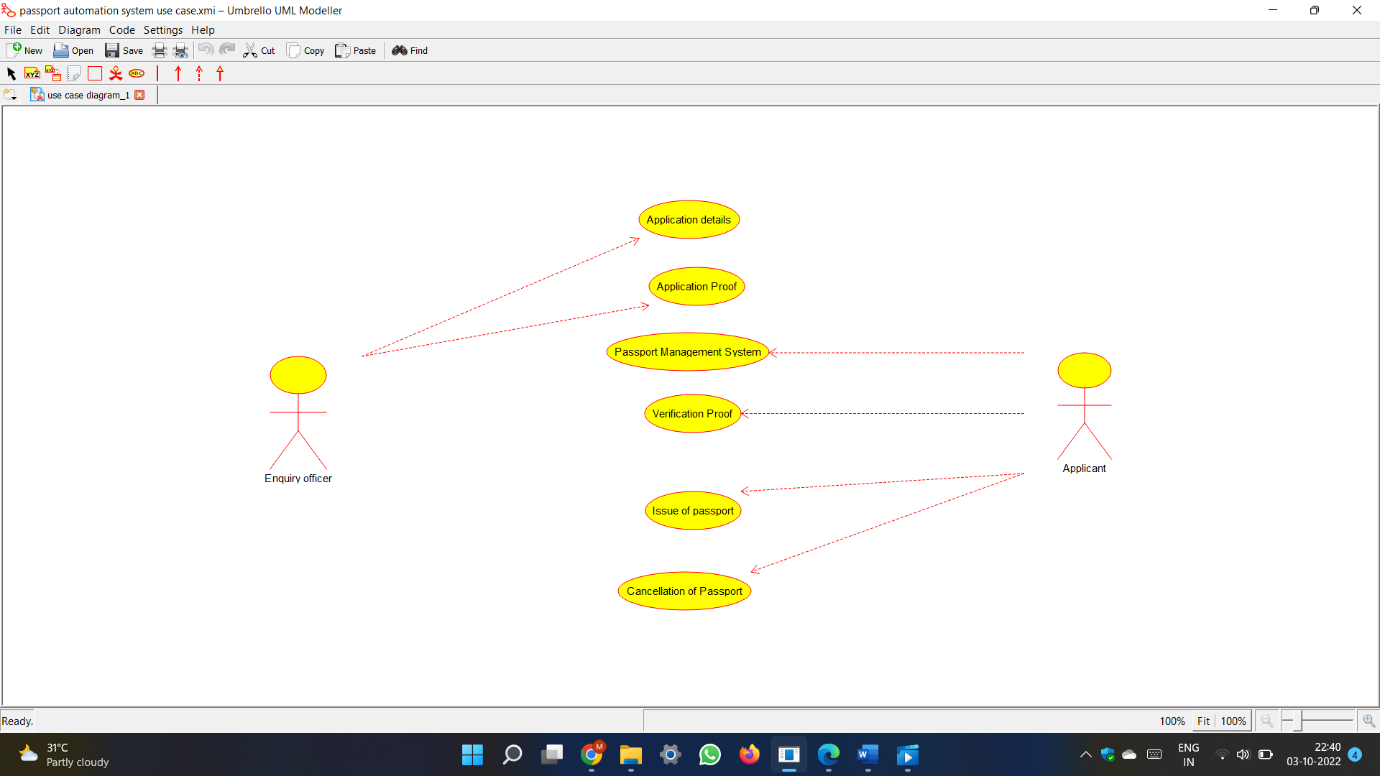
