**Ex no:**

# FOREIGN TRADING SYSTEM

**Date:**

**AIM:**

To create a system to perform foreign trading system.

## (I) PROBLEM STATEMENT:

The main activity of international marketing is the export-import procedure. This procedure involves the actual and operational procedure of export and import trade. It also involves documentation, procedures, rules and regulations imposed by both the exporting and importing countries. These procedures include excise clearance, foreign exchange, etc.

## (II) SOFTWARE REQUIREMENTS SPECIFICATION:

### 1.0 INTRODUCTION

Foreign trading system is the interface between the exporter and buyer. It aims at improving the efficiency in the production, export process and reduce the complexities involved in it to the maximum possible extent.

### 1.1 PURPOSE

Considering the fact that the number of buyer is increasing every year, an Automated System becomes essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process. The system has been carefully verified and validated in order to satisfy it.

### 1.2 SCOPE

The System provides an online interface to the buyer where they can fill in their personal details and submit the necessary documents (may be by scanning). The authority concerned with the production and shipment of goods can use this system to reduce his workload and process the application in a speedy manner.

### 1.3 DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS

- **Exporter** – One who wishes to export the goods
- **Importer** - One who wishes to obtain the goods

- **Regional authorities –** Who provide service authority to individual and business firms intending to export and/or import goods
- **IEC Number –** Importer-Exporter Number
- **QC –** Quality Control in order to ensure the quality of products

### 1.4 REFERENCES

IEEE Software Requirement Specification format

### 1.5 TECHNOLOGIES TO BE USED

- HTML
- JSP
- Java script
- Java

### 1.6 TOOLS TO BE USED

- Eclipse IDE (Integrated Development Environment)
- Rational Rose tool (for developing UML Patterns)

### 1.7 OVERVIEW

SRS includes two sections overall description and specific requirements.

**Overall description** will describe major role of the system components and inter-connections.

**Specific requirements** will describe roles & functions of the actors.

## 2.0 OVERALL DESCRIPTION:

It will describe major role of the system components and inter-connections.

### 2.1 PRODUCT PERSPECTIVE

This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time.

### 2.2 SOFTWARE INTERFACE

- **Front End Client** - The exporter online interface is built using JSP and HTML.
- **Web Server** – Apache Tomcat Server (Oracle Corporation)
- **Back End** - Oracle 11g database

## 2.3 HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

## (III)USE-CASE DIAGRAM:

The book bank use cases are:

1. Preliminaries
2. Offer order
3. Production of goods
4. Shipment
5. Negotiation of documents

## ACTORS INVOLVED:

1. Exporter
2. Buyer

## USECASE NAME : PRELIMINARIES

Individual and business firms intending to export and/or import goods and/or services should obtain an Importer-Exporter Number from the regional licensing authorities.

## USECASE NAME : OFFER ORDER

Offer is a proposal submitted by an exporter expressing his intention to export specific goods at a specific price with specific terms and conditions. Exporter usually makes an offer in the form of a 'Proforma Invoice'.

## USECASE NAME : PRODUCTION OF GOODS

The exporting house after obtaining a confirmed order should produce the goods exactly as specified in the invoice. If the exporting house does not have facilities, it has to procure the products from others.

## USECASE NAME : SHIPMENT

The exporter transports the goods to the buyer.

**USECASE NAME : NEGOTIATION OF DOCUMENTS**

The exporter submits the relevant documents to his buyer (banker) for getting the payment for the goods exported.

