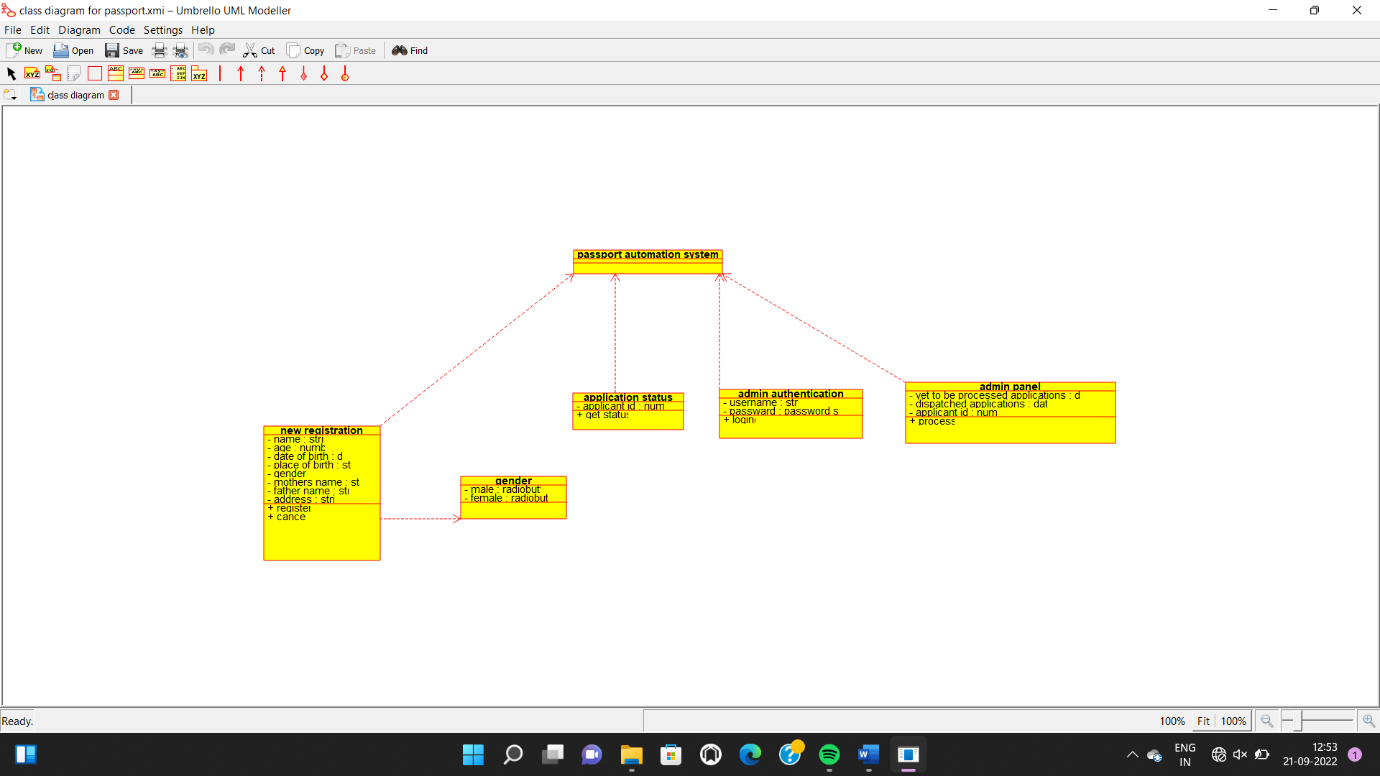
1.passport system

Usecase diagram



Class diagram



Source code

"new\_registration.h"#include

// Constructors/Destructors

//

new\_registration::new\_registration () {

initAttributes();

}

new\_registration::~new\_registration () { }

//

// Methods

//

// Accessor methods

//

// Other methods

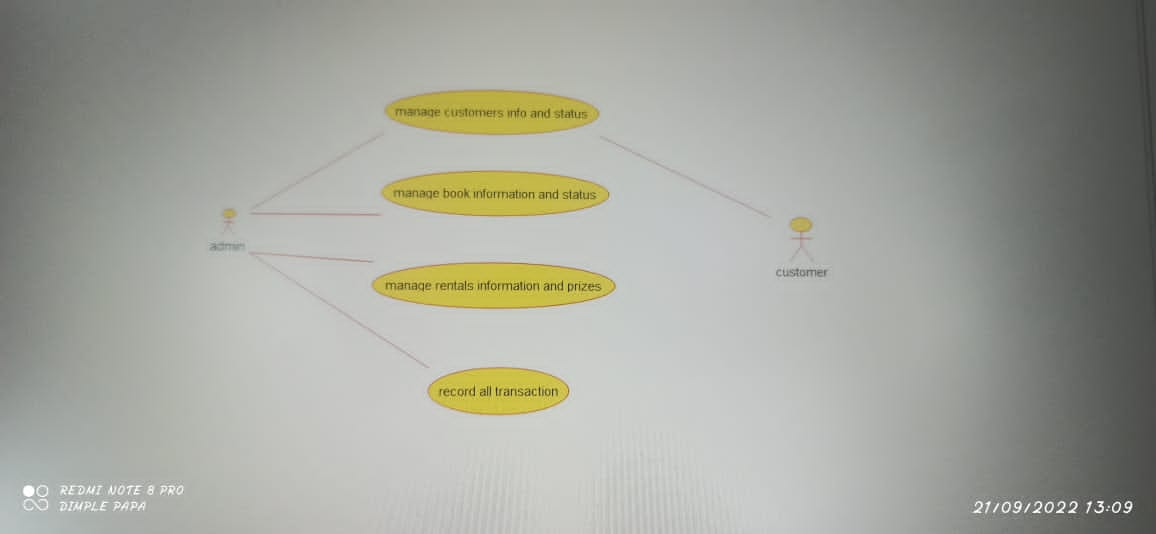
//

void new\_registration::initAttributes ()

