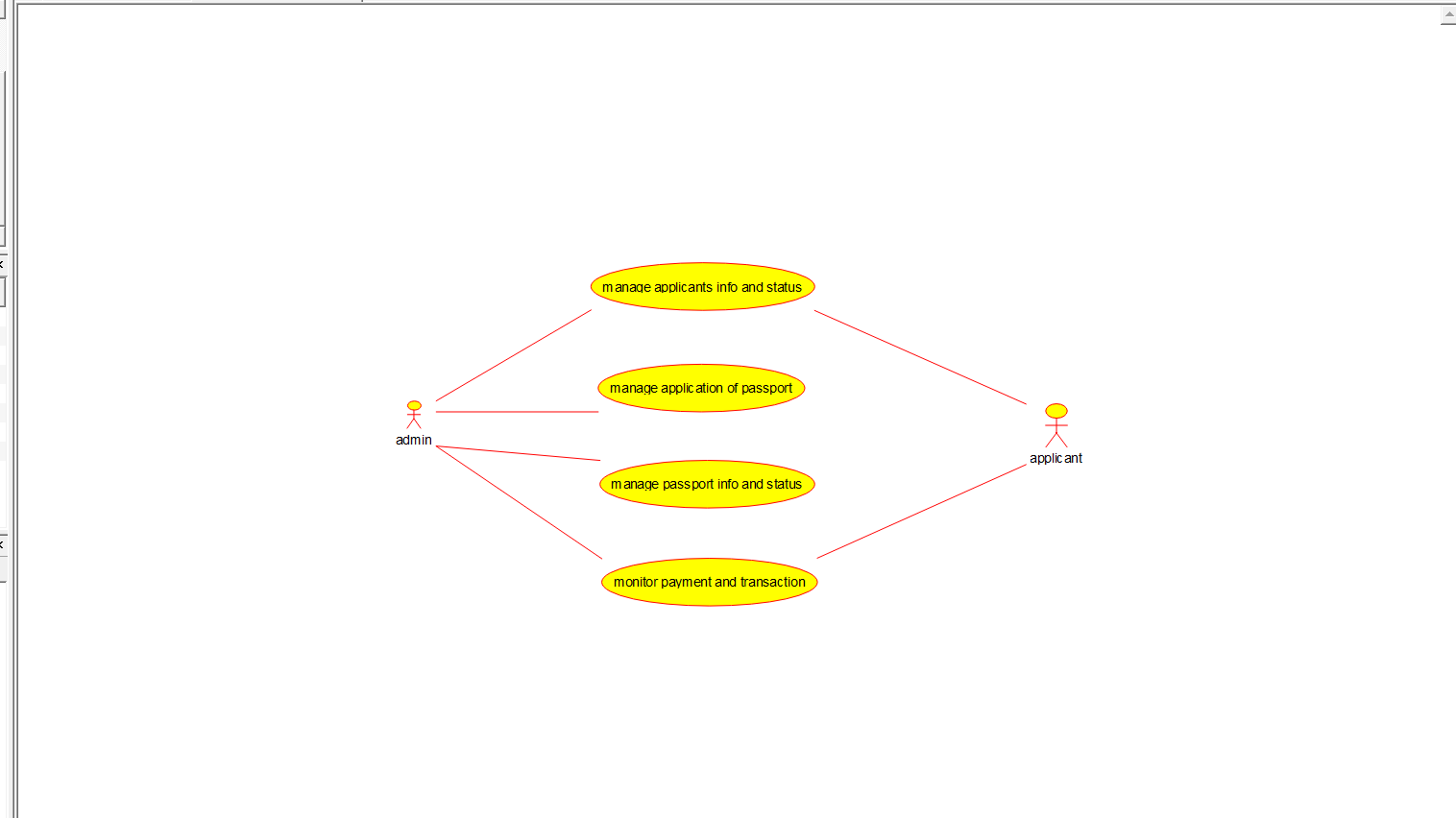
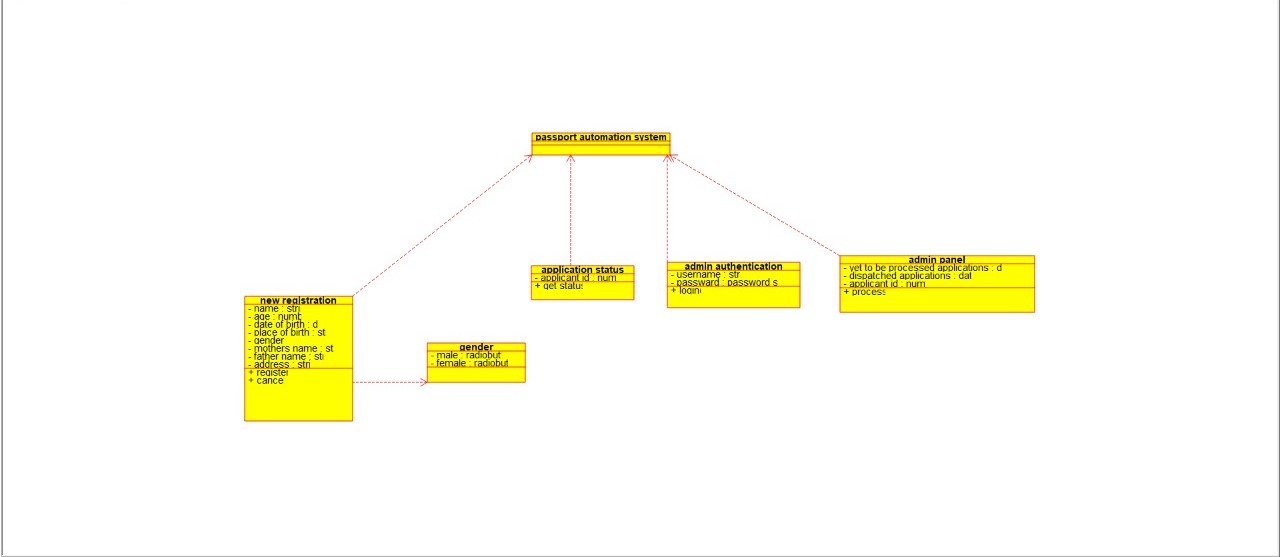
1.passport system

Use case 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

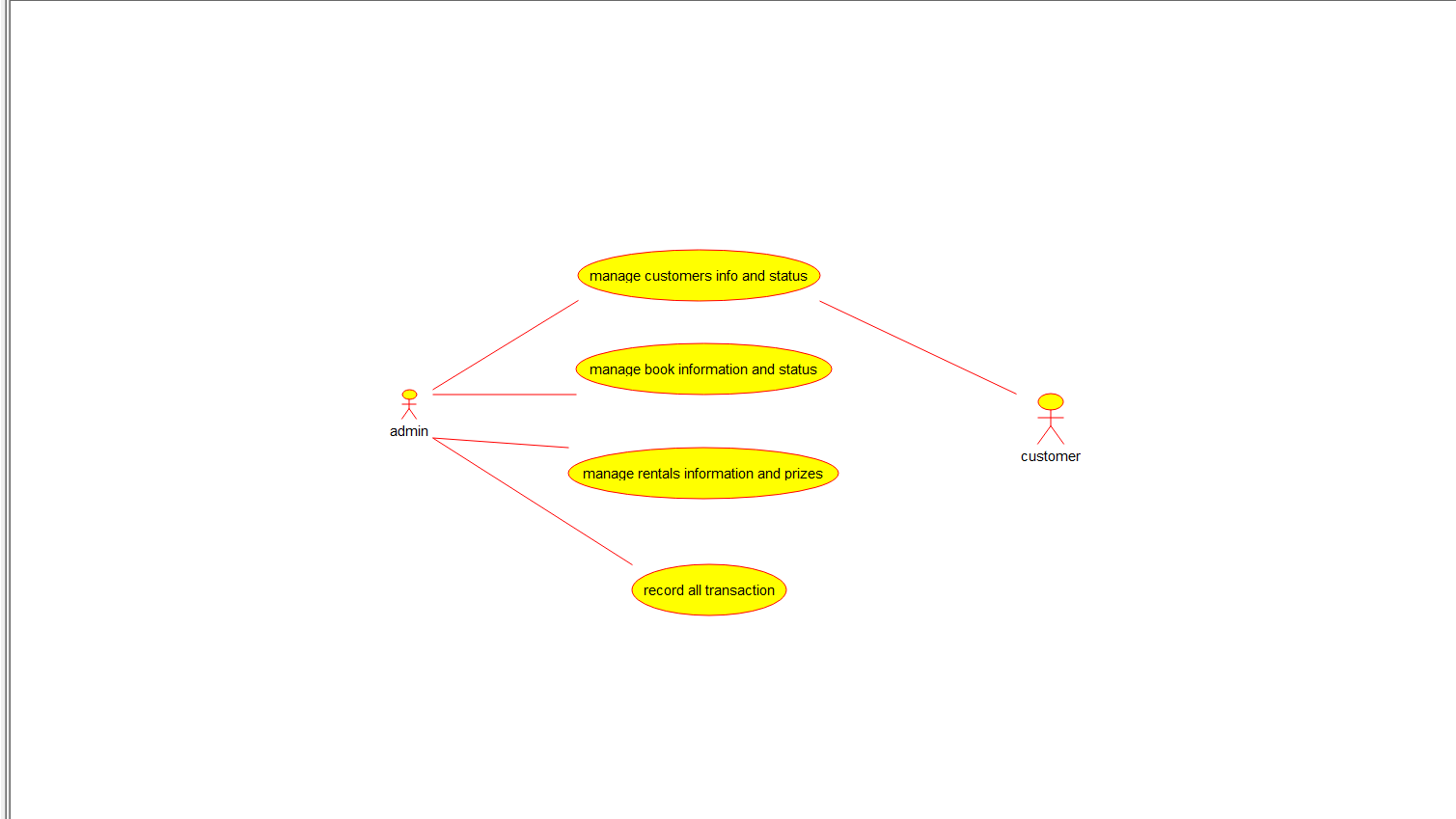
//

void new\_registration::initAttributes ()

