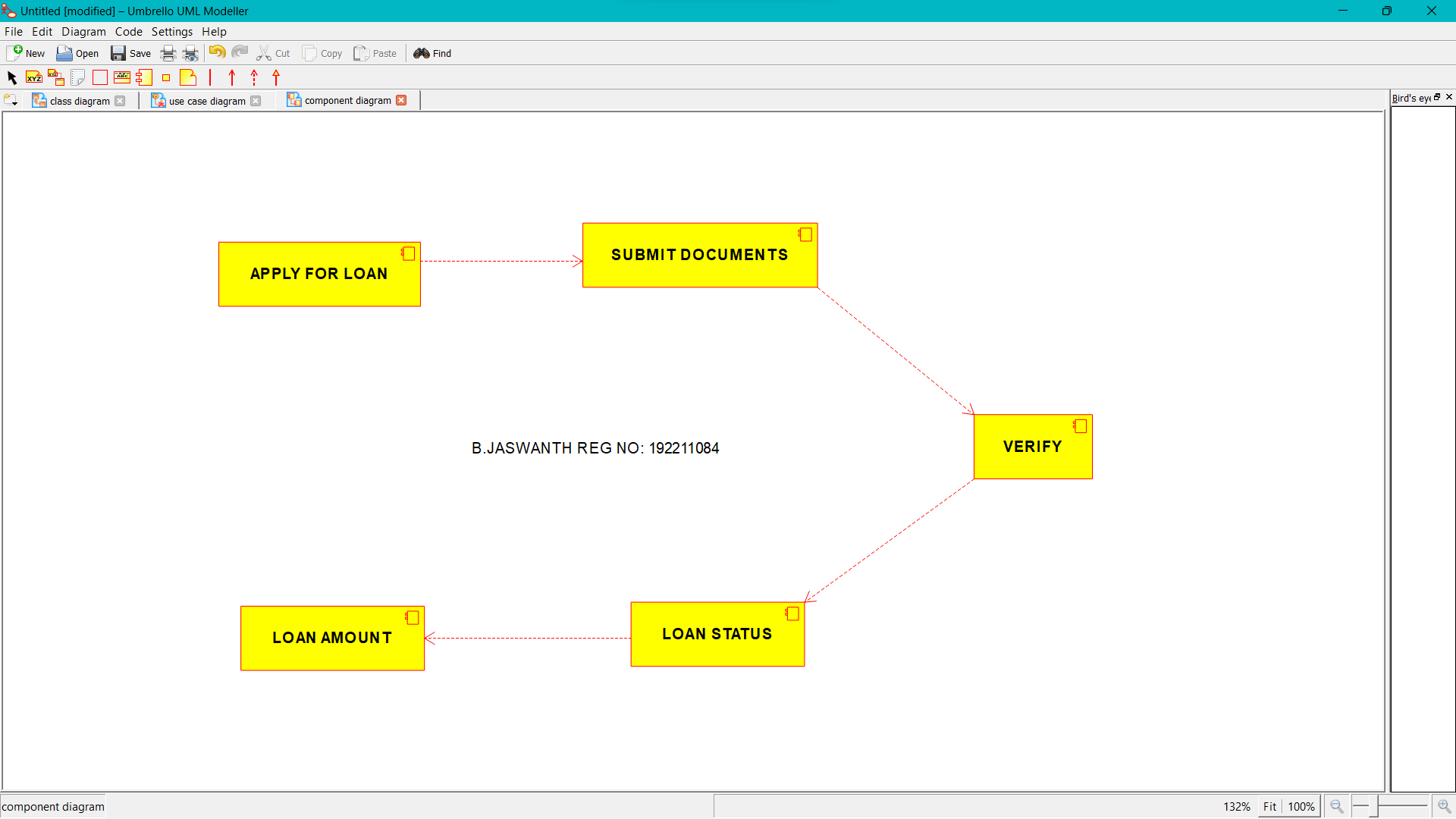
OBJECT ORIENTED ANALYSIS AND DESIGN IN

INDUSTRIAL PERSPECTIVE

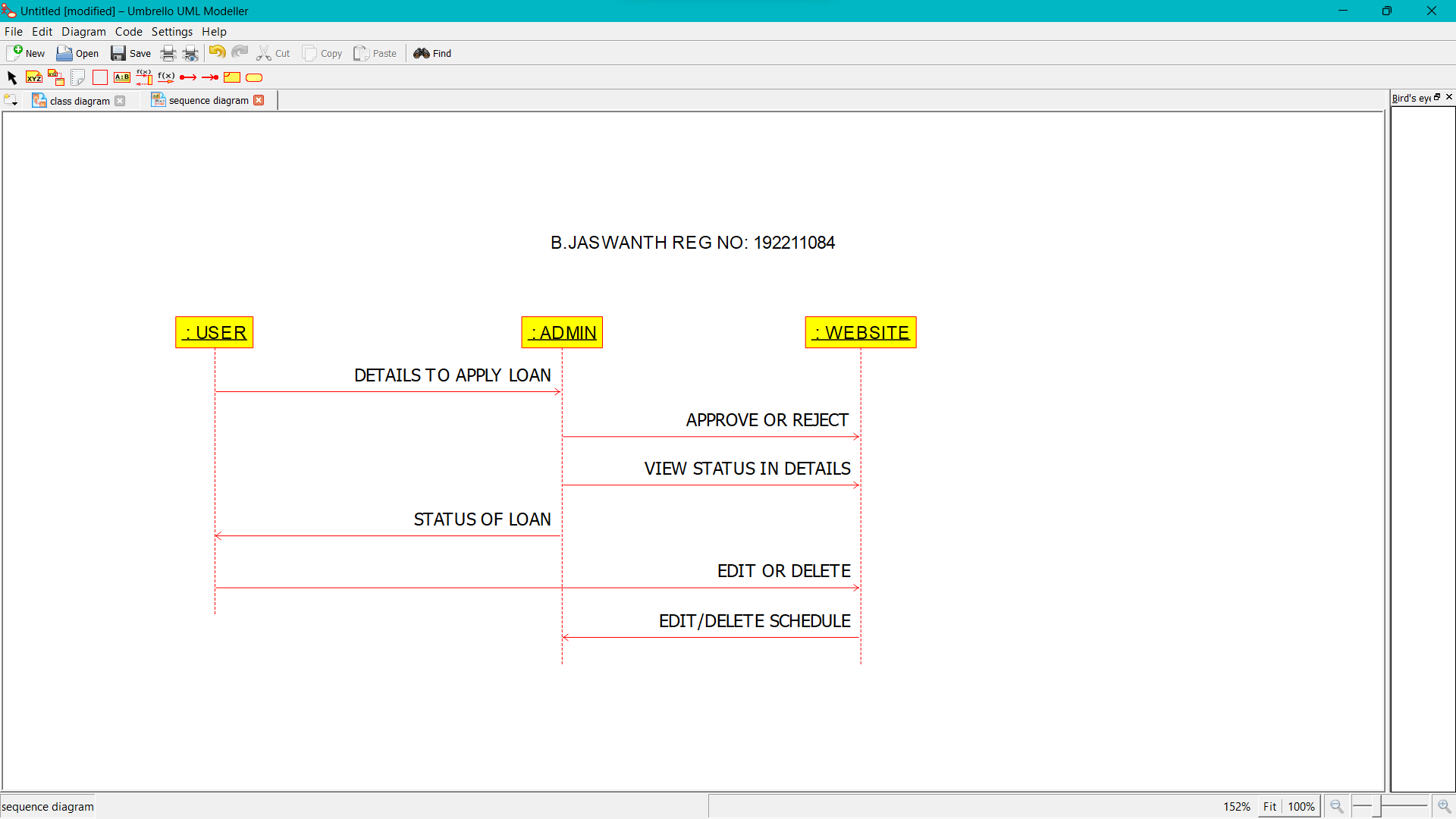
SLOT-D LAB PRACTICALS DAY-4

31. Develop a system using UML for Agriculture Loan Management System. The admin should add new customers and view his/her records. The admin should check the defaulters. The customer can see his/her account detail and apply for a new loan by providing the required documents. The admin should verify the document submitted and approve the loan after verifying the payback potential of the customer.