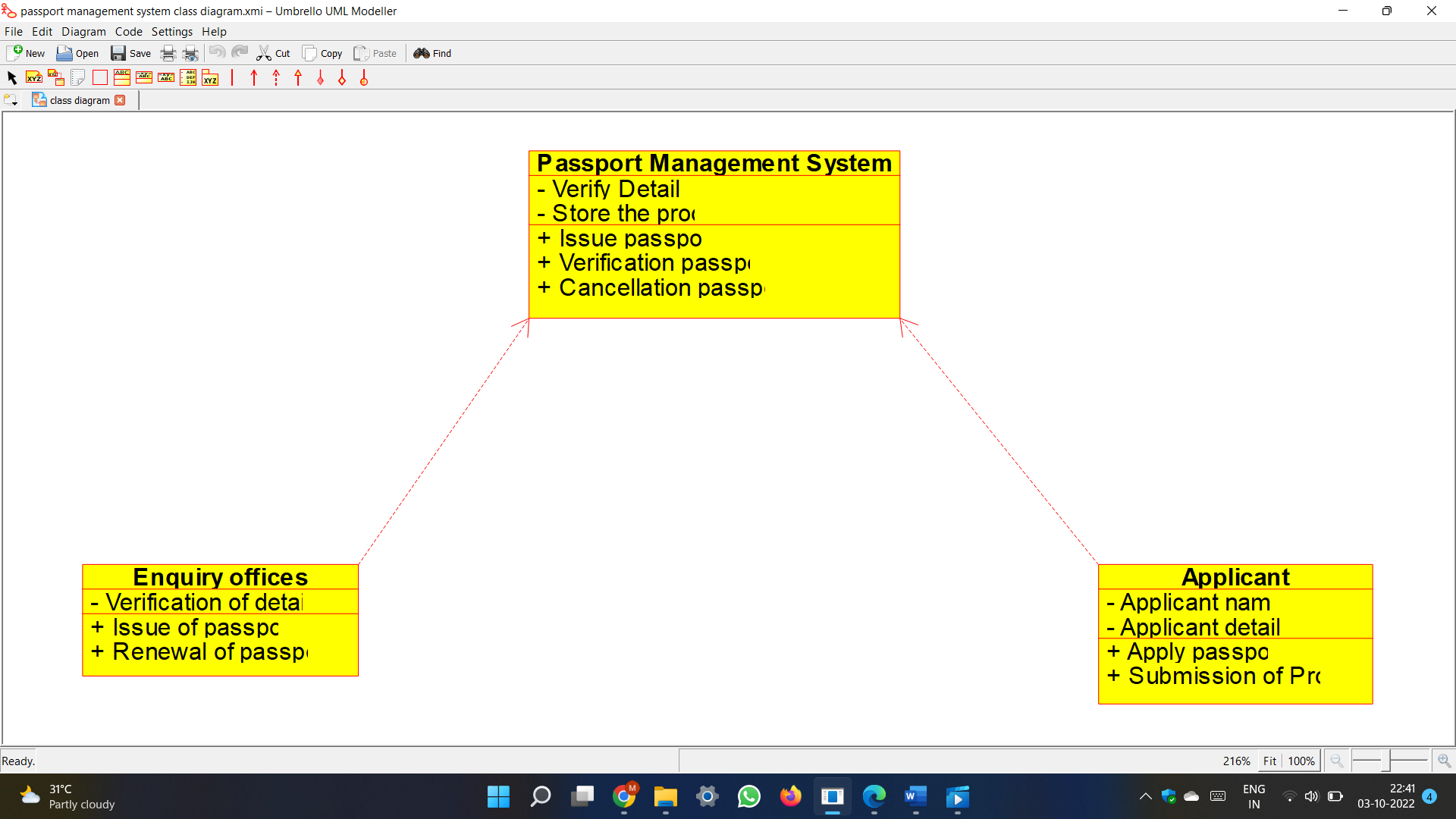
Passport Automation System

Use case :



Class diagram :



Code :

#ifndef PASSPORT\_MANAGEMENT\_SYSTEM\_H

#define PASSPORT\_MANAGEMENT\_SYSTEM\_H

#include <string>

/\*\*

\* class Passport\_Management\_System

\*

\*/

class Passport\_Management\_System

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

Passport\_Management\_System ();

/\*\*

\* Empty Destructor

\*/

virtual ~Passport\_Management\_System ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void Issue\_passport ()

{

}

/\*\*

\*/

void Verification\_passport ()

{

}

/\*\*

\*/

void Cancellation\_passport ()

{

}

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 Verify\_Details;

void Store\_the\_proof;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of Verify\_Details

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

\*/

void setVerify\_Details (void new\_var) {

Verify\_Details = new\_var;

}

/\*\*

\* Get the value of Verify\_Details

\* @return the value of Verify\_Details

\*/

void getVerify\_Details () {

return Verify\_Details;

}

/\*\*

\* Set the value of Store\_the\_proof

\* @param new\_var the new value of Store\_the\_proof

\*/

void setStore\_the\_proof (void new\_var) {

Store\_the\_proof = new\_var;

}

/\*\*

\* Get the value of Store\_the\_proof

\* @return the value of Store\_the\_proof

\*/

void getStore\_the\_proof () {

return Store\_the\_proof;

}

private:

void initAttributes () ;

};

#endif // PASSPORT\_MANAGEMENT\_SYSTEM\_H

#ifndef ENQUIRY\_OFFICES\_H

#define ENQUIRY\_OFFICES\_H

#include <string>

/\*\*

\* class Enquiry\_offices

\*

\*/

class Enquiry\_offices

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

Enquiry\_offices ();

/\*\*

\* Empty Destructor

\*/

virtual ~Enquiry\_offices ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void Issue\_of\_passport ()

{

}

/\*\*

\*/

void Renewal\_of\_passport ()

{

}

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 Verification\_of\_details;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of Verification\_of\_details

\* @param new\_var the new value of Verification\_of\_details

\*/

void setVerification\_of\_details (void new\_var) {

Verification\_of\_details = new\_var;

}

/\*\*

\* Get the value of Verification\_of\_details

\* @return the value of Verification\_of\_details

\*/

void getVerification\_of\_details () {

return Verification\_of\_details;

}

private:

void initAttributes () ;

};

#endif // ENQUIRY\_OFFICES\_H

#ifndef APPLICANT\_H

#define APPLICANT\_H

#include <string>

/\*\*

\* class Applicant

\*

\*/

class Applicant

{

public:

// Constructors/Destructors

//

/\*\*

\* Empty Constructor

\*/

Applicant ();

/\*\*

\* Empty Destructor

\*/

virtual ~Applicant ();

// Static Public attributes

//

// Public attributes

//

// Public attribute accessor methods

//

// Public attribute accessor methods

//

/\*\*

\*/

void Apply\_passport ()

{

}

/\*\*

\*/

void Submission\_of\_Proof ()

{

}

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 Applicant\_name;

void Applicant\_details;

public:

// Private attribute accessor methods

//

private:

public:

// Private attribute accessor methods

//

/\*\*

\* Set the value of Applicant\_name

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

\*/

void setApplicant\_name (void new\_var) {

Applicant\_name = new\_var;

}

/\*\*

\* Get the value of Applicant\_name

\* @return the value of Applicant\_name

\*/

void getApplicant\_name () {

return Applicant\_name;

}

/\*\*

\* Set the value of Applicant\_details

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

\*/

void setApplicant\_details (void new\_var) {

Applicant\_details = new\_var;

}

/\*\*

\* Get the value of Applicant\_details

\* @return the value of Applicant\_details

\*/

void getApplicant\_details () {

return Applicant\_details;

}

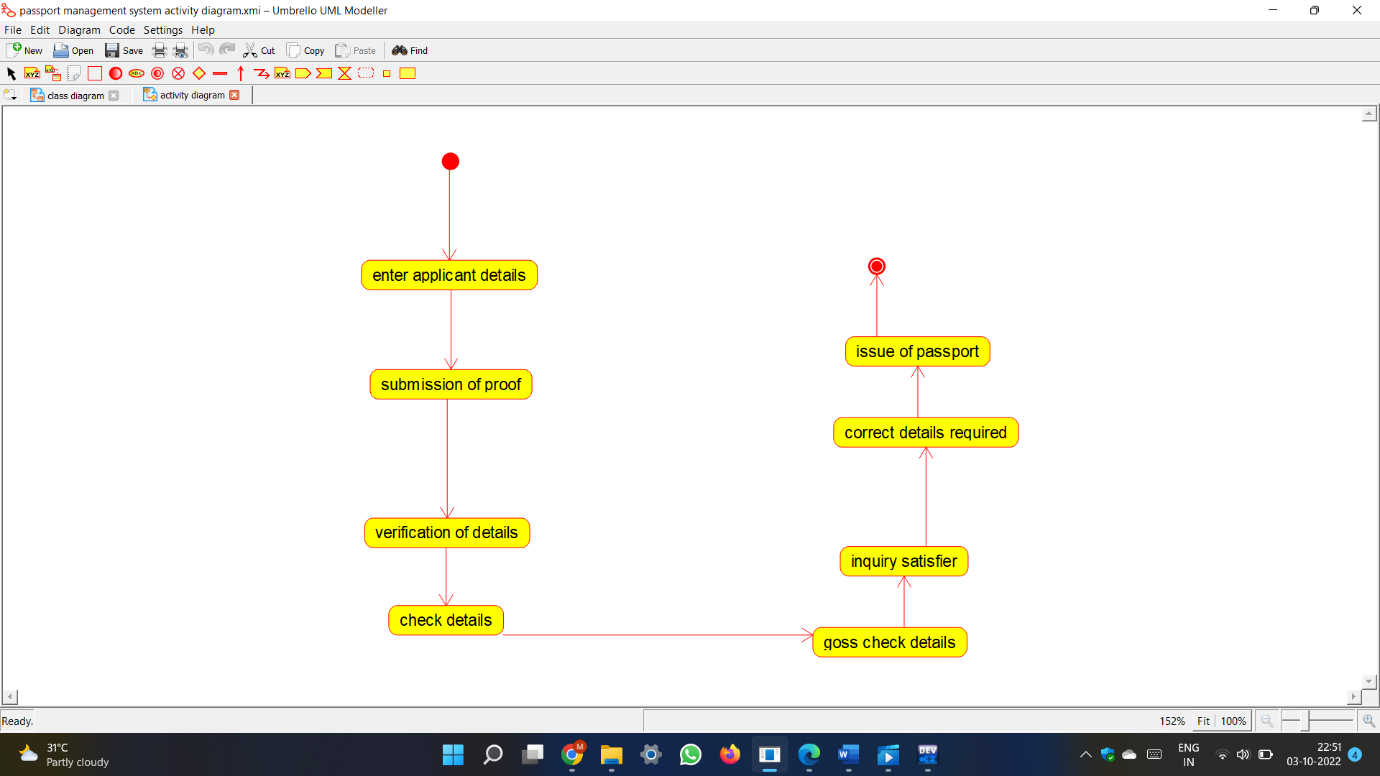
private:

void initAttributes () ;

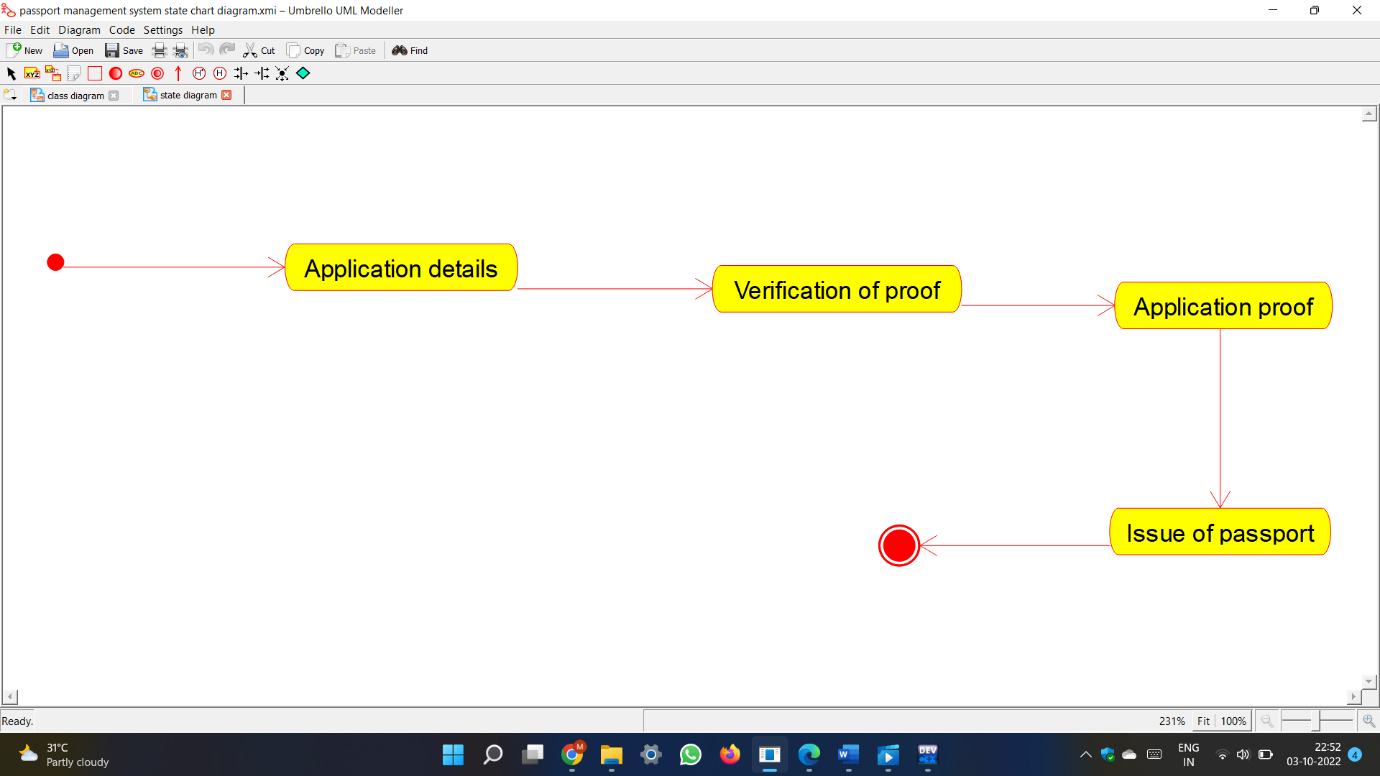
};

#endif // APPLICANT\_H

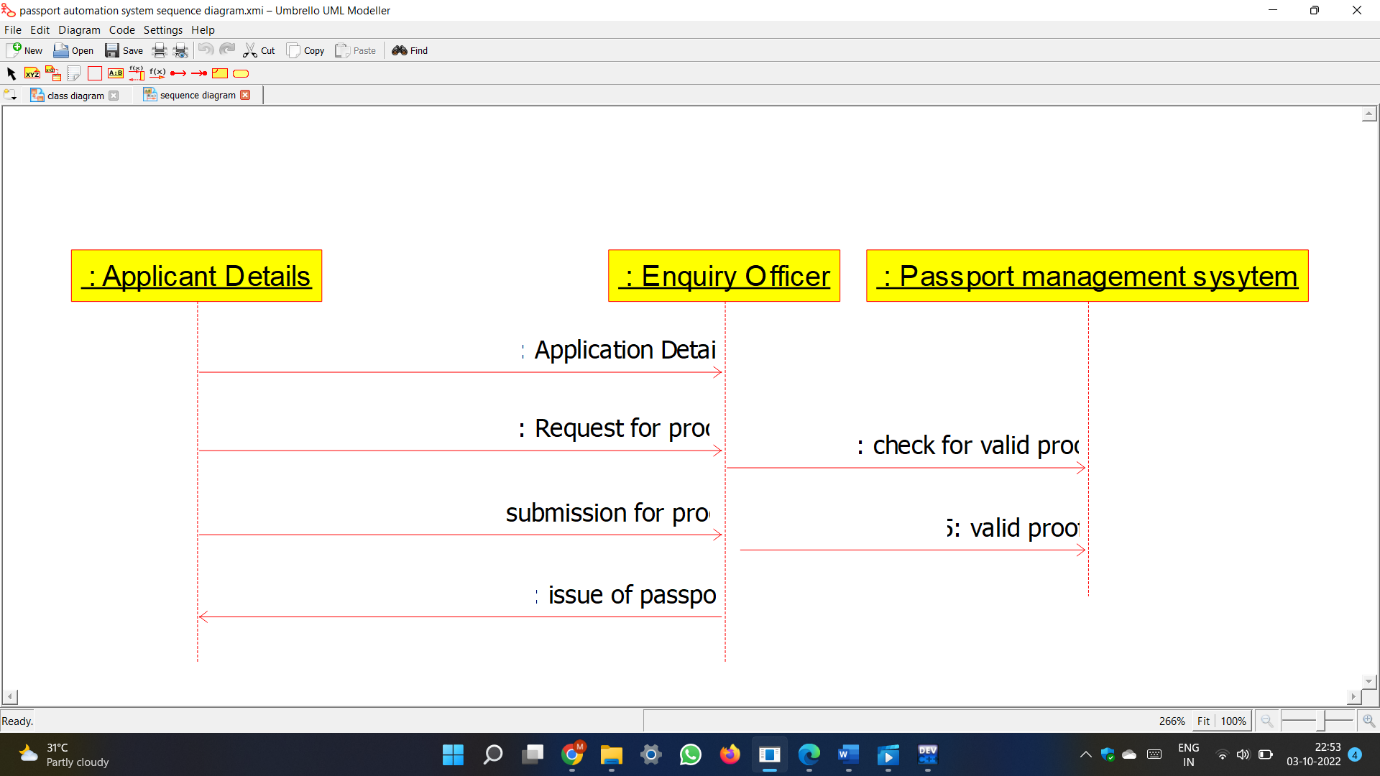
Activity diagram :



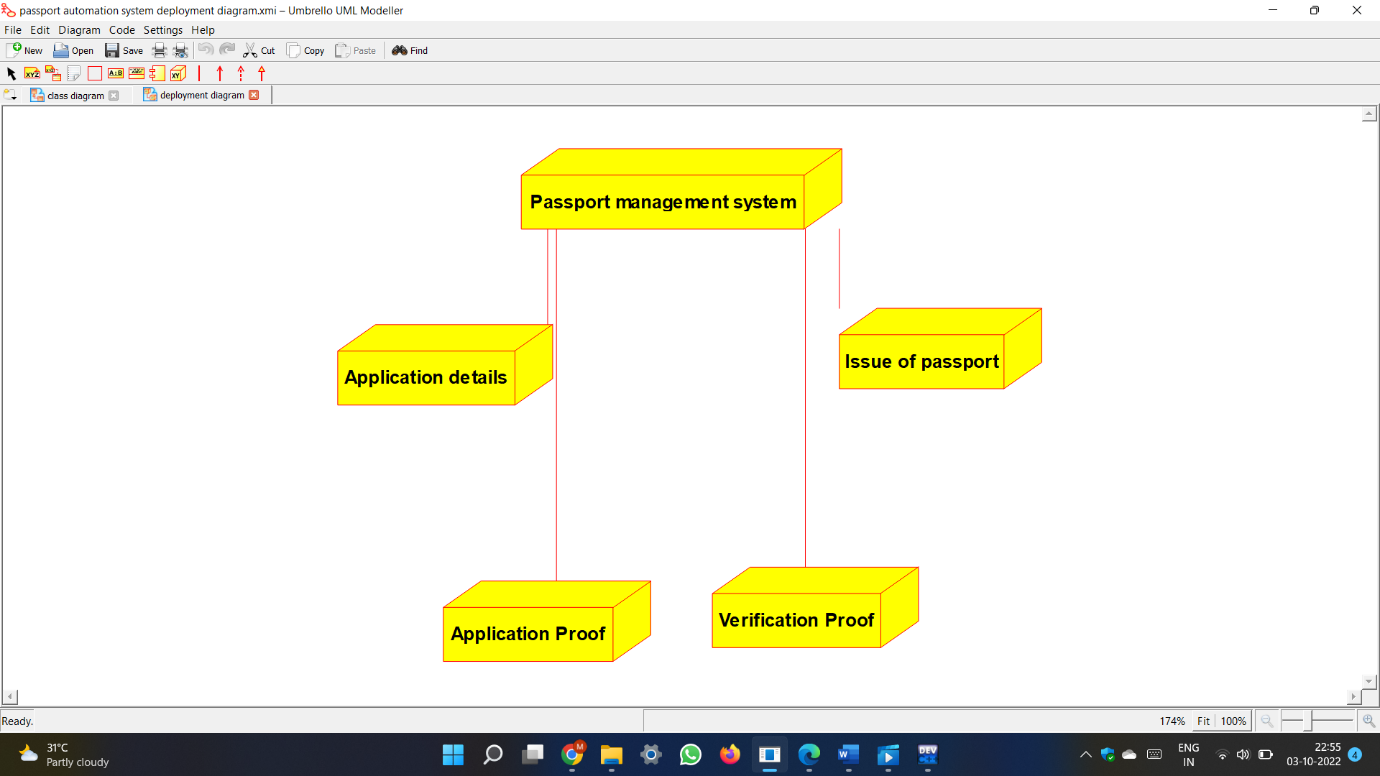
State chart diagram :



Sequence diagram :



Deployment diagram :



Component diagram :