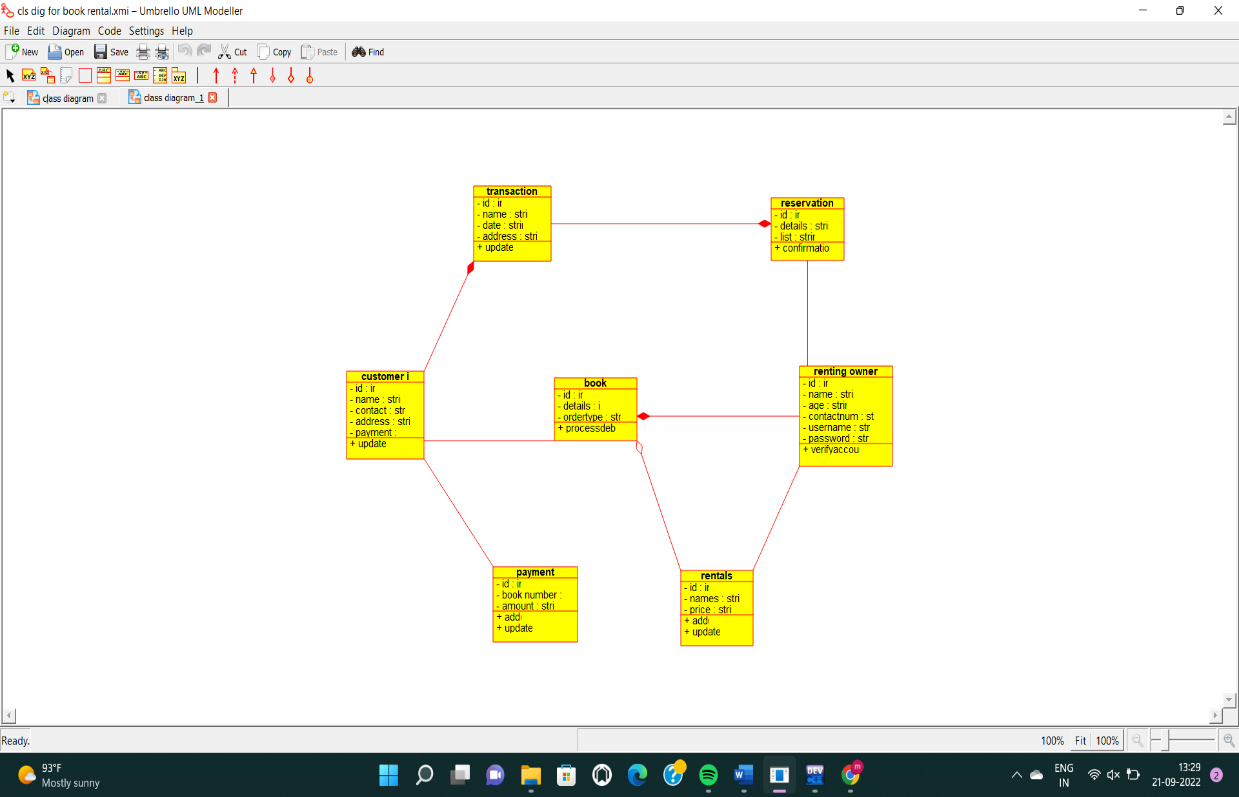
{

2.Book rental system

Use case diagram



Class diagram



Source code

"book.h"

e#includ

// Constructors/Destructors

//

book::book () {

initAttributes();

}

book::~book () { }

//

// Methods

//

// Accessor methods

//

// Other methods

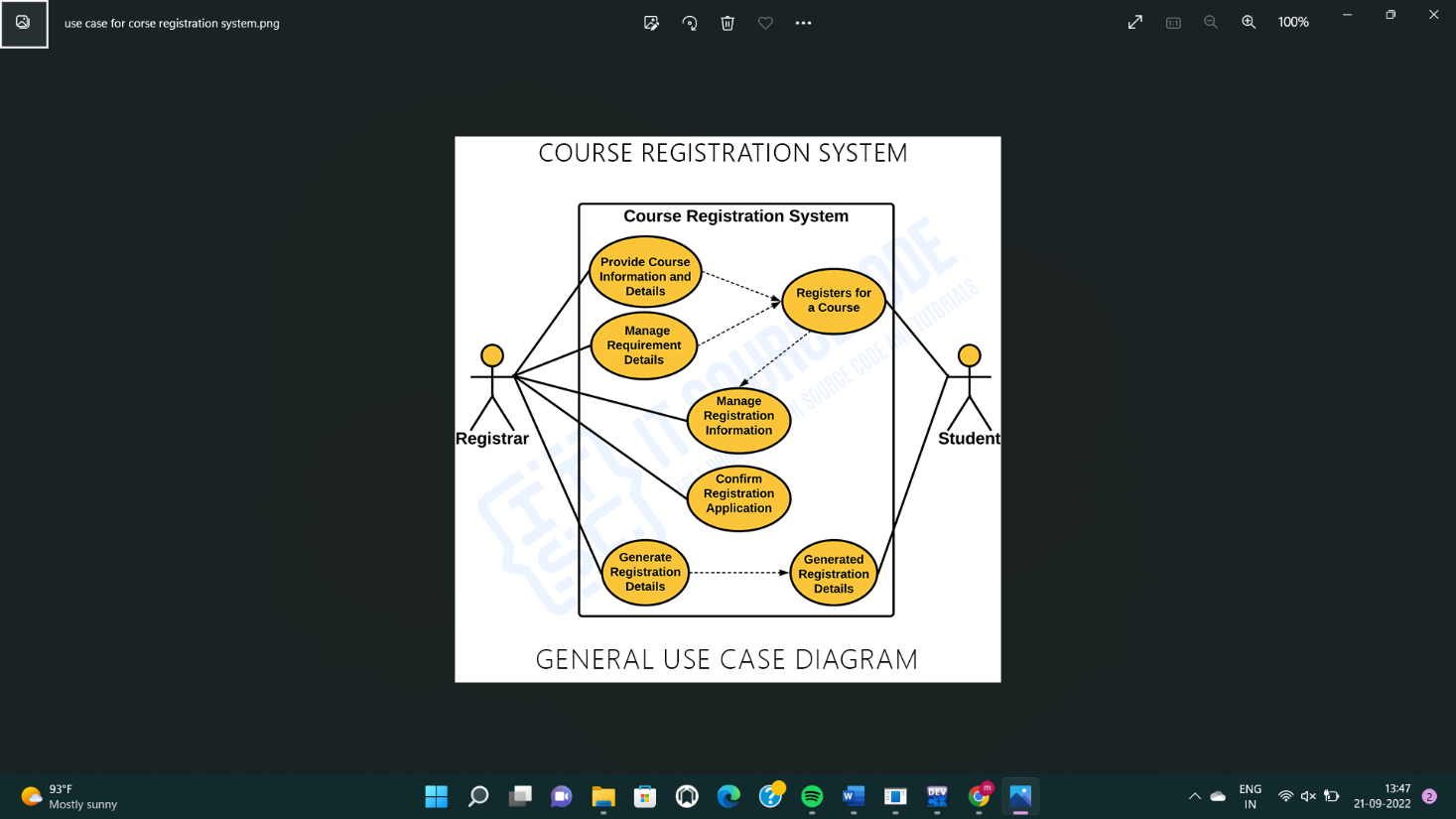
//

void book::initAttributes () {

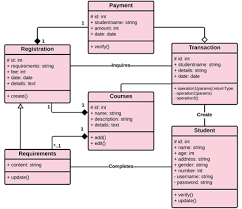
}

3.course registration system

Use case diagram



Class diagram



Source code

#ifndef ACCOUNT\_H

#define ACCOUNT\_H

#include <string>

/\*\*

\* class account

\*

\*/

class account

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

account ();

/\*\*

\* Empty Destructor

\*/

virtual ~account ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void createaccount ()

{

}

/\*\*

\*/

void confirmaccount ()

{

}

protected:

// Static Protected attributes

//

// Protected attributes

//

public:

// Protected attribute accessor methods

//

protected:

public:

// Protected attribute accessor methods

//

protected:

private:

// Static Private attributes

//

// Private attributes

//

string account\_id;

string type;

string student\_id;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of account\_id

\* @param new\_var the new value of account\_id

\*/

void setAccount\_id (string new\_var) {

account\_id = new\_var;

}

/\*\*

\* Get the value of account\_id

\* @return the value of account\_id

\*/

string getAccount\_id () {

return account\_id;

}

/\*\*

\* Set the value of type

\* @param new\_var the new value of type

\*/

void setType (string new\_var) {

type = new\_var;

}

/\*\*

\* Get the value of type

\* @return the value of type

\*/

string getType () {

return type;

}

/\*\*

\* Set the value of student\_id

\* @param new\_var the new value of student\_id

\*/

void setStudent\_id (string new\_var) {

student\_id = new\_var;

}

/\*\*

\* Get the value of student\_id

\* @return the value of student\_id

\*/

string getStudent\_id () {

return student\_id;

}

private:

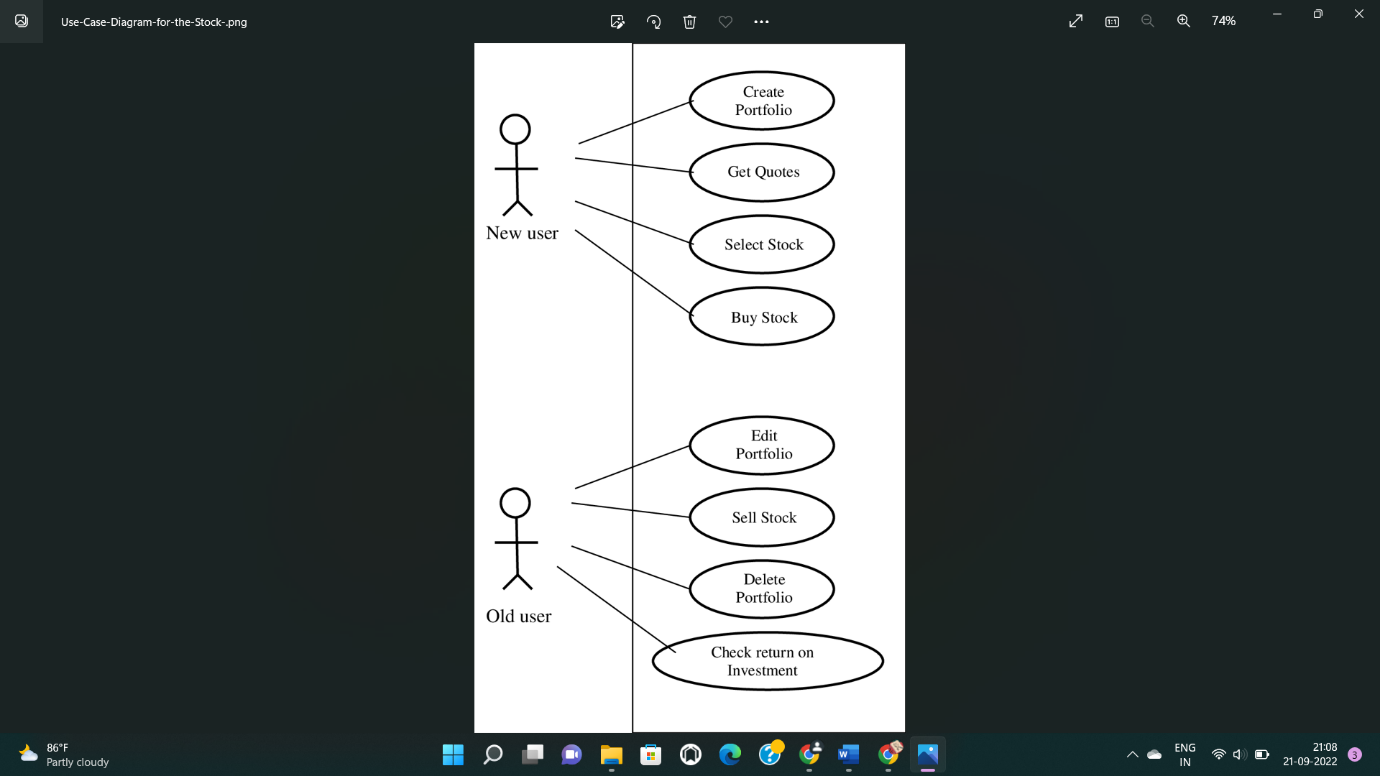
void initAttributes () ;

};

#endif // ACCOUNT\_H

4.stock analysis system

Use case diagram



Class diagram

[9:18 pm, 21/09/2022] sneha Ooad: #include "user.h"

[9:20 pm, 21/09/2022] sneha Ooad: // Constructors/Destructors

//

user::user () {

initAttributes();

}

user::~user () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void user::initAttributes () {

}

Stock

#include "stock.h"

// Constructors/Destructors

//

stock::stock () {

initAttributes();

}

stock::~stock () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void stock::initAttributes () {

}

Product

#include "product.h"

// Constructors/Destructors

//

product::product () {

initAttributes();

}

product::~product () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void product::initAttributes () {

}

Store

#include "store.h"

// Constructors/Destructors

//

store::store () {

initAttributes();

}

store::~store () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void store::initAttributes () {

}

Bill

#include "bill.h"

// Constructors/Destructors

//

bill::bill () {

initAttributes();

}

bill::~bill () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void bill::initAttributes () {

}

Permission

#include "permission.h"

// Constructors/Destructors

//

permission::permission () {

initAttributes();

}

permission::~permission () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void permission::initAttributes () {

}

Customer

#include "customer.h"

// Constructors/Destructors

//

customer::customer () {

initAttributes();

}

customer::~customer () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void customer::initAttributes () {

}

[9:20 pm, 21/09/2022] sneha Ooad: 4

[9:21 pm, 21/09/2022] sneha Ooad: SOURCE CODE:

Admin

#include "admin.h"

// Constructors/Destructors

//

admin::admin () {

initAttributes();

}

admin::~admin () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void admin::initAttributes () {

}

Course class

#include "course\_class.h"

// Constructors/Destructors

//

course\_class::course\_class () {

initAttributes();

}

course\_class::~course\_class () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void course\_class::initAttributes () {

}

Registration class

#include "registration\_class.h"

// Constructors/Destructors

//

registration\_class::registration\_class () {

initAttributes();

}

registration\_class::~registration\_class () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void registration\_class::initAttributes () {

}

[9:21 pm, 21/09/2022] sneha Ooad: Staff

#include "staff.h"

// Constructors/Destructors

//

staff::staff () {

initAttributes();

}

staff::~staff () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void staff::initAttributes () {

}

Student

#include "student.h"

// Constructors/Destructors

//

student::student () {

initAttributes();