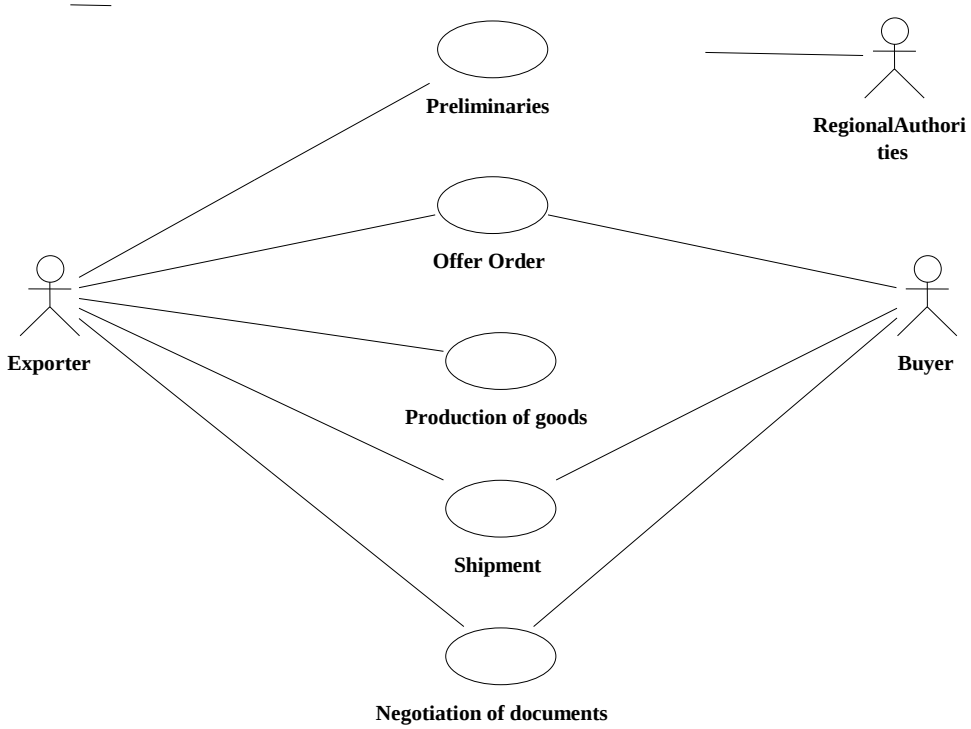


**Fig 3.1 USE-CASE DIAGRAM FOR BOOK BANK SYSTEM**

## (IV) ACTIVITY DIAGRAM:

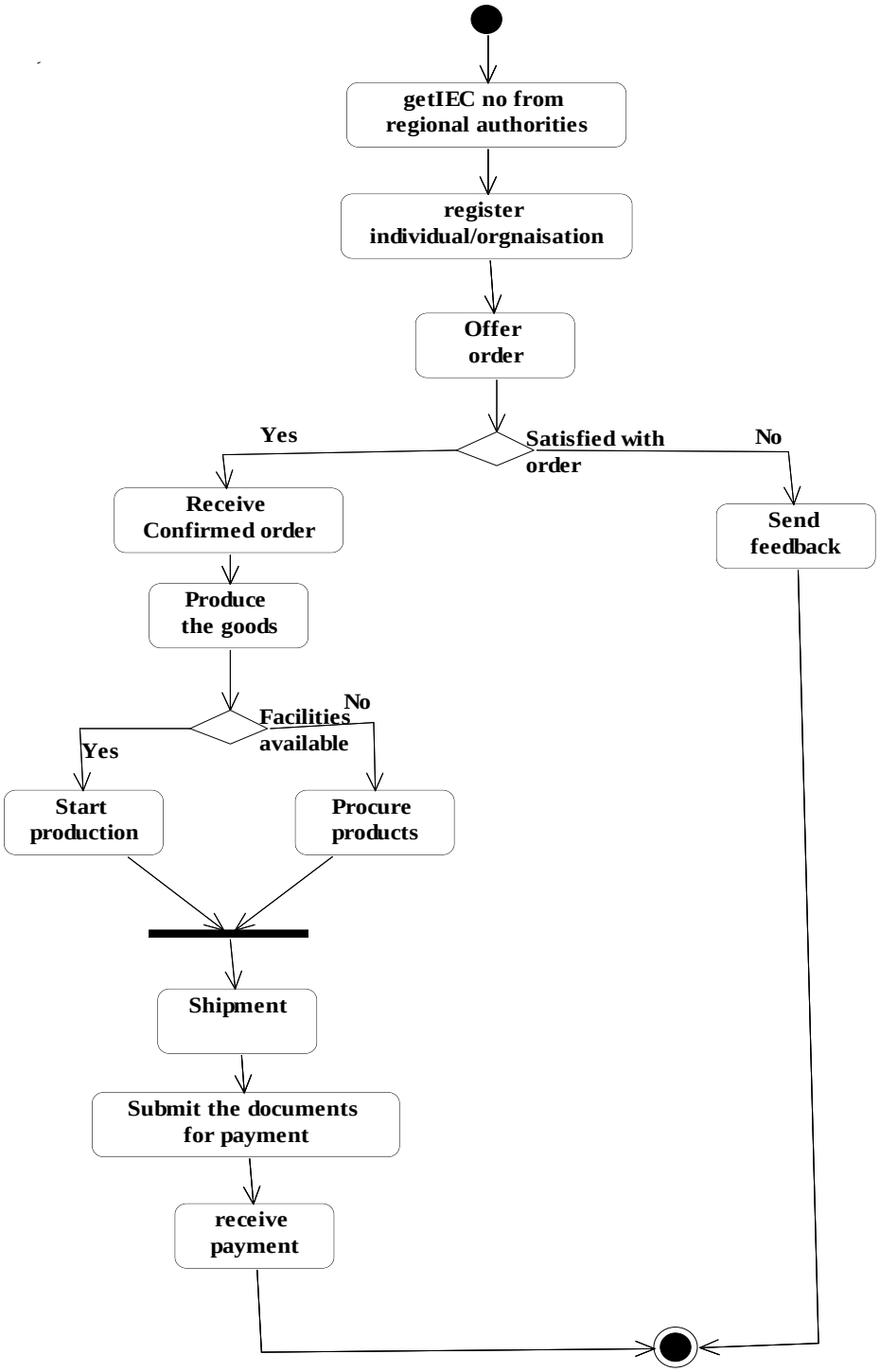


**Fig.4.1 ACTIVITY DIAGRAM**

An activity diagram is a variation or special case of a state machine in which the states or activity representing the performance of operation and transitions are triggered by the completion of operation.

The purpose is to provide view of close and what is going on inside a use case or among several classes. An activity is shown as rounded box containing the name of operation.

**DOCUMENTATION OF ACTIVITY DIAGRAM**

- Perform preliminaries activities ie, getting IEC number from regional licensing authorities
- Submit a proposal order to the buyer
- After obtaining a confirmed order should produce the goods exactly as specifies in the invoice.
- If the exporting house does not have production facilities, it has to procure the products from others.
- Transport the goods to the buyer.
- The exporter submits the relevant documents to his buyer (banker) for getting the payment for the goods exported.

**(V) CLASS DIAGRAM:**

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

The ATM system class diagram consists of four classes:

1. ExporterInfo
2. OfferOrder
3. Production
4. BuyerInfo

**1) EXPORTERINFO:**

It consists of six attributes and two operations. The attributes are exportername, IEC_No, Reg_no, address, contactno, and Email_id. The operations of this class are getIECNo() and getReg().

**2) OFFERORDER:**

This class includes the following items:

Buyer name, order_no, description of goods, price, condition of sale, payment terms (credit, bill of exchange, etc,)

**3) PRODUCTION:**

It consists of packing and marking process, quality control process, shipment and submission of documents.

**4) BUYERINFO:**

This class is used to maintain the buyer information such as buyer name, buyer address, etc.