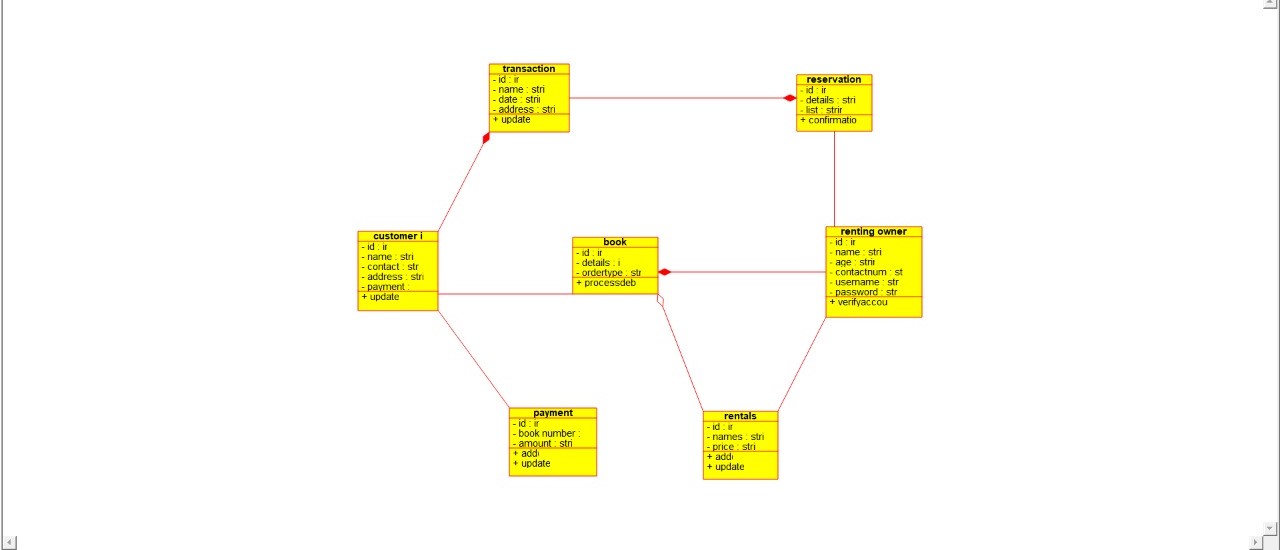
{

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

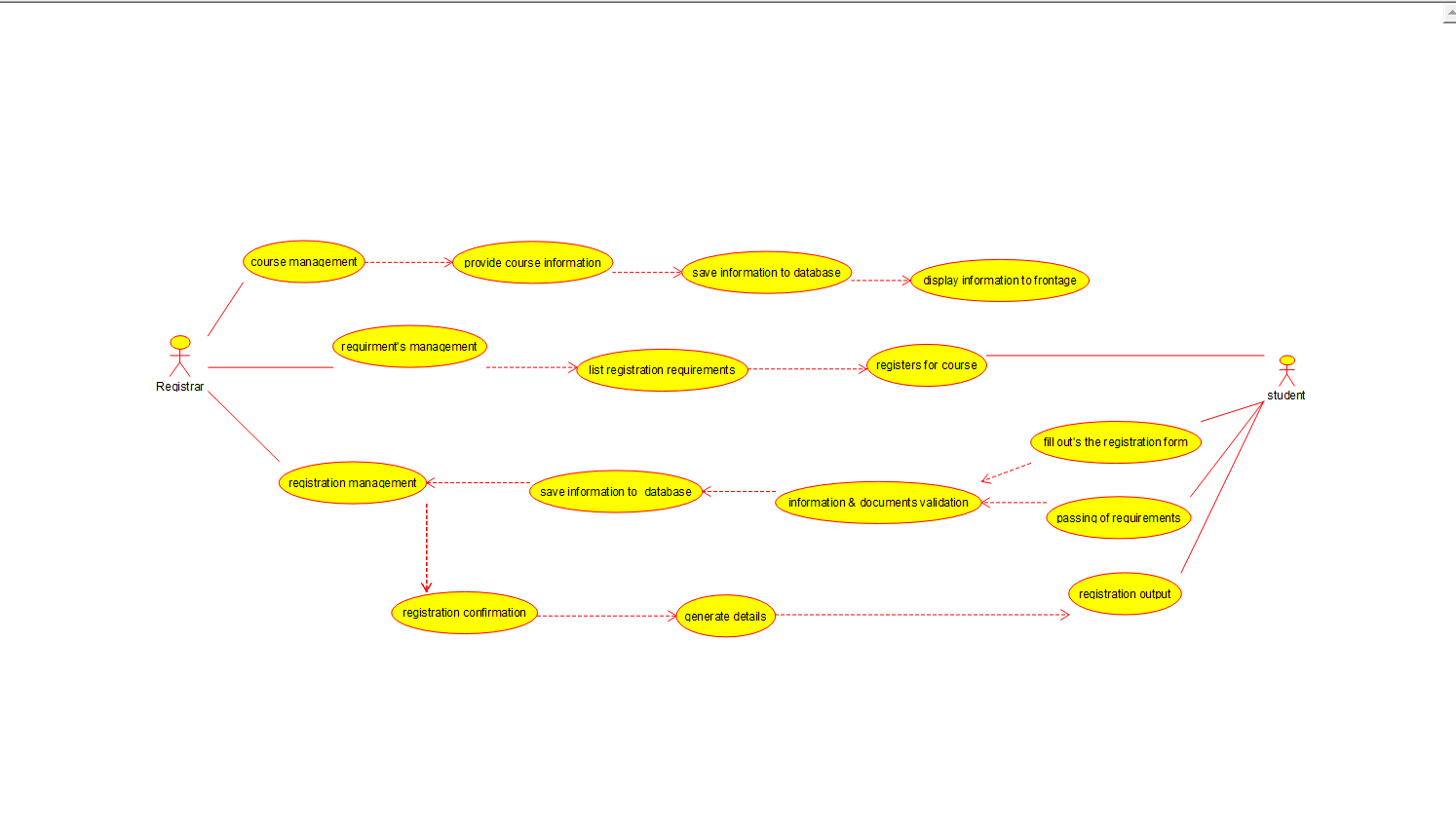
//

void book::initAttributes () {

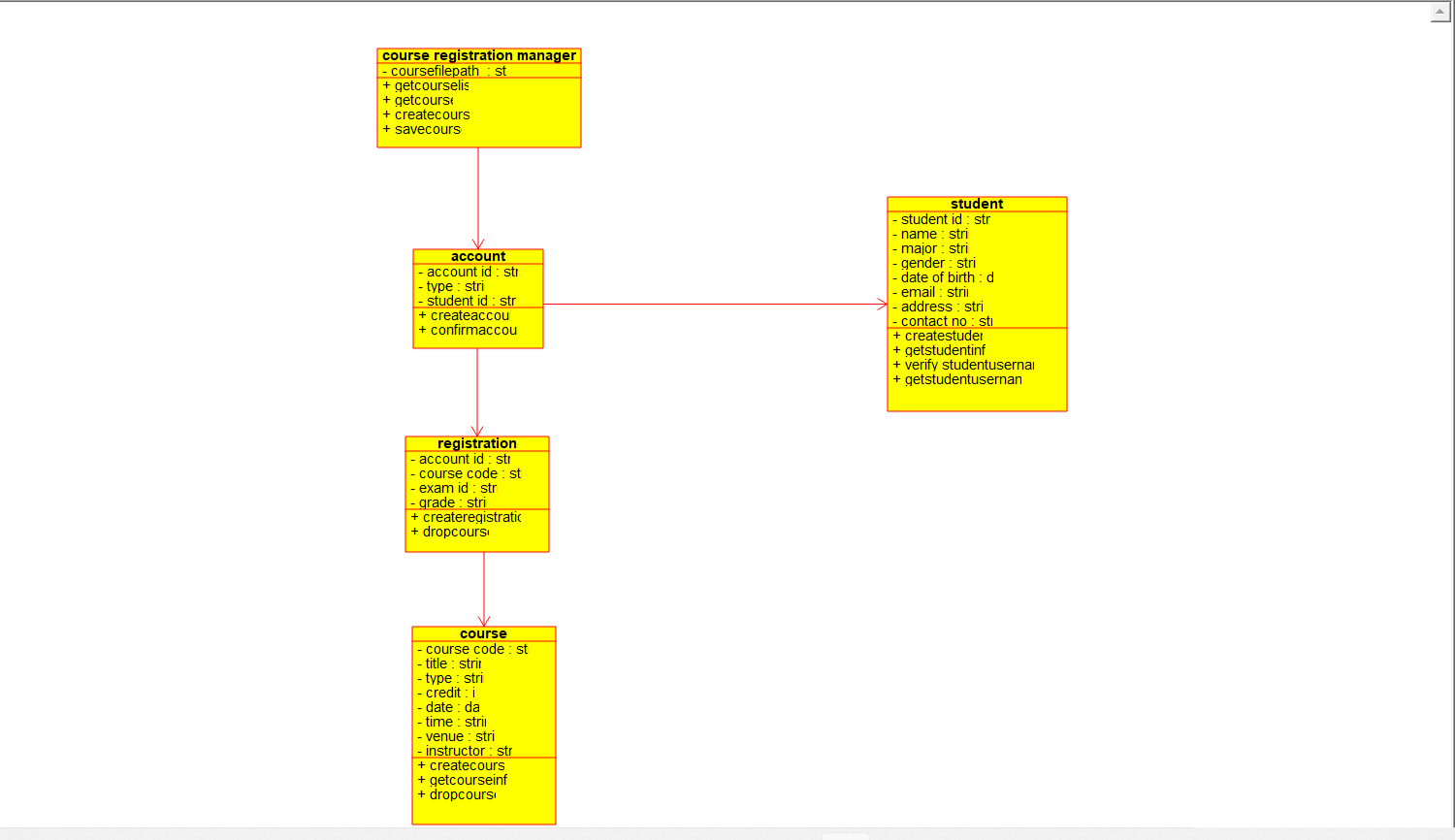
}

3.course registration