}5

student::~student () { }

//

// Methods

//

// Accessor methods

//

// Other methods

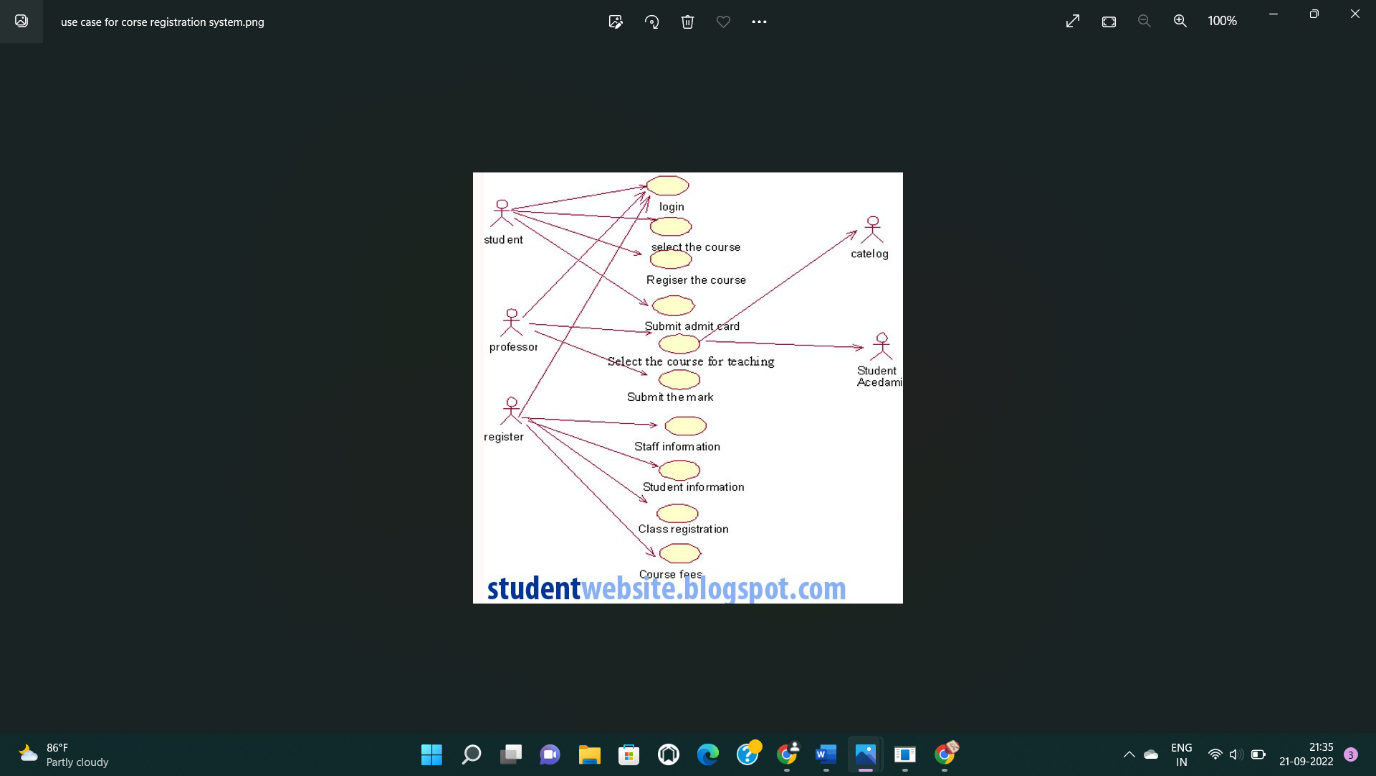
//

void student::initAttributes () {

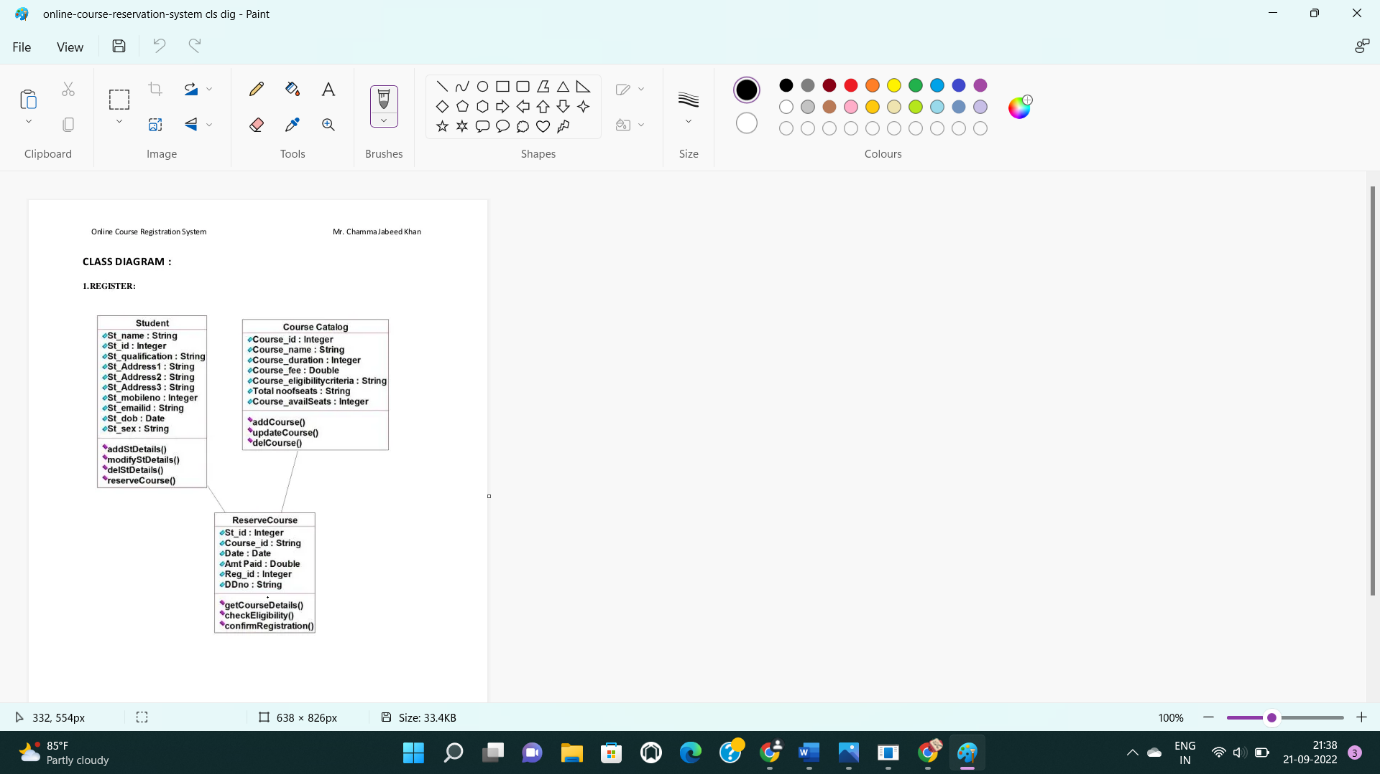
}

5.online course registration

Use diagram



Class diagram



Source code

[9:21 pm, 21/09/2022] sneha Ooad: Admin

#include "admin.h"

// Constructors/Destructors

//

admin::admin () {

initAttributes();

}

admin::~admin () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void admin::initAttributes () {

}

Course class

#include "course\_class.h"

// Constructors/Destructors

//

course\_class::course\_class () {

initAttributes();

}

course\_class::~course\_class () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void course\_class::initAttributes () {

}

Registration class

#include "registration\_class.h"

// Constructors/Destructors

//

registration\_class::registration\_class () {

initAttributes();

}

registration\_class::~registration\_class () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void registration\_class::initAttributes () {

}

[9:21 pm, 21/09/2022] sneha Ooad: Staff

#include "staff.h"

// Constructors/Destructors

//

staff::staff () {

initAttributes();

}

staff::~staff () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void staff::initAttributes () {

}

Student

#include "student.h"

// Constructors/Destructors

//

student::student () {

initAttributes();