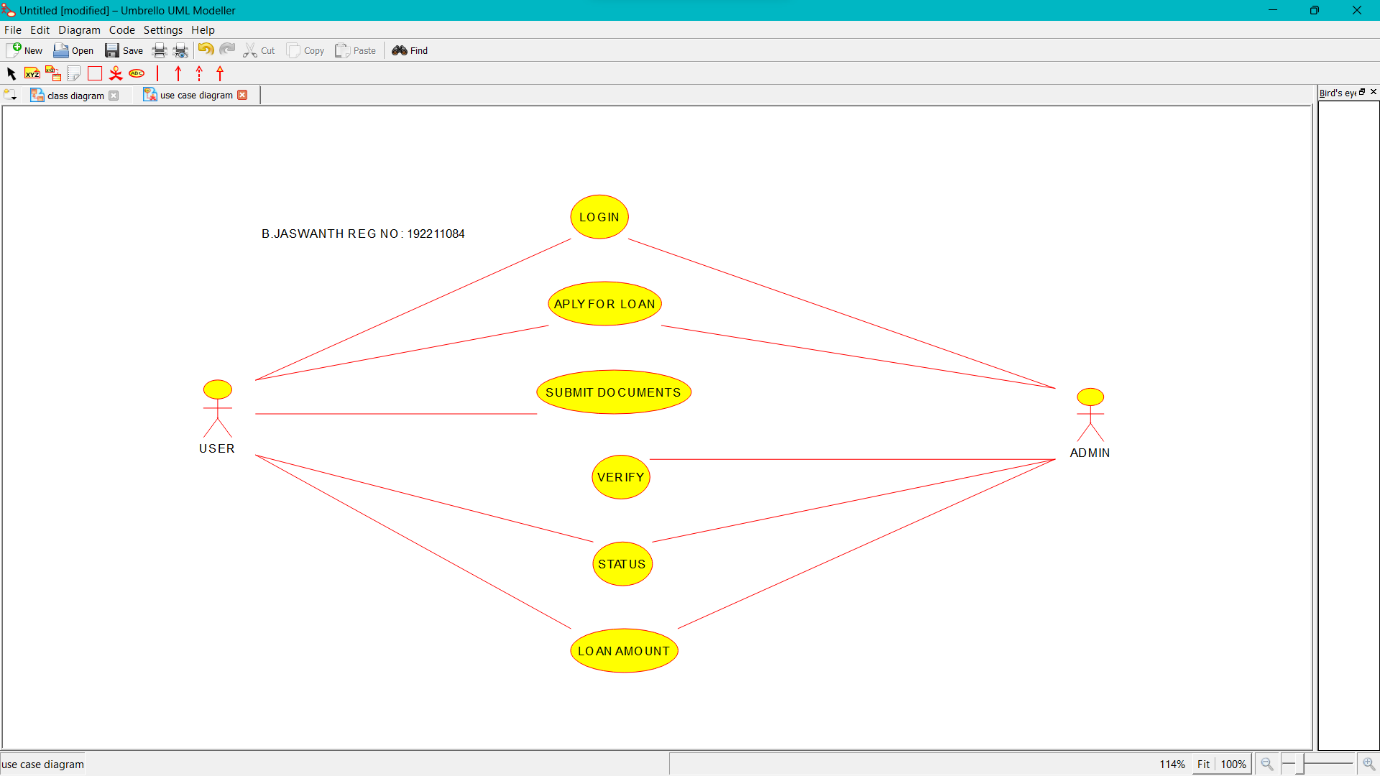
**COMPONENT DIAGRAM**



**SEQUENCE DIAGRAM**

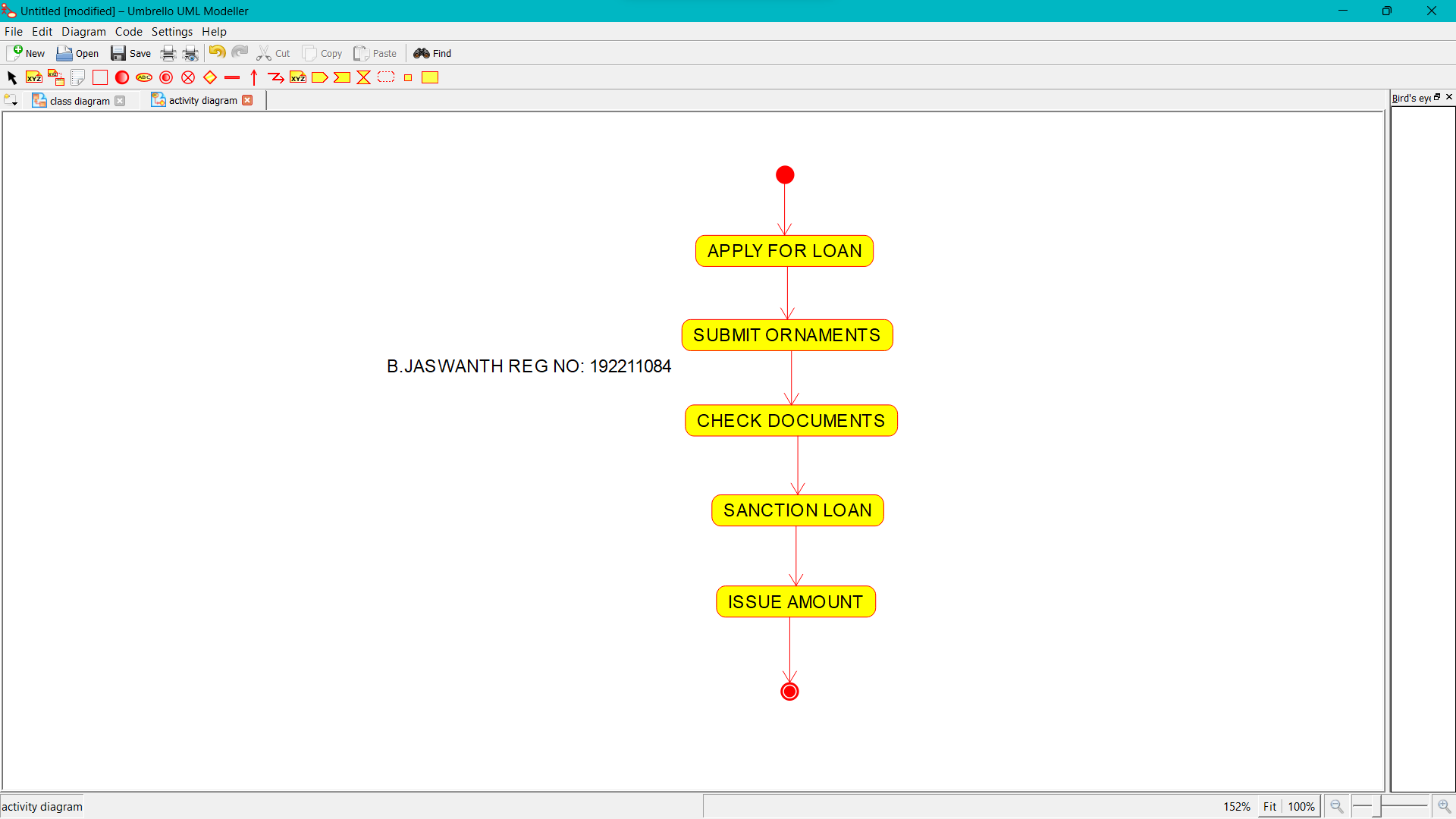


**USE CASE DIAGRAM**

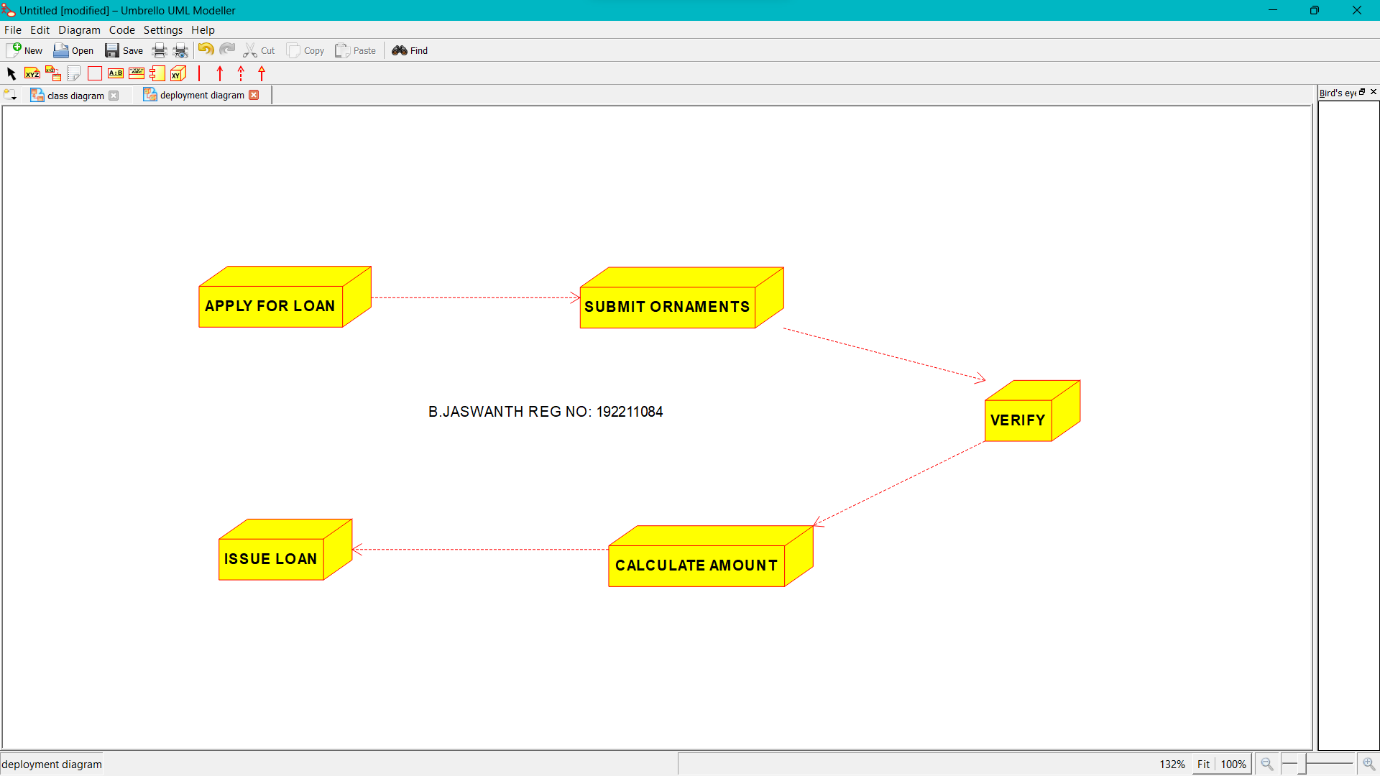


32. Develop a system using UML for Jewel Loan Management System. The admin should add new customers and view the customers' records. The admin should check the defaulters. The customer can see his/her account detail and apply for a new loan by providing the required ornaments. The admin should verify the purity of the ornaments submitted and approve the loan after verifying the payback potential of the customer.