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:

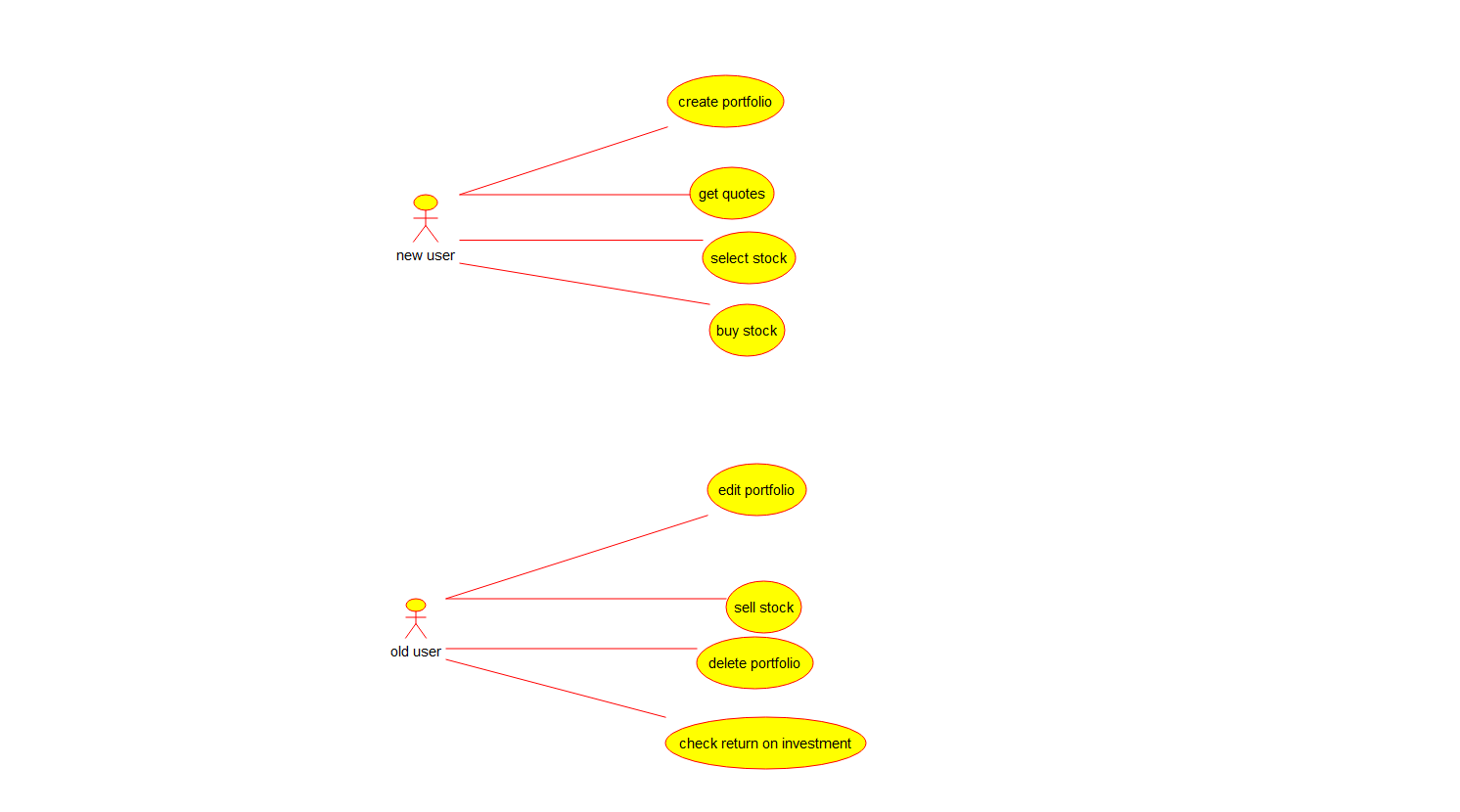
void initAttributes () ;

};

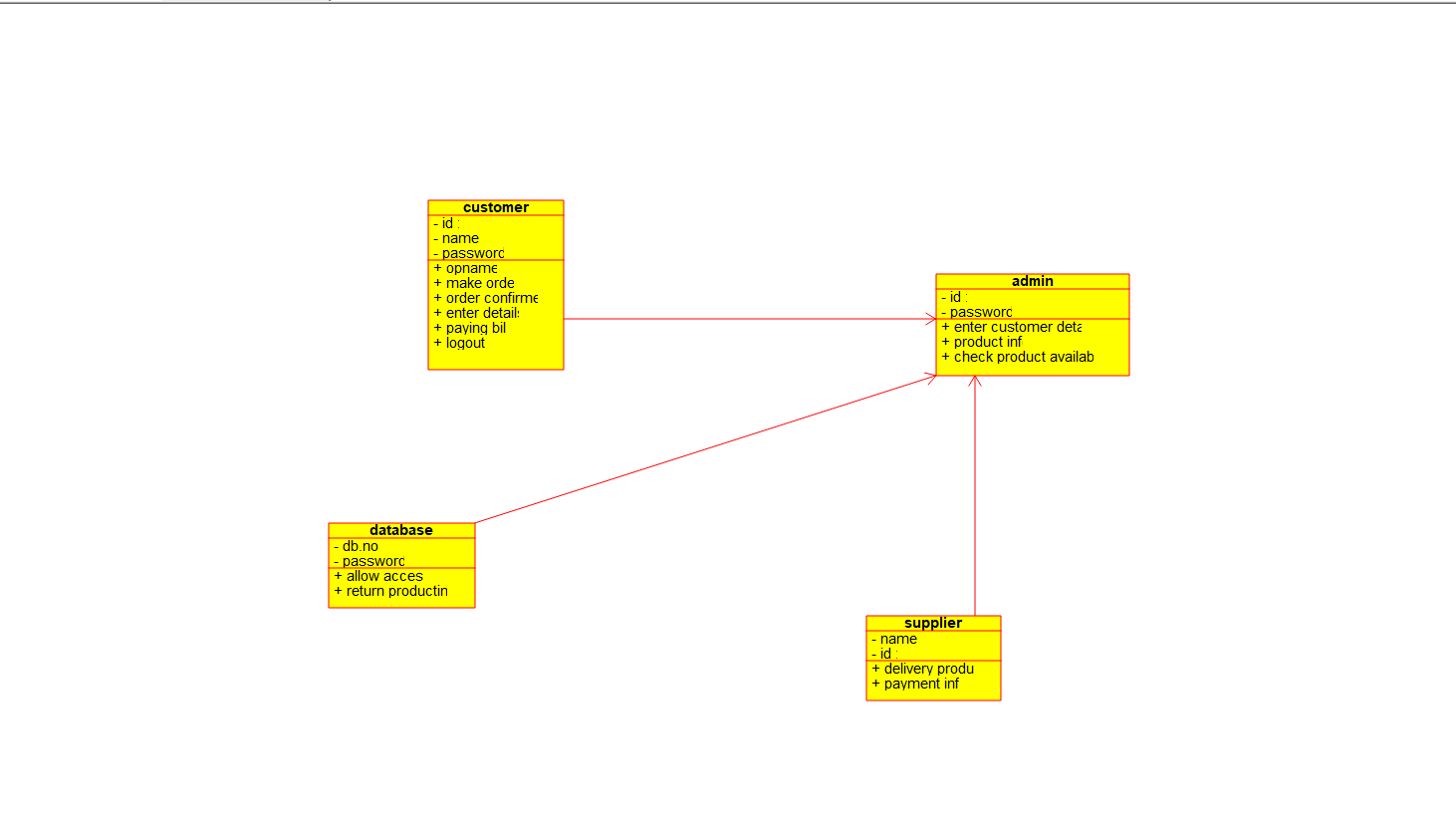
#endif // ACCOUNT\_H

4.stock analysis system

Use case diagram:



Class diagram:



Source code:

#ifndef ADMIN\_H

#define ADMIN\_H

#include <string>

/\*\*

\* class admin

\*

\*/

class admin

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

admin ();

/\*\*

\* Empty Destructor

\*/

virtual ~admin ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void enter\_customer\_details ()

{

}

/\*\*

\*/

void product\_info ()

{

}

/\*\*

\*/

void check\_product\_availability ()

{

}

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 id;

void password;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of id

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

\*/

void setId (void new\_var) {

id = new\_var;

}

/\*\*

\* Get the value of id

\* @return the value of id

\*/

void getId () {

return id;

}

/\*\*

\* Set the value of password

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

\*/

void setPassword (void new\_var) {

password = new\_var;

}

/\*\*

\* Get the value of password

\* @return the value of password

\*/

void getPassword () {

return password;

}

private:

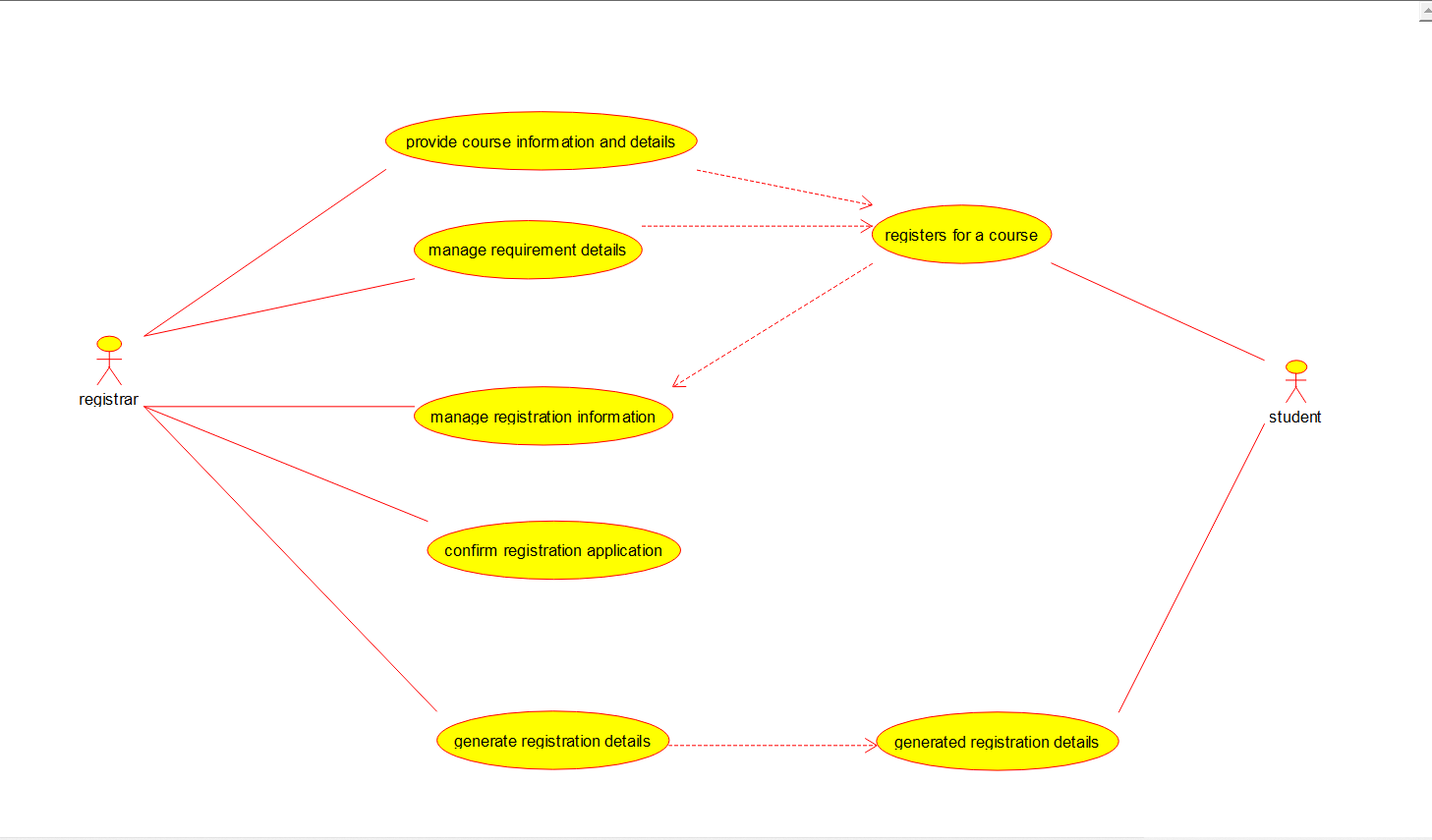
void initAttributes () ;

};

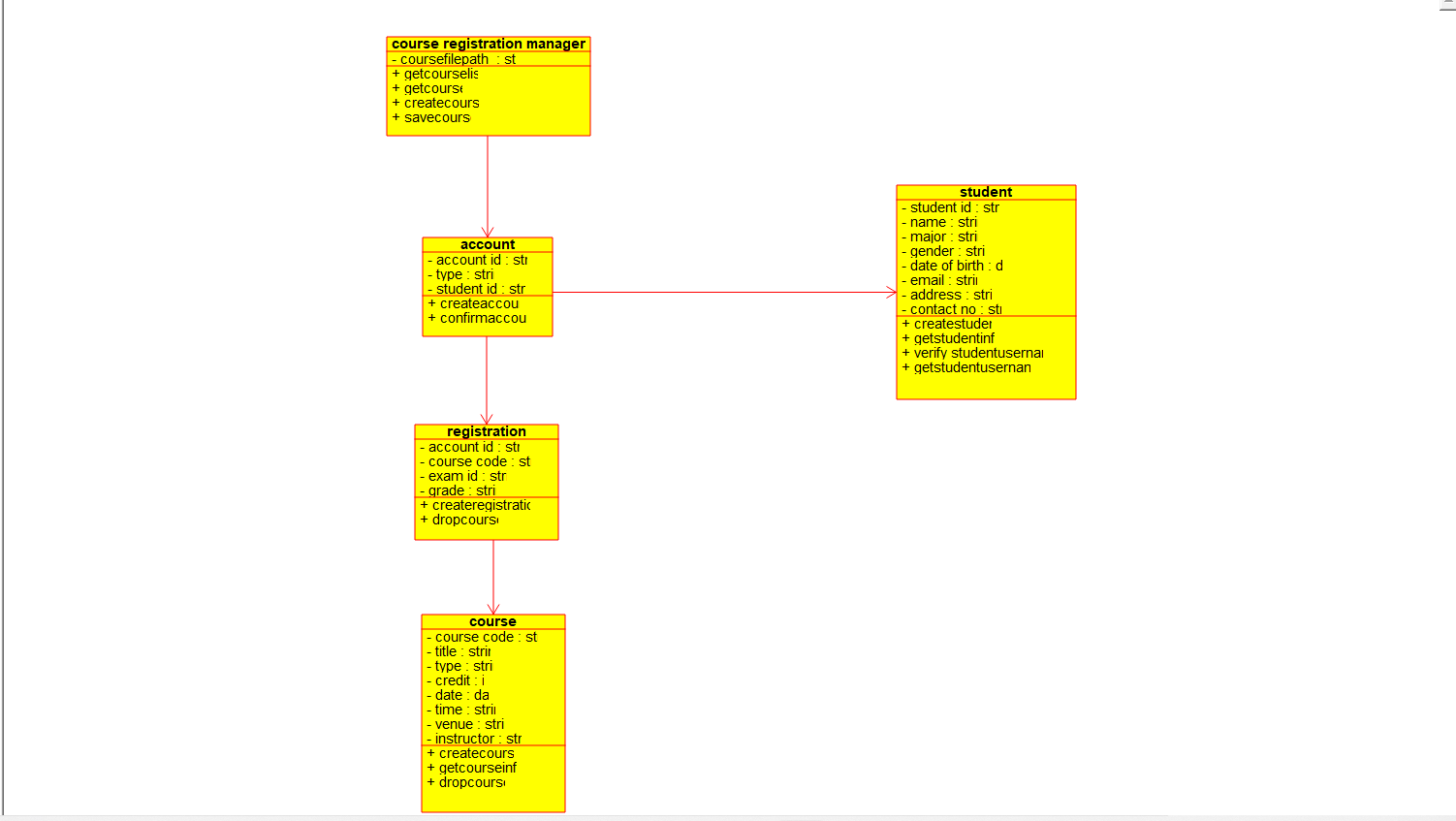
#endif // ADMIN\_H

5.online course reservation 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:

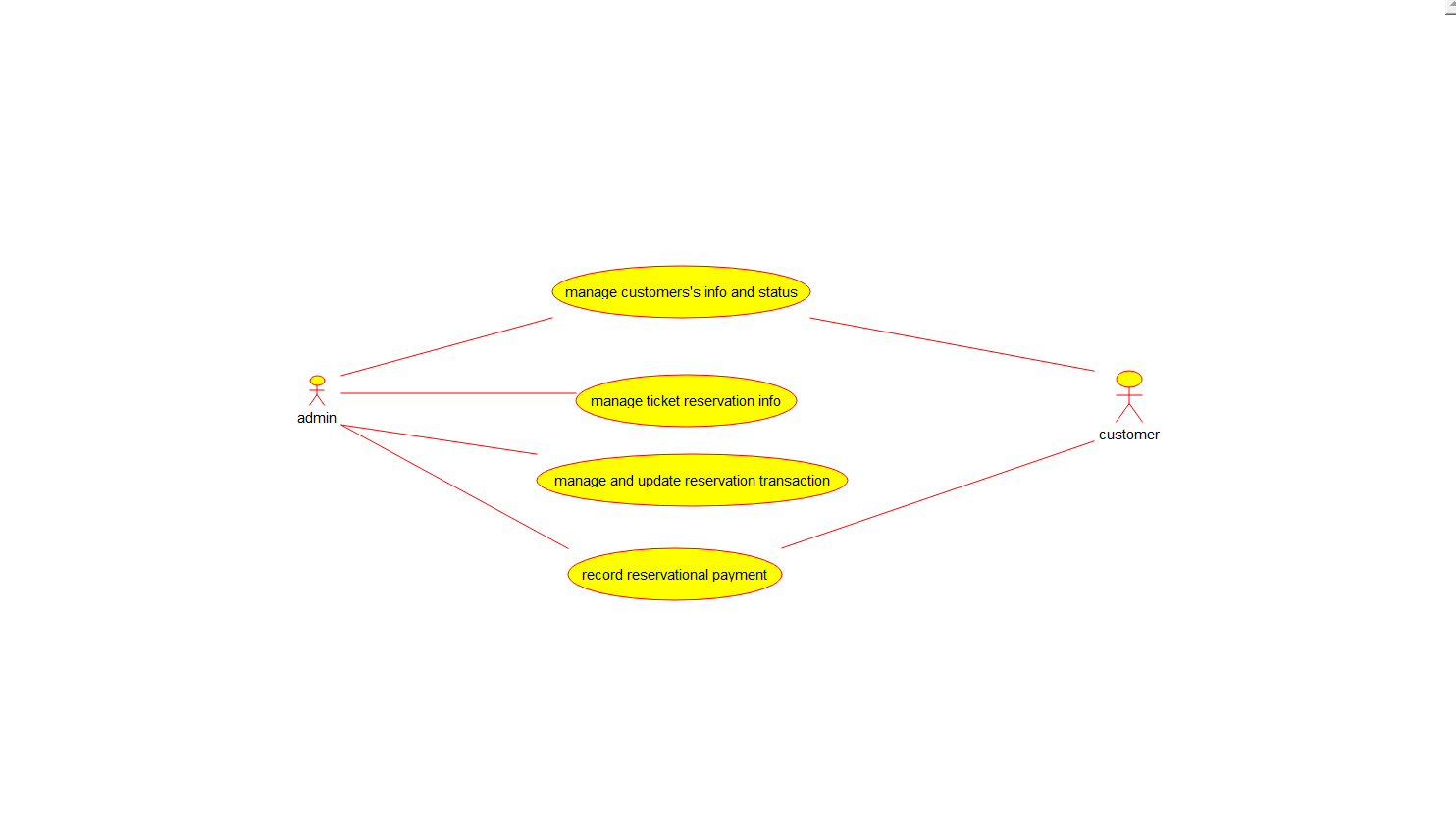
void initAttributes () ;

};

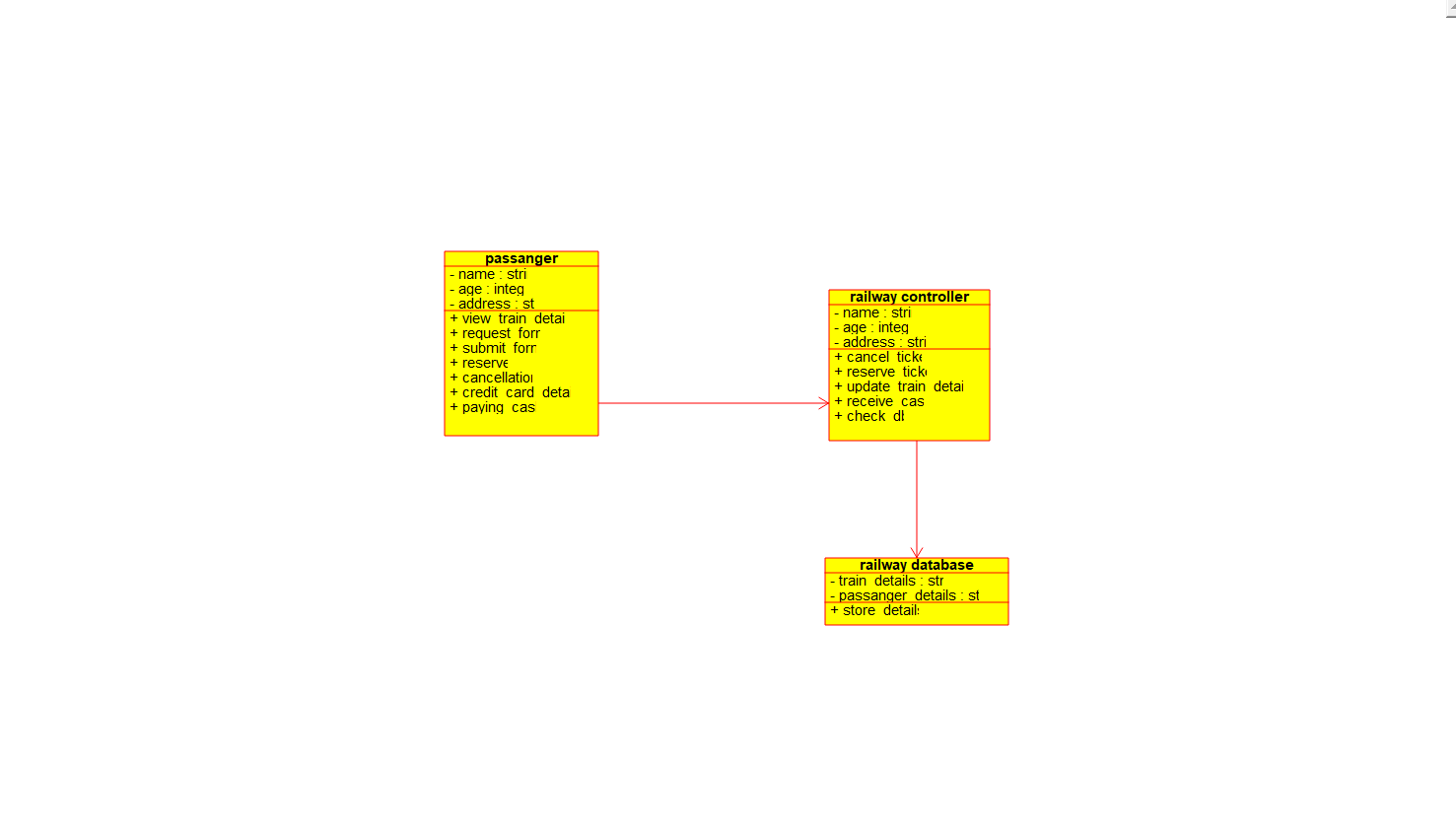
#endif // ACCOUNT\_H

6.F-ticketing

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

//

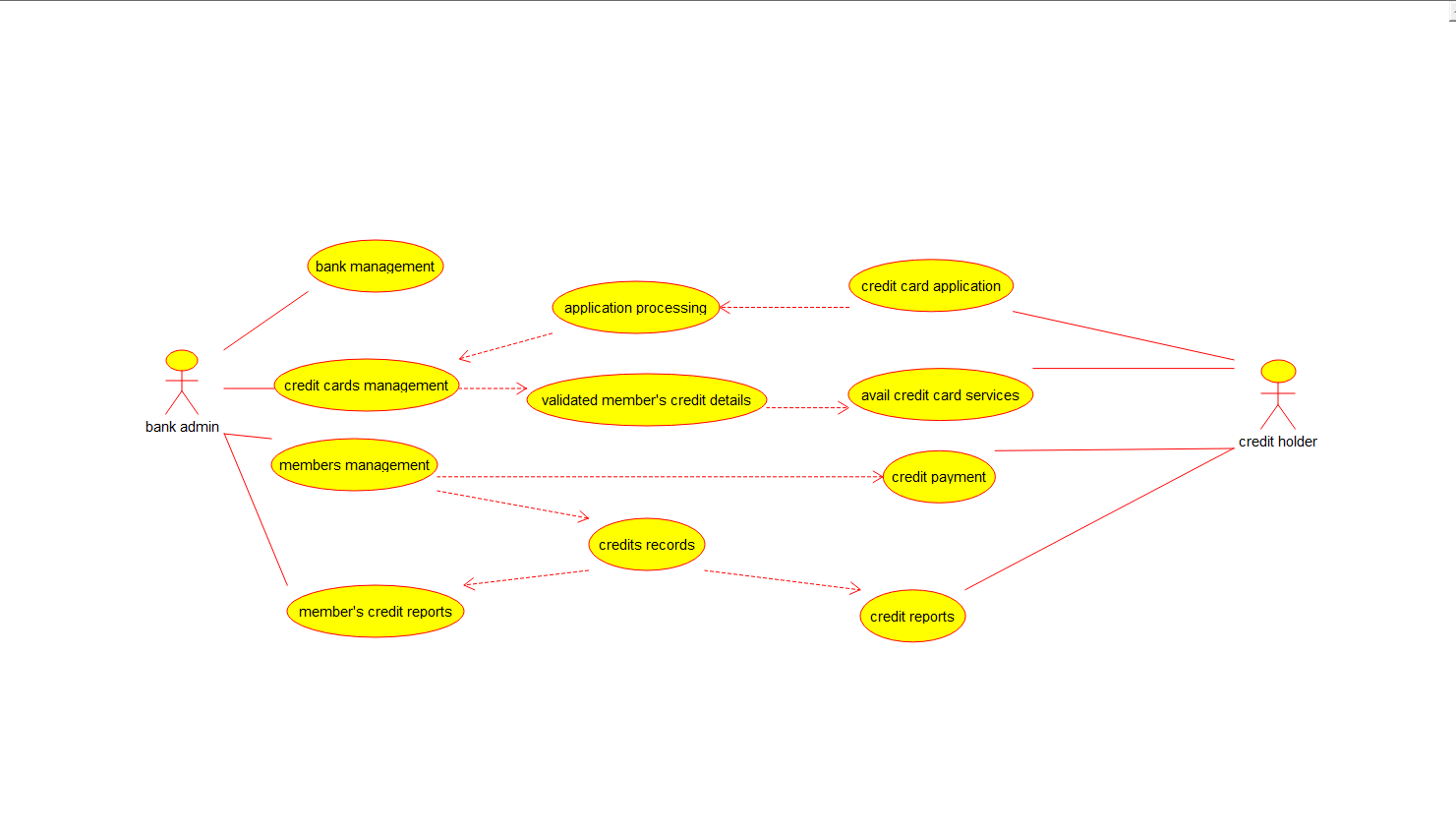
private:

};

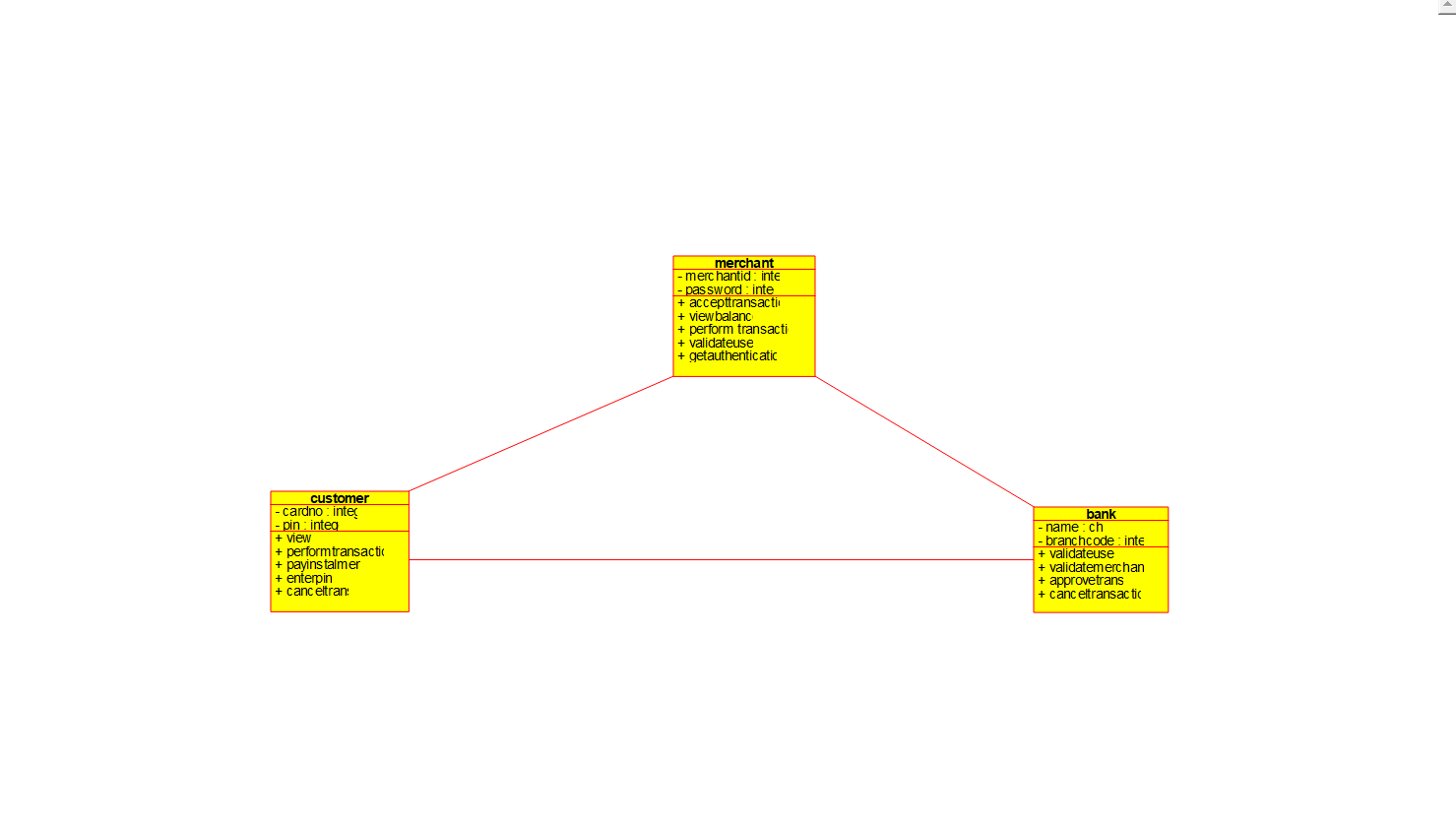
#endif // INTEGER\_H

7.credit card processing system:

Use case diagram:



Class diagram:



Source code:

#ifndef BANK\_H

#define BANK\_H

#include <string>

#include vector

/\*\*

\* class bank

\*

\*/

class bank

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

bank ();

/\*\*

\* Empty Destructor

\*/

virtual ~bank ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void validateuser ()

{

}

/\*\*

\*/

void validatemerchants ()

{

}

/\*\*

\*/

void approvetransc ()

{

}

/\*\*

\*/

void canceltransaction ()

{

}

protected:

// Static Protected attributes

//

// Protected attributes

//

public:

// Protected attribute accessor methods

//

protected:

public:

// Protected attribute accessor methods

//

protected:

private:

// Static Private attributes

//

// Private attributes

//

char name;

integer branchcode;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of name

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

\*/

void setName (char new\_var) {

name = new\_var;

}

/\*\*

\* Get the value of name

\* @return the value of name

\*/

char getName () {

return name;

}

/\*\*

\* Set the value of branchcode

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

\*/

void setBranchcode (integer new\_var) {

branchcode = new\_var;

}

/\*\*

\* Get the value of branchcode

\* @return the value of branchcode

\*/

integer getBranchcode () {

return branchcode;

}

private:

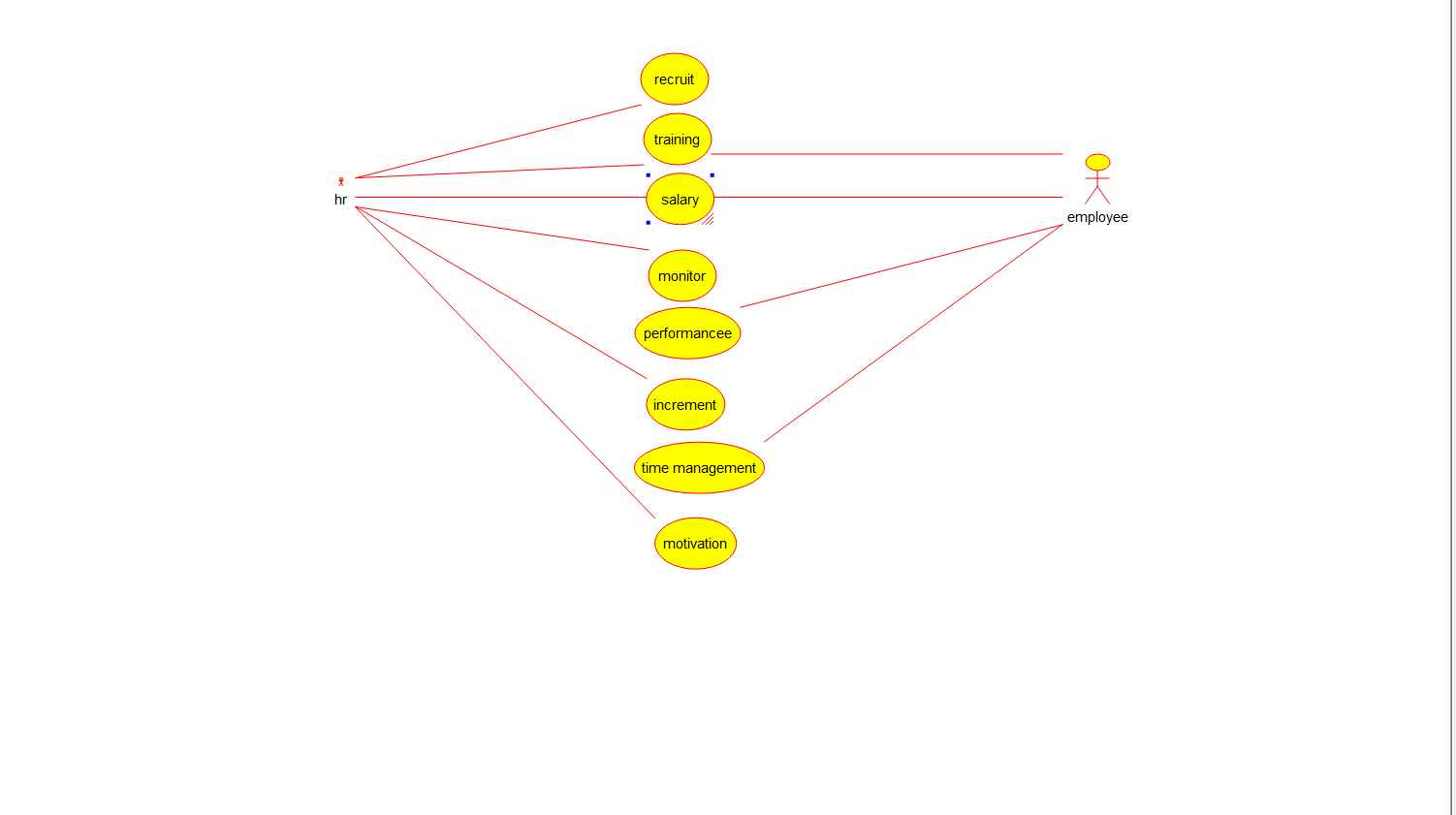
void initAttributes () ;