}5

student::~student () { }

//

// Methods

//

// Accessor methods

//

// Other methods

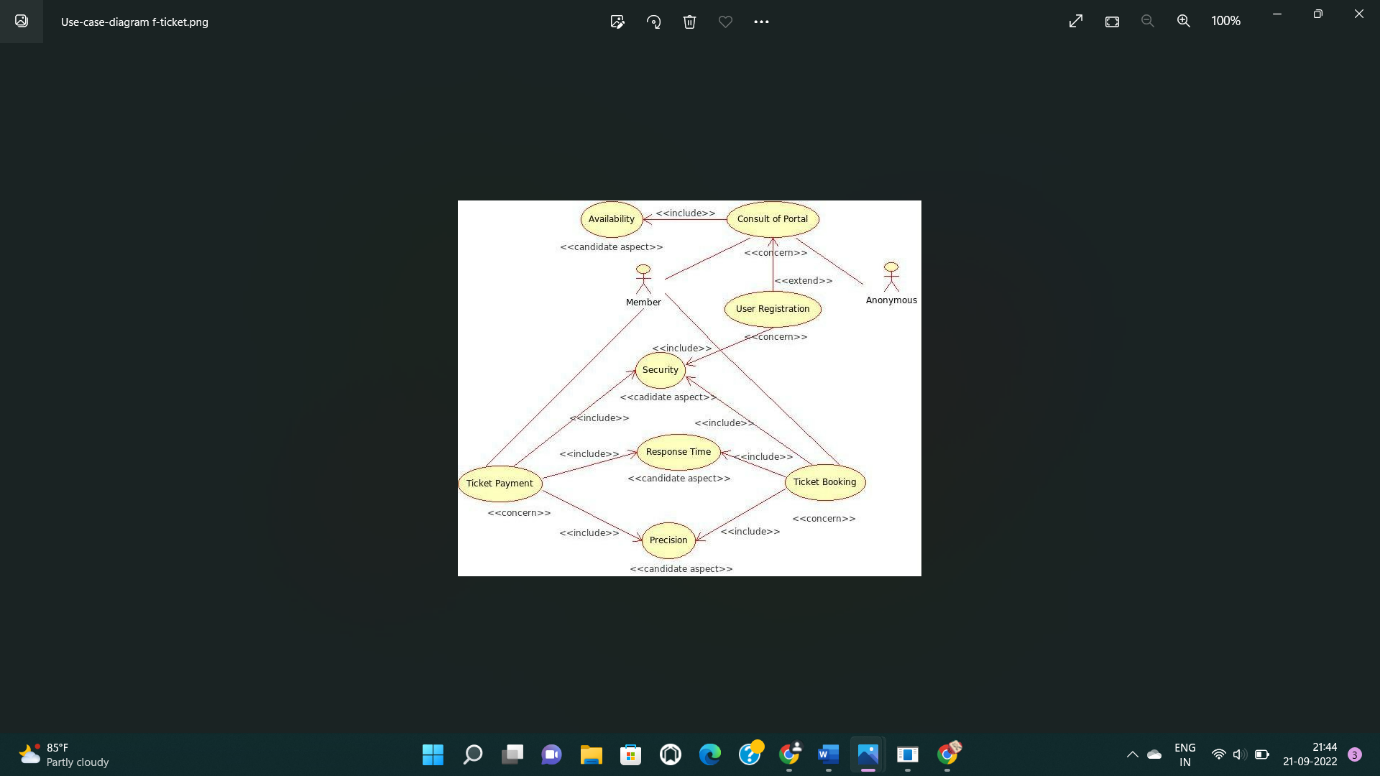
//

void student::initAttributes () {

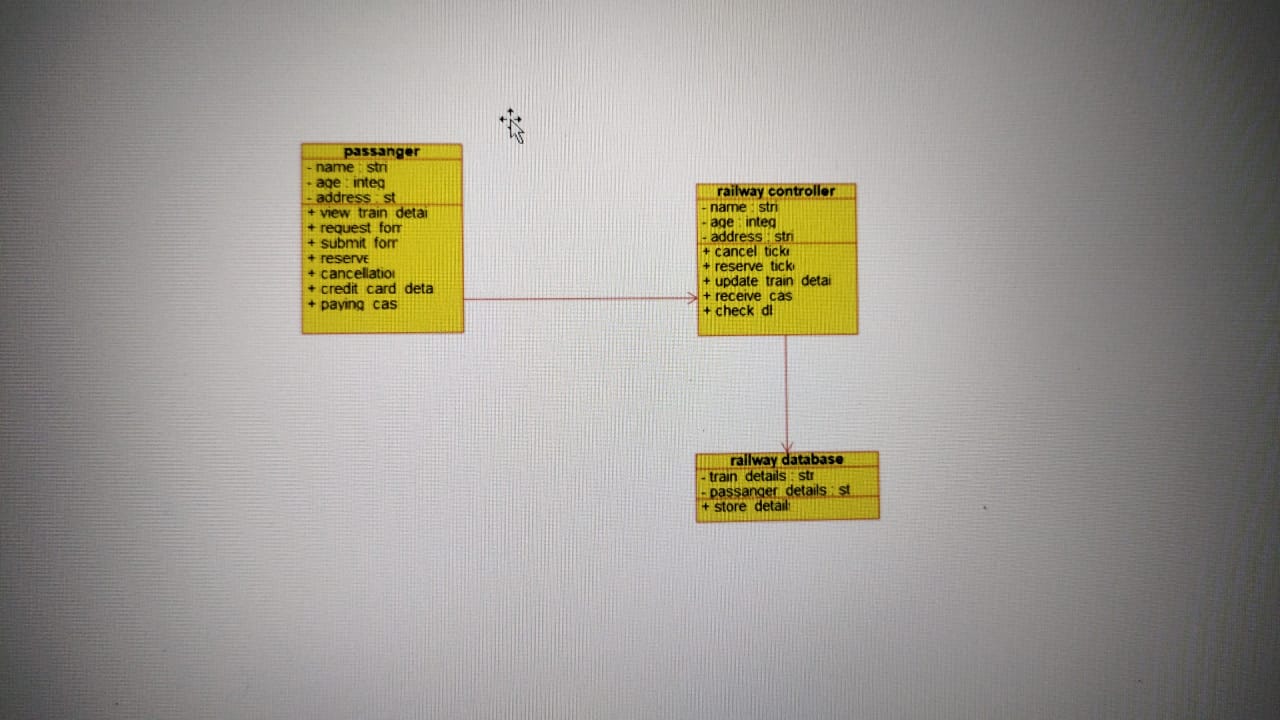
}

6.f-ticketing

Use cse diagram



Class diagram



Sourse code

#ifndef INTEGER\_H

#define INTEGER\_H

#include <string>

/\*\*

\* class integer

\*

\*/

class integer

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

integer ();

/\*\*

\* Empty Destructor

\*/

virtual ~integer ();

// 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

//

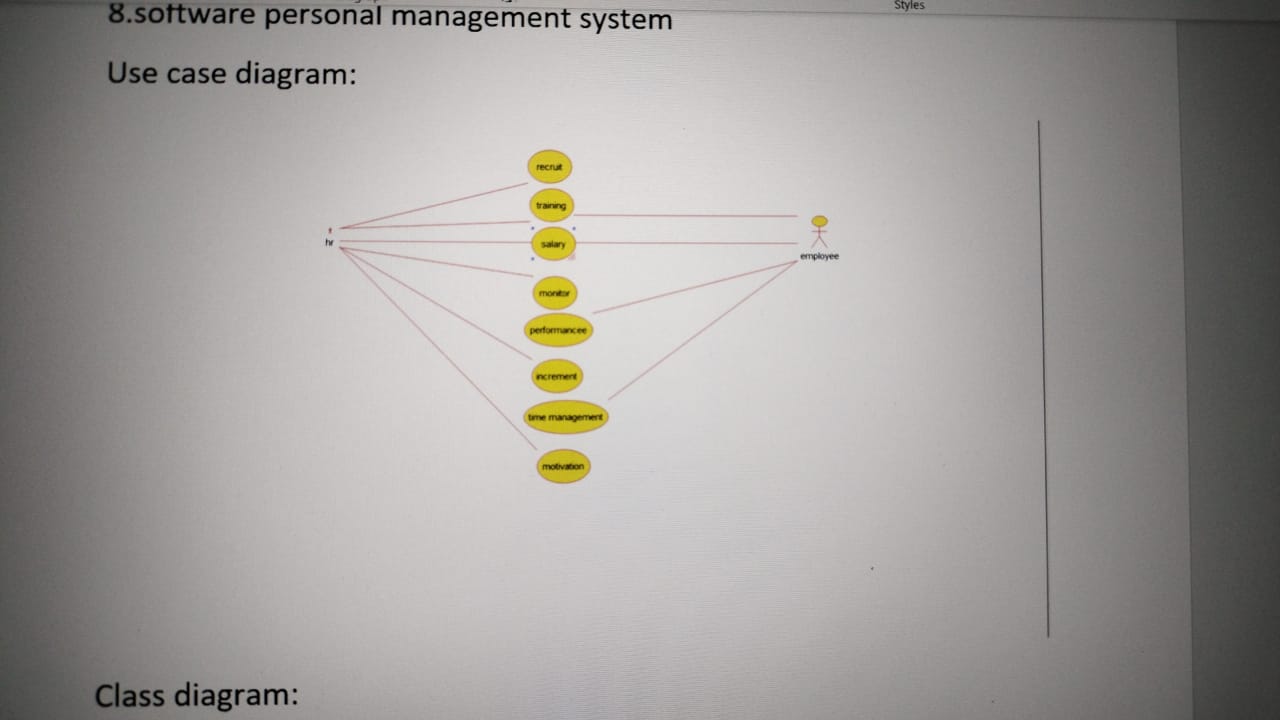
private:

};