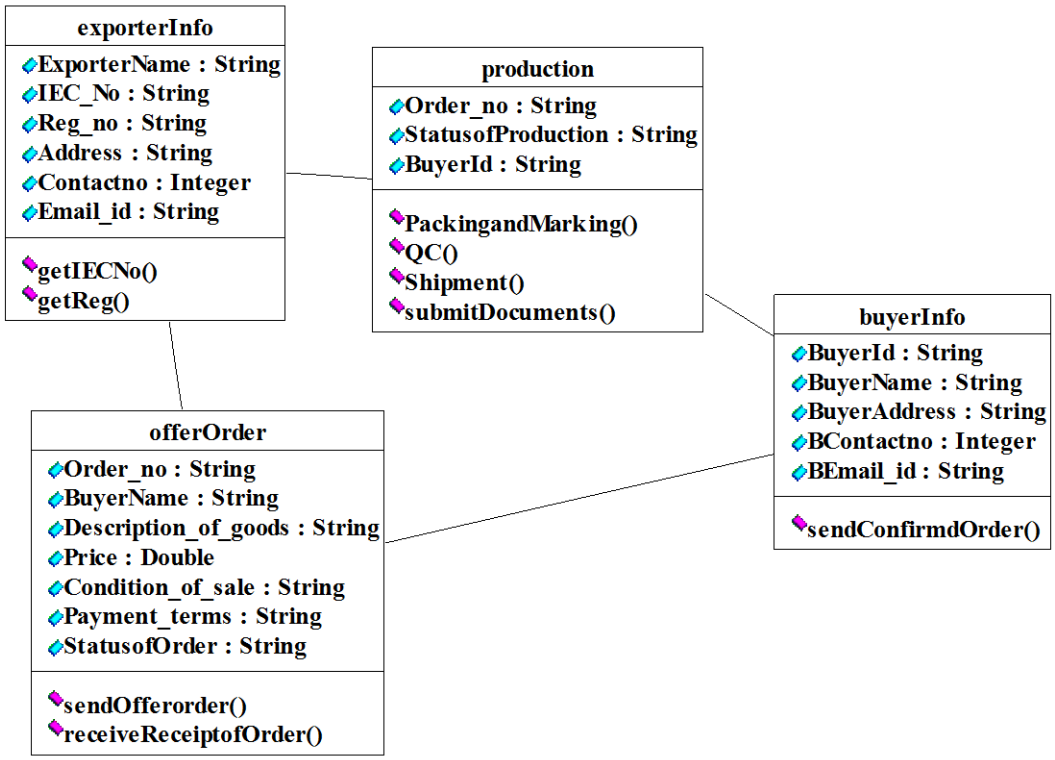


**Fig.5.1 CLASS DIAGRAM FOR FOREIGN TRADING SYSTEM**

## (VI) SEQUENCE DIAGRAM:

A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another, in which the "form" object is requesting an operation be performed by the "to" object. The "to" object performs the operation using a method that the class contains.

It is also represented by the order in which things occur and how the objects in the system send message to one another.

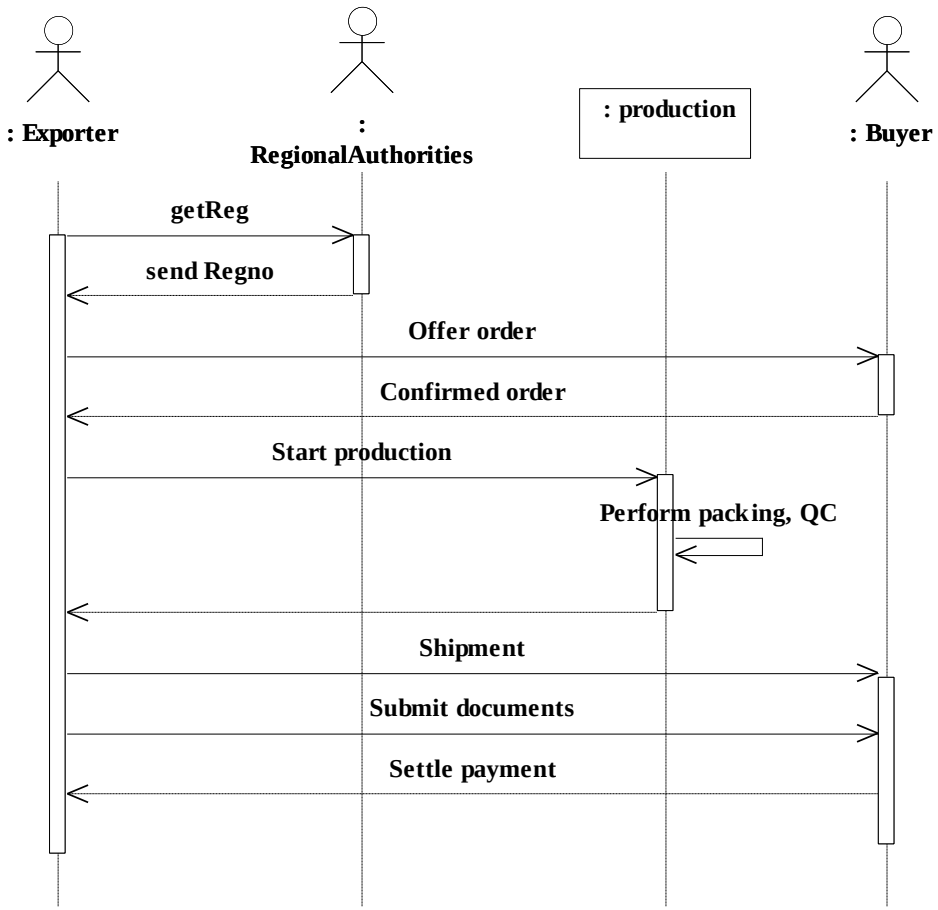


**Fig. 6.1. SEQUENCE DIAGRAM FOR FOREIGN TRADING SYSTEM**

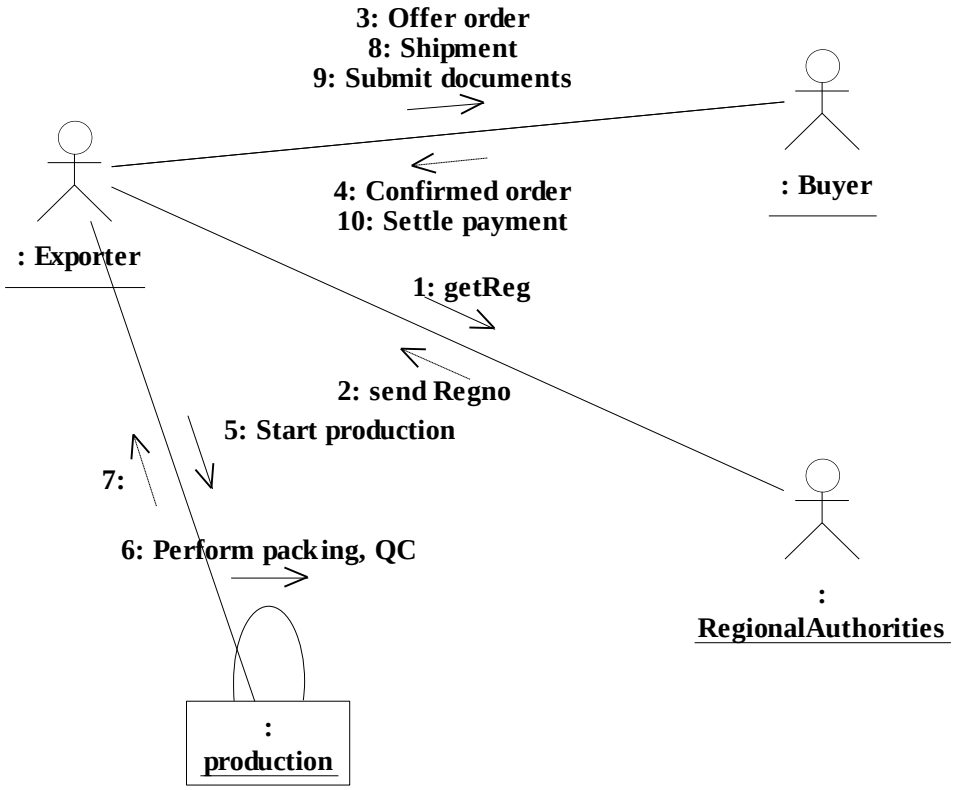


**Fig. 6.2. COLLABORATION DIAGRAM FOR FOREIGN TRADING SYSTEM**

## (VII) DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

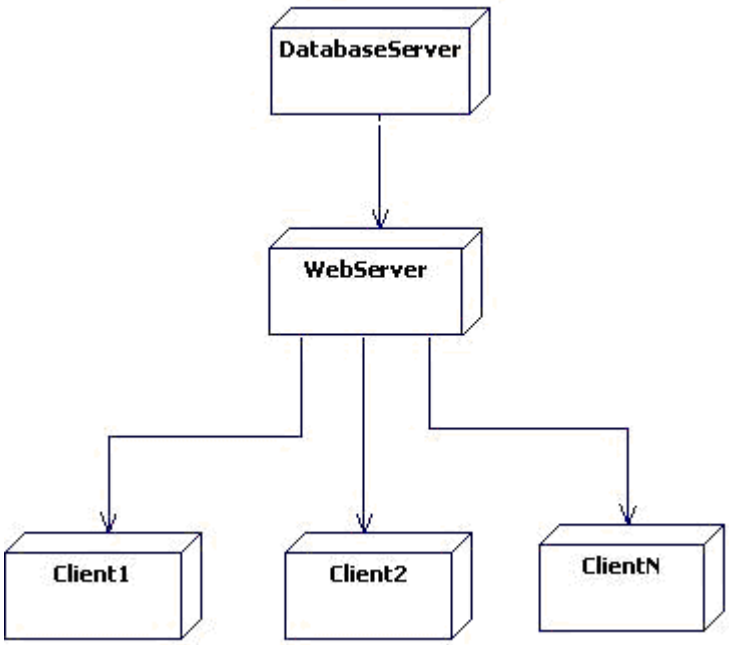


**Fig.7.1.DEPLOYMENT DIAGRAM**

## (VIII) IMPLEMENTATION OF DOMAIN OBJECTS LAYER AND TECHNICAL SERVICE LAYER

***//Source file: F:\\vaish\\BuyerInfo.java***

```
public class BuyerInfo
{
  private String BuyerId;
  private String BuyerName;
  private String BuyerAddress;
  private nteger BcontactNo;
  private String BEmailId;
  public Production theProduction;
```

```
   /**
   @roseuid 515917A5029F
    */
   public BuyerInfo()
   {

   }

   /**
   @roseuid 515916D502DE
    */
   public void SendConfirmedOrder()
   {

   }
}
```

***//Source file: F:\\vaish\\exporterInfo.java***

```
public class exporterInfo
{
   private String ExporterName;
   private String IEC_no;
   private String Reg_no;
   private String Address;
   private Integer ContactNo;
   private String Email_id;
   public OfferOrder theOfferOrder;

   /**
   @roseuid 515917A50213
    */
   public exporterInfo()
   {

   }

   /**
   @roseuid 51591524006D
    */
   public void getIECNo()
   {

   }

   /**
   @roseuid 51591530035B
    */
   public void getReg()
```

```
  {

  }
}
```

***//Source file: F:\\vaish\\OfferOrder.java***

```
public class OfferOrder
{
  private String Order_No;
  private String BuyerName;
  private String DescriptionOfGoods;
  private Double Price;
  private String ConditionOfSale;
  private String Payment_Terms;
  private String StatusOfOrder;
  public exporterInfo theExporterInfo;

  /**
  @roseuid 515917A502DE
   */
  public OfferOrder()
  {

  }

  /**
  @roseuid 515916510213
   */
  public void SendOfferOrder()
  {

  }

  /**
  @roseuid 5159165A009C
   */
  public void RecieveRecieptOfOrder()
  {

  }
}
```

***//Source file: F:\\vaish\\Production.java***

```
public class Production
{
  private String Order_no;
  private String StatusOfProduction;
  private String BuyerId;
```

```
  public BuyerInfo theBuyerInfo;

  /**
  @roseuid 515917A50261
   */
  public Production()
  {

  }

  /**
  @roseuid 5159157C0186
   */
  public void PackingAndMaking()
  {

  }

  /**
  @roseuid 5159158C0261
   */
  public void QC()
  {

  }

  /**
  @roseuid 5159158F037A
   */
  public void Shipment()
  {

  }

  /**
  @roseuid 5159159602BF
   */
  public void SubmitDocuments()
  {

  }
}
```

## RESULT:

Thus the mini project for Book Bank System has been successfully executed and codes are generated.