};

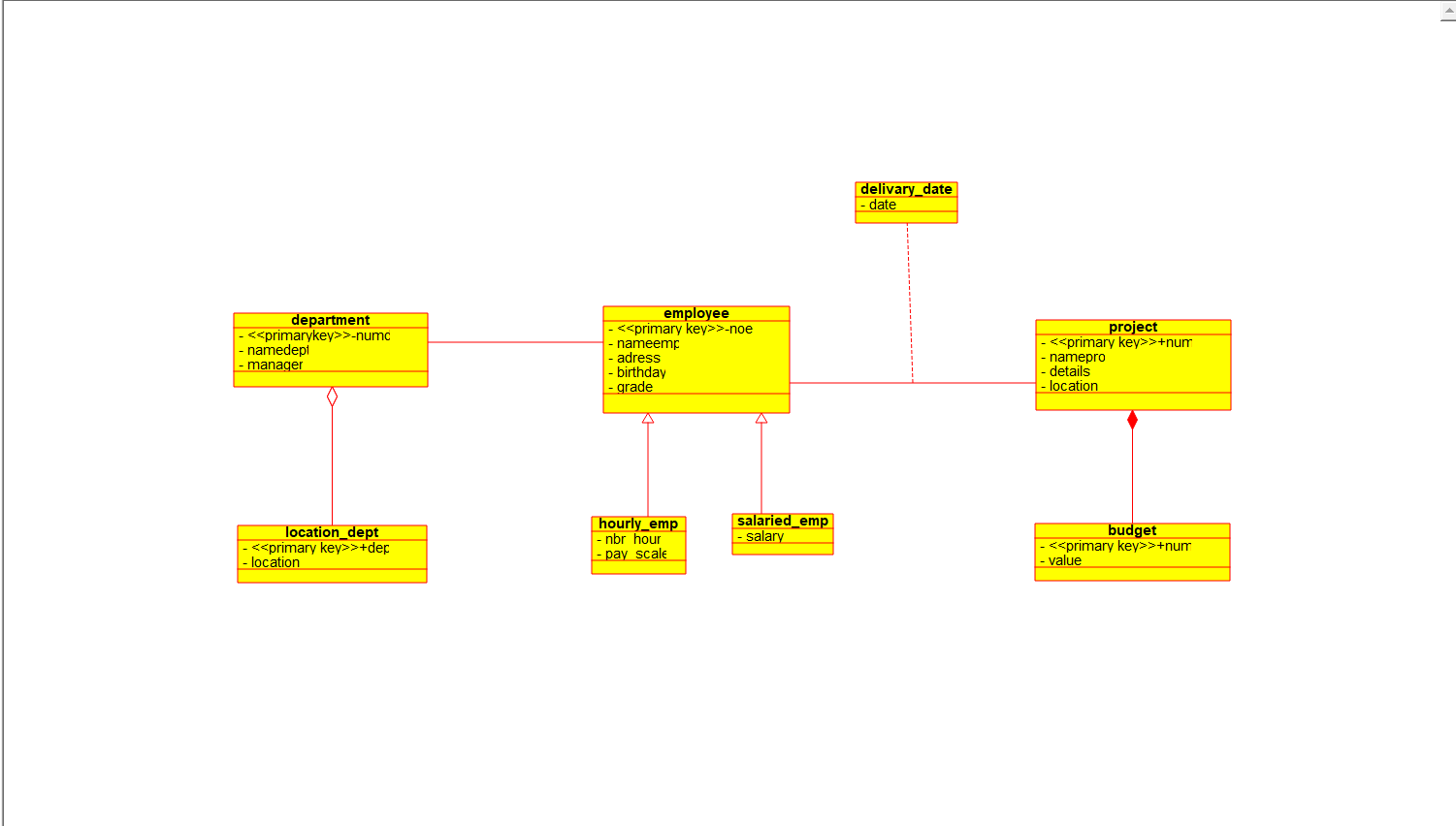
#endif // BANK\_H

8.software personal management 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:

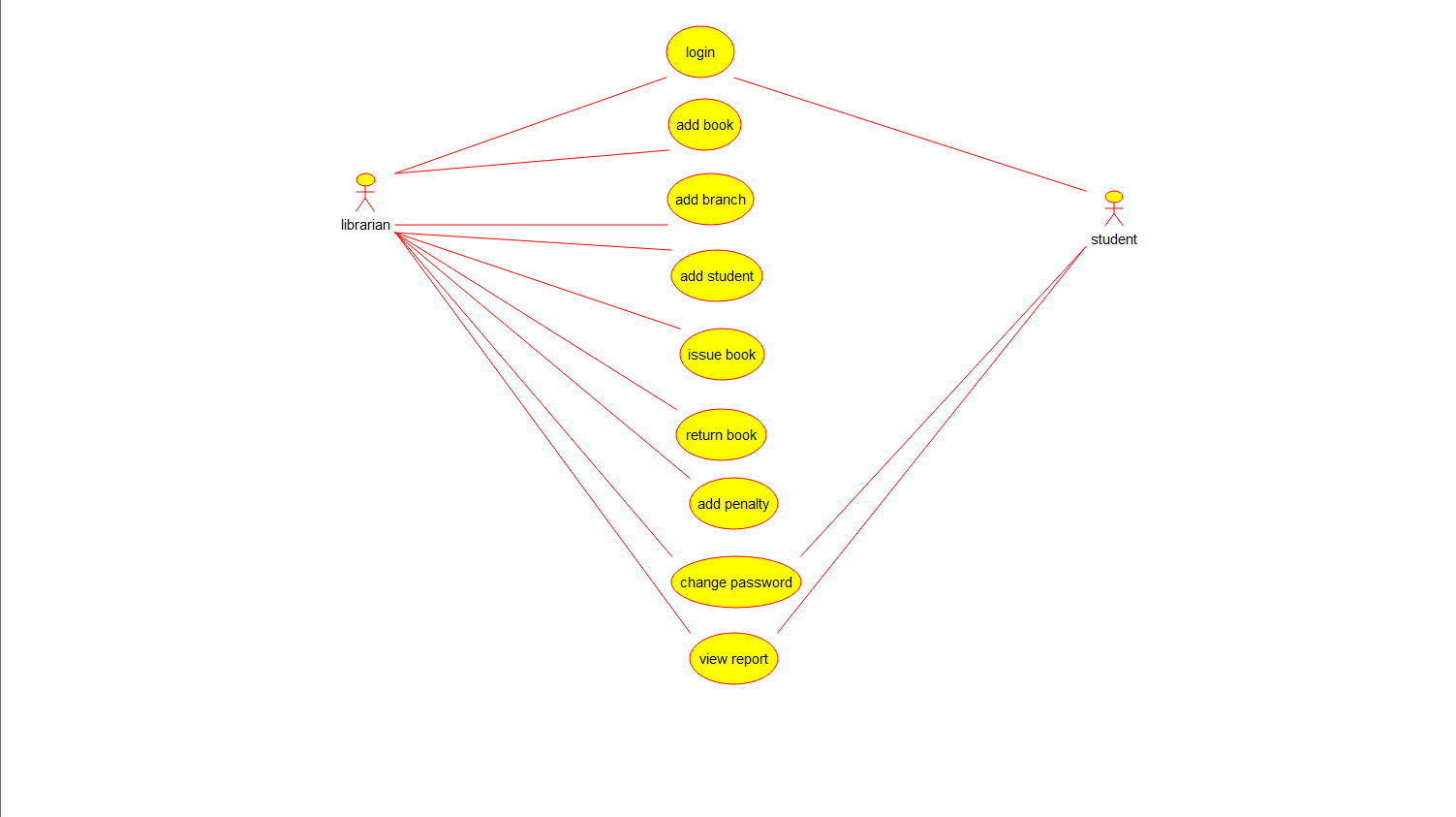
void initAttributes () ;