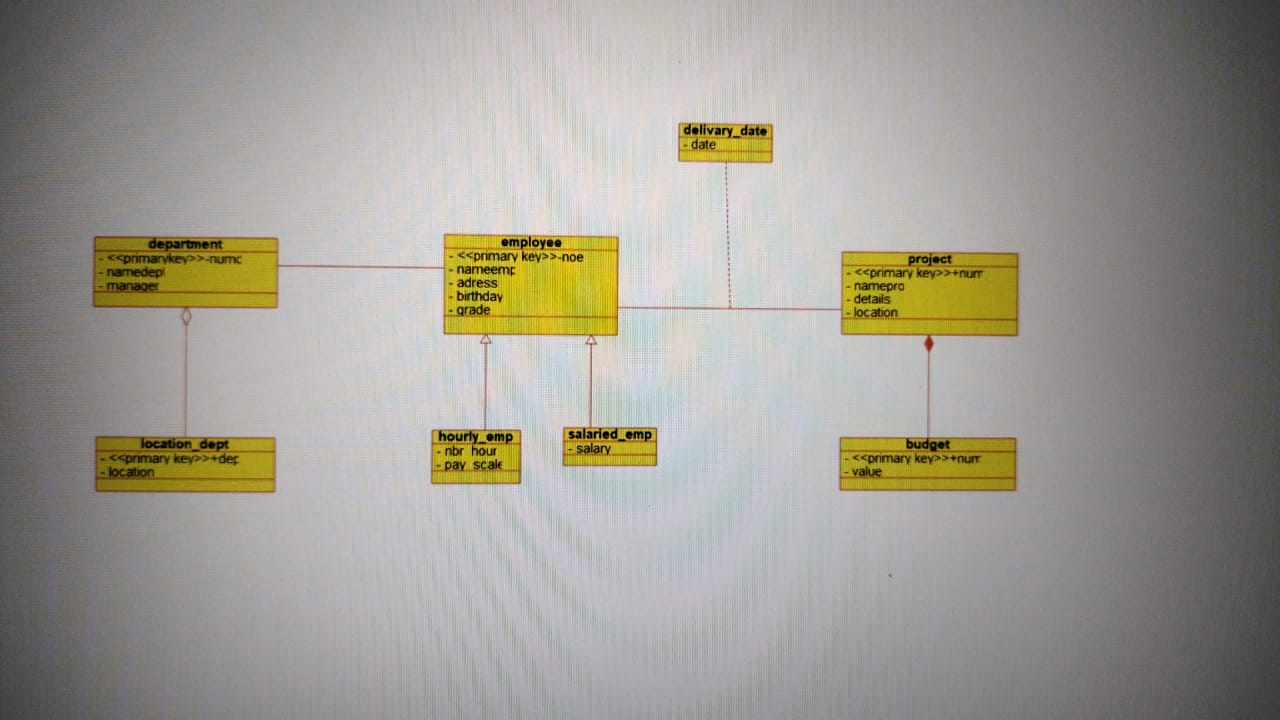
#endif // INTEGER\_

7.credit card processing system

Use case diagram



Class diagram



Source code

ifndef BUDGET\_H

#define BUDGET\_H

#include <string>

#include vector

/\*\*

\* class budget

\*

\*/

class budget

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

budget ();

/\*\*

\* Empty Destructor

\*/

virtual ~budget ();

// 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

//

void \_primary\_key\_numpro;

void value;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of \_primary\_key\_numpro

\* @param new\_var the new value of \_primary\_key\_numpro

\*/

void set\_primary\_key\_numpro (void new\_var) {

\_primary\_key\_numpro = new\_var;

}

/\*\*

\* Get the value of \_primary\_key\_numpro

\* @return the value of \_primary\_key\_numpro

\*/

void get\_primary\_key\_numpro () {

return \_primary\_key\_numpro;

}

/\*\*

\* Set the value of value

\* @param new\_var the new value of value

\*/

void setValue (void new\_var) {

value = new\_var;

}

/\*\*

\* Get the value of value

\* @return the value of value

\*/

void getValue () {

return value;

}

private:

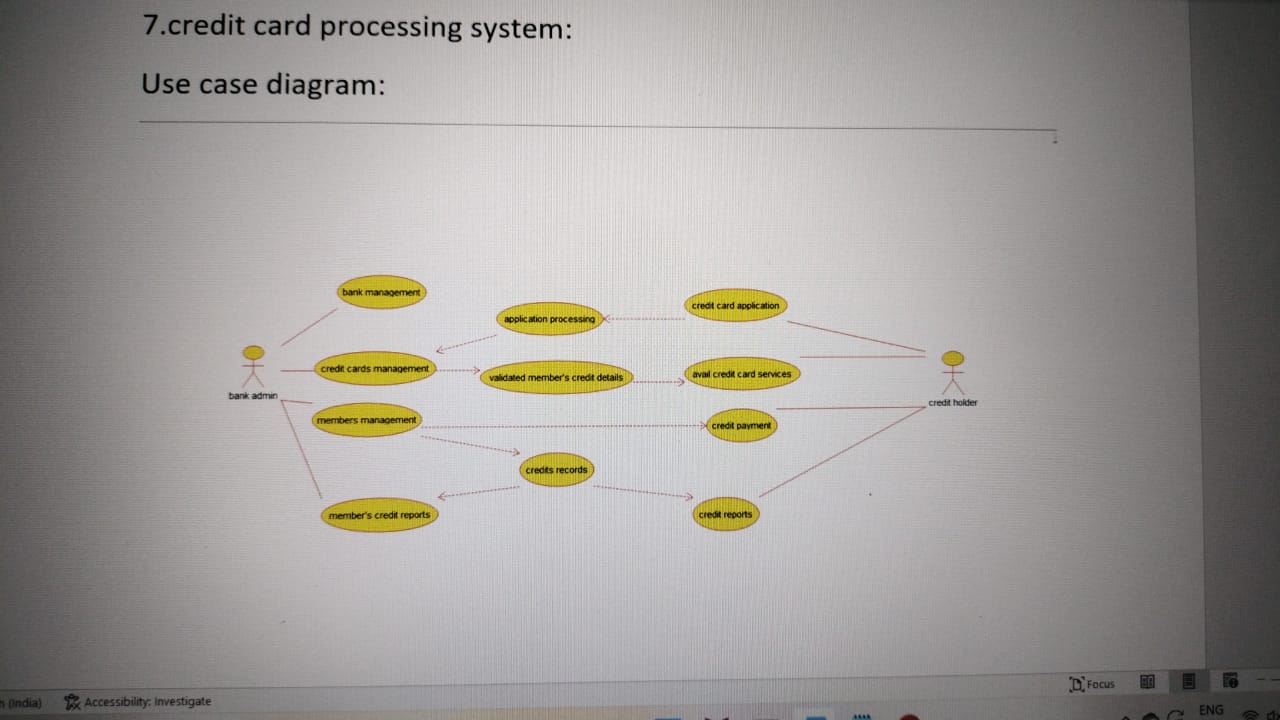
void initAttributes () ;

};