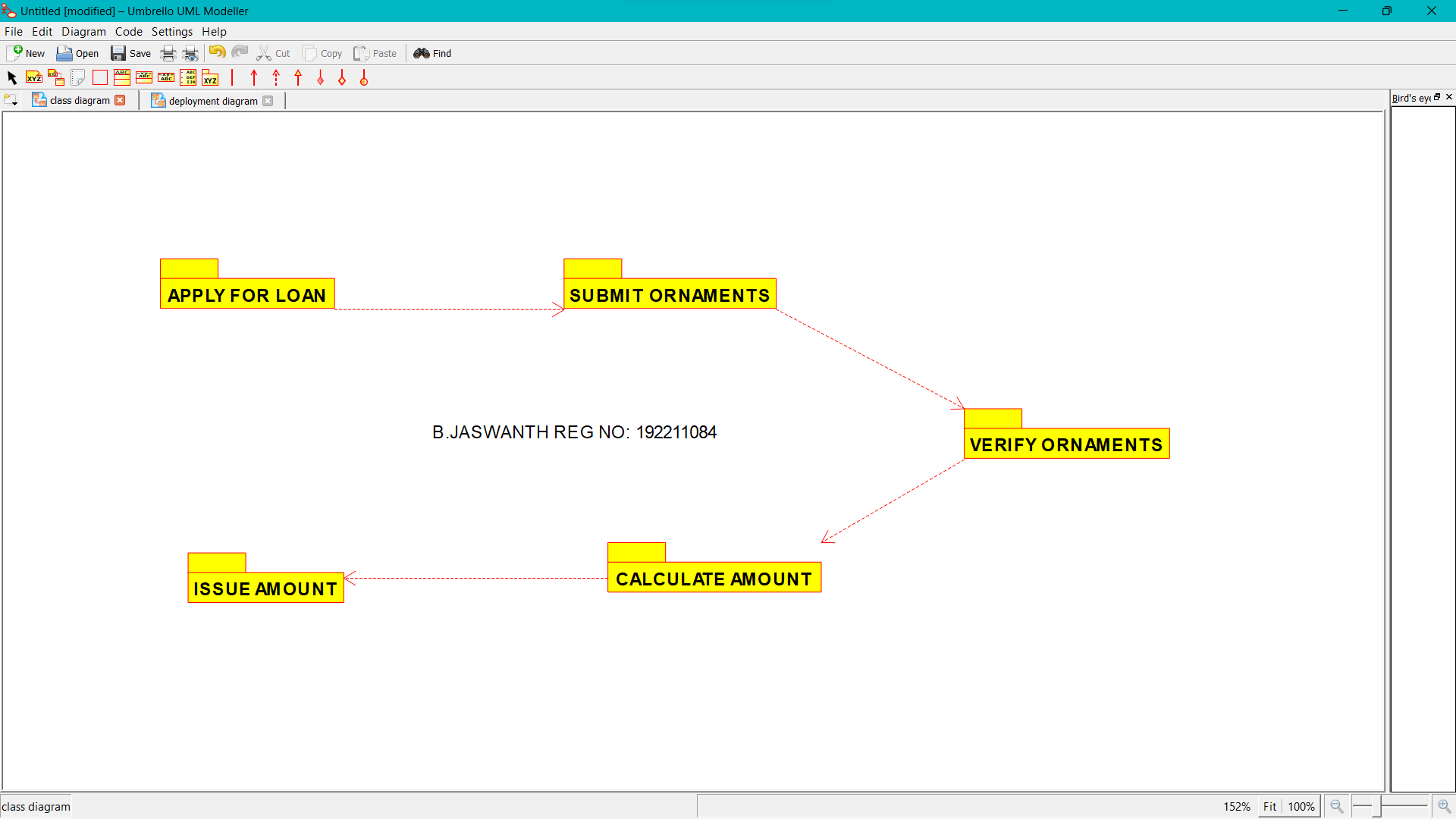
**ACTIVITY DIAGRAM**



**DEPLOYMENT DIAGRAM**