};

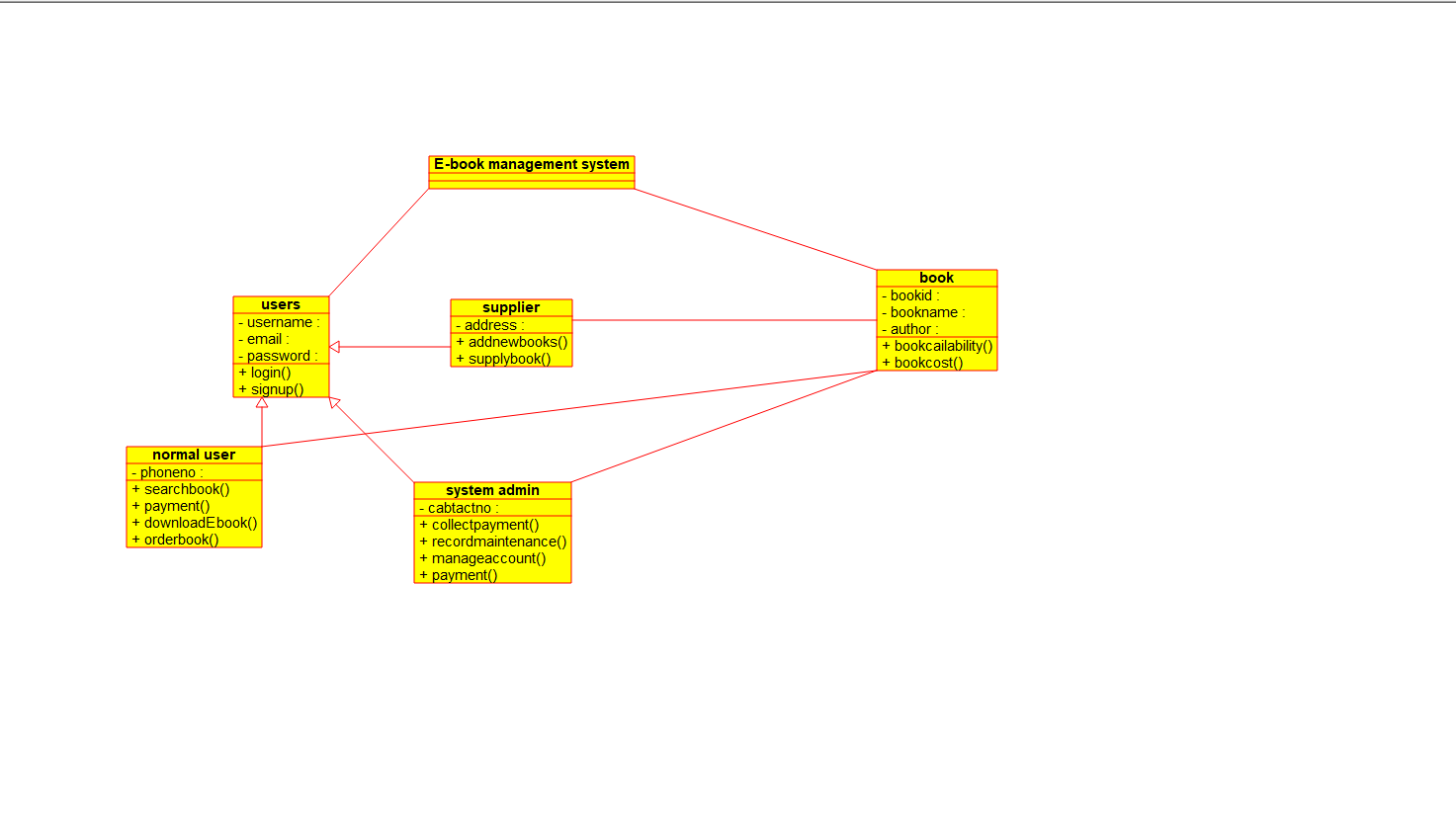
#endif // BUDGET\_H

9.E-book management system

Use case diagram:



Class diagram:



Source code:

#ifndef BOOK\_H

#define BOOK\_H

#include <string>

#include vector

/\*\*

\* class book

\*

\*/

class book

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

book ();

/\*\*

\* Empty Destructor

\*/

virtual ~book ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void bookcailability ()

{

}

/\*\*

\*/

void bookcost ()

{

}

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 bookid;

void bookname;

void author;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of bookid

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

\*/

void setBookid (void new\_var) {

bookid = new\_var;

}

/\*\*

\* Get the value of bookid

\* @return the value of bookid

\*/

void getBookid () {

return bookid;

}

/\*\*

\* Set the value of bookname

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

\*/

void setBookname (void new\_var) {

bookname = new\_var;

}

/\*\*

\* Get the value of bookname

\* @return the value of bookname

\*/

void getBookname () {

return bookname;

}

/\*\*

\* Set the value of author

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

\*/

void setAuthor (void new\_var) {

author = new\_var;

}

/\*\*

\* Get the value of author

\* @return the value of author

\*/

void getAuthor () {

return author;

}

private:

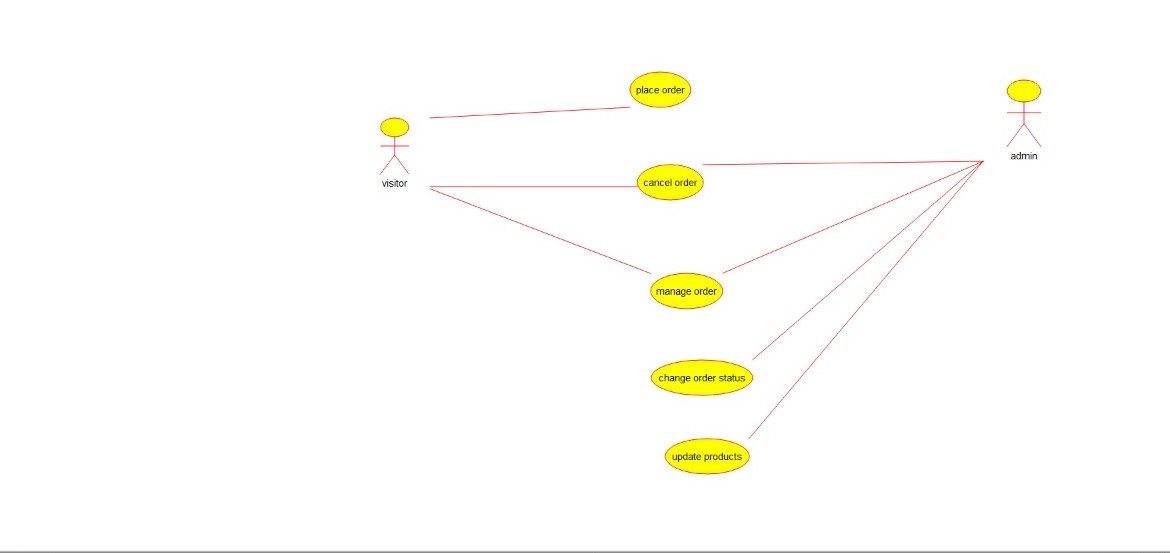
void initAttributes () ;

};

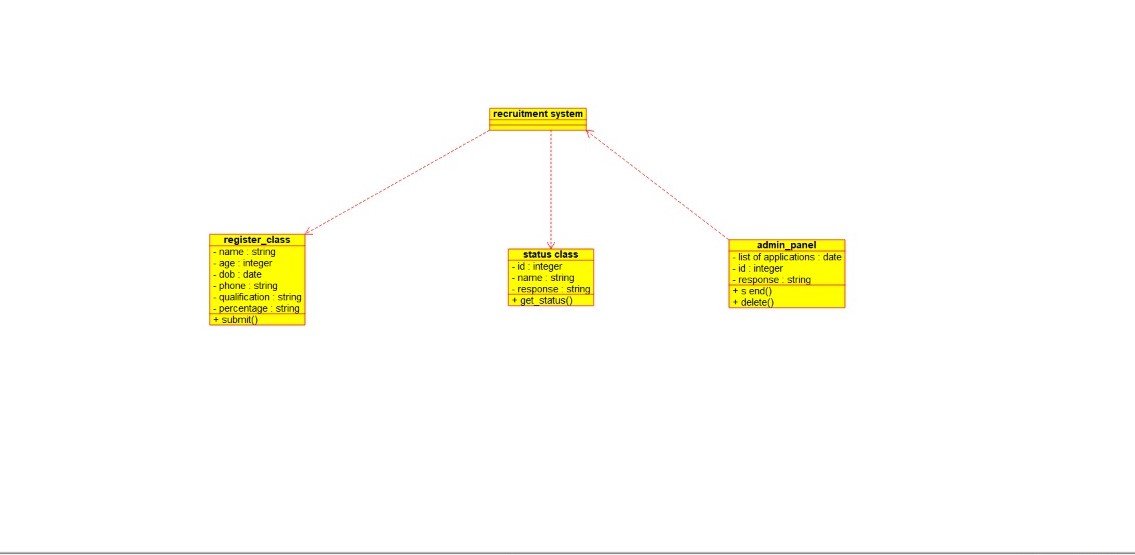
#endif // BOOK\_H

10.recruitment system

Use case diagram:



Class diagram:



Source 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 () {

}