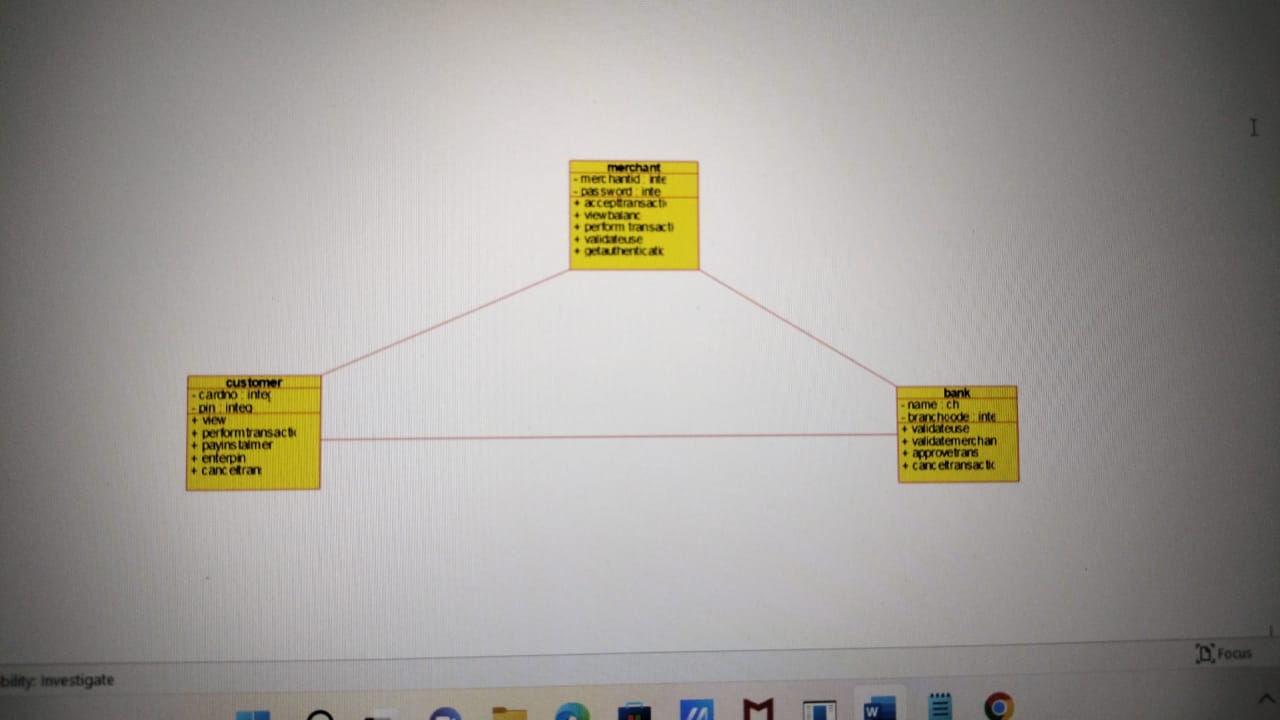
#endif // BUDGET\_H

8.software management system

Use case diagram



Class diagram



Source code

#ifndef INTEGER\_H

#define INTEGER\_H

#include <string>

/\*\*

\* class integer

\*

\*/

class integer

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

integer ();

/\*\*

\* Empty Destructor

\*/

virtual ~integer ();

// 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

//

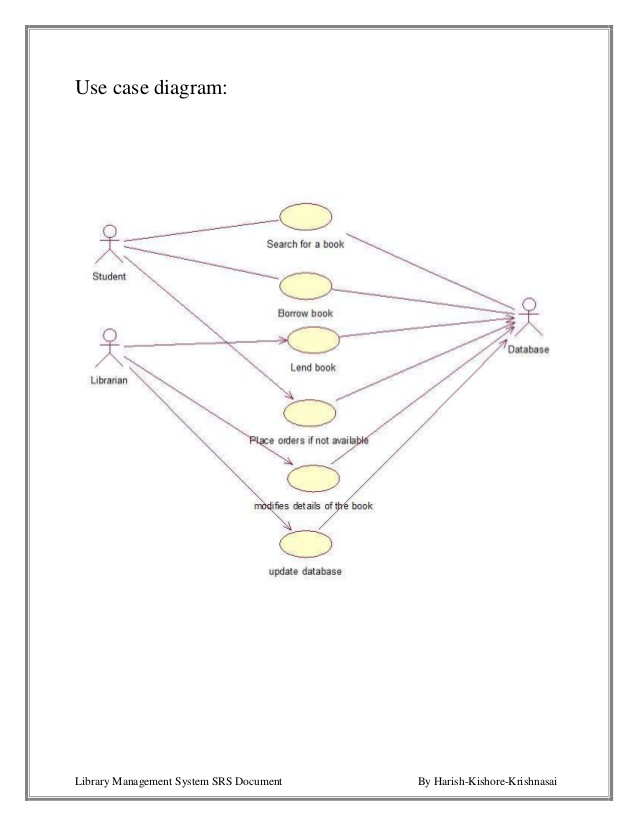
private:

};

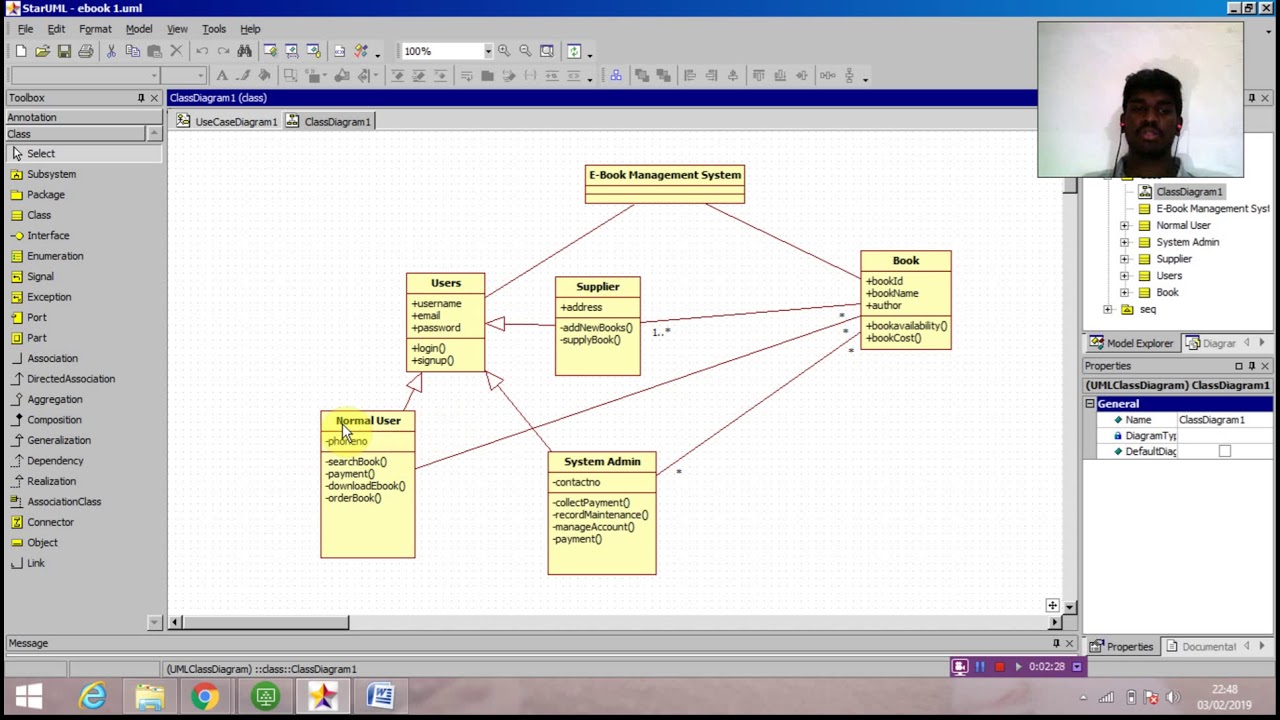
#endif // INTEGER\_H

9.e book management system

Use case diagram



Class diagram



Source code

#include "administrator.h"

// Constructors/Destructors

//

administrator::administrator () {

initAttributes();

}

administrator::~administrator () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void administrator::initAttributes () {

}

Download

#include "download.h"

// Constructors/Destructors

//

download::download () {

initAttributes();

}

download::~download () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void download::initAttributes () {

}

Book

#include "book.h"

// Constructors/Destructors

//

book::book () {

initAttributes();

}

book::~book () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void book::initAttributes () {

}

Registered user

#include "registered\_user.h"

// Constructors/Destructors

//

registered\_user::registered\_user () {

initAttributes();

}

registered\_user::~registered\_user () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void registered\_user::initAttributes () {

}

Visitor

#include "visitor.h"

// Constructors/Destructors

//

visitor::visitor () {

initAttributes();

}

visitor::~visitor () { }

//

// Methods

//

// Accessor methods

//

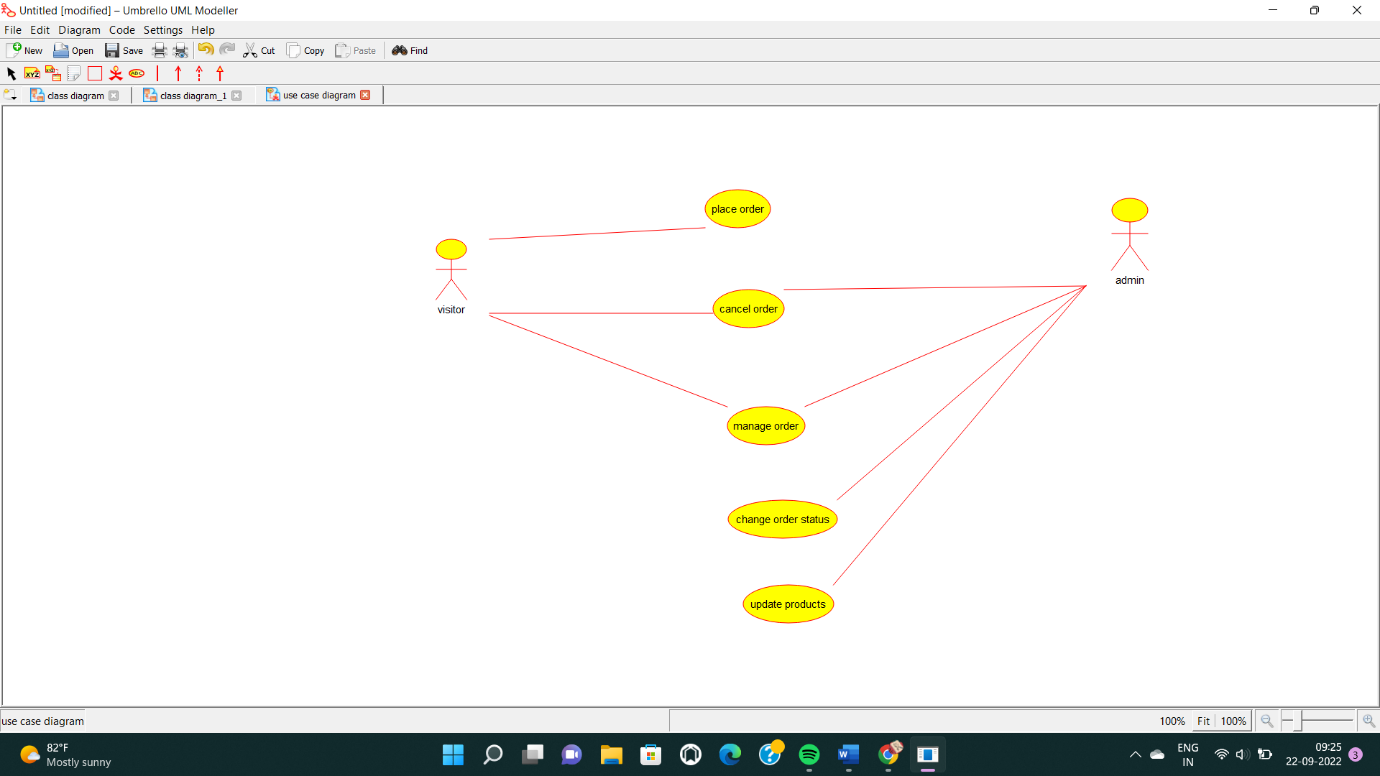
// Other methods

//

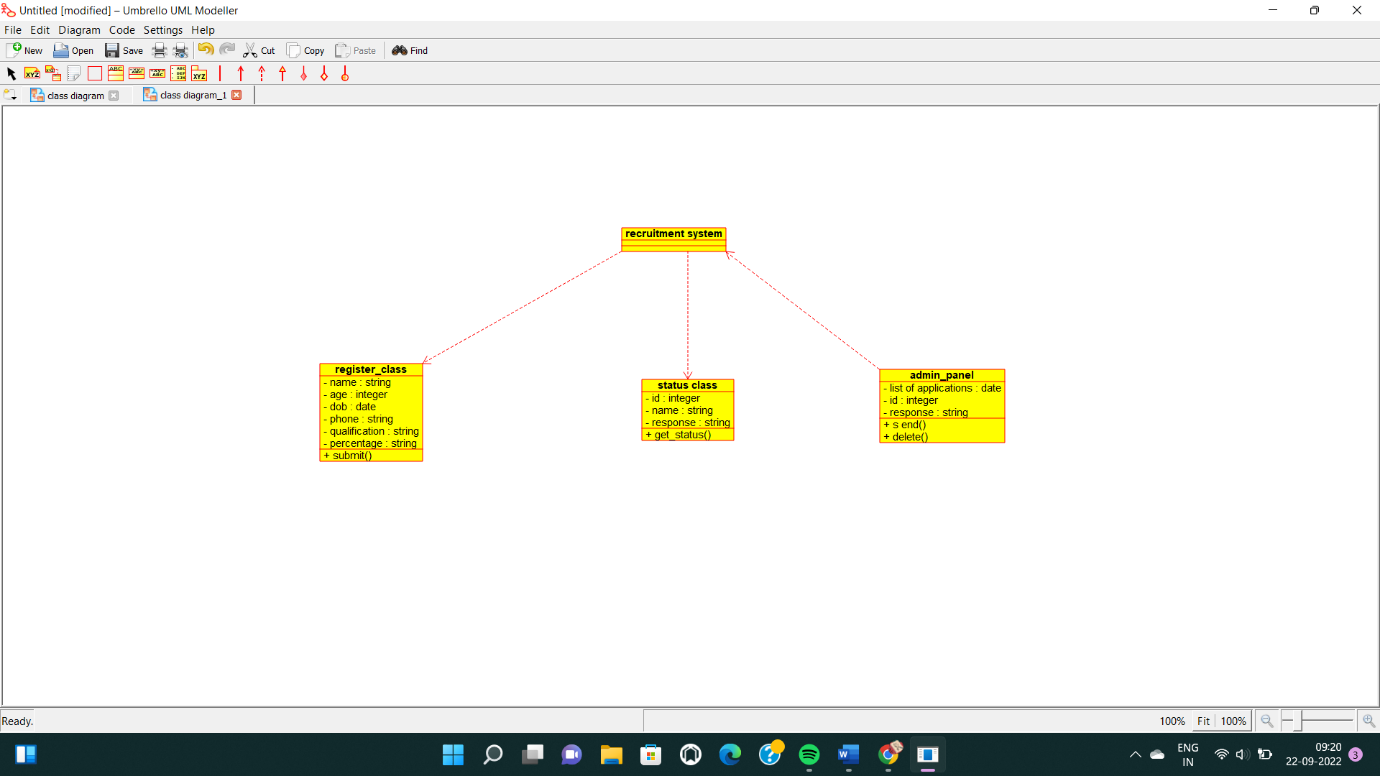
void visitor::initAttributes () {

10.recruitment system

Use case digram



Class digram



Sorure code

#include "admin\_panel.h"

// Constructors/Destructors

//

admin\_panel::admin\_panel () {

initAttributes();

}

admin\_panel::~admin\_panel () { }

//

// Methods

//

// Accessor methods

//

// Other methods

//

void admin\_panel::initAttributes () {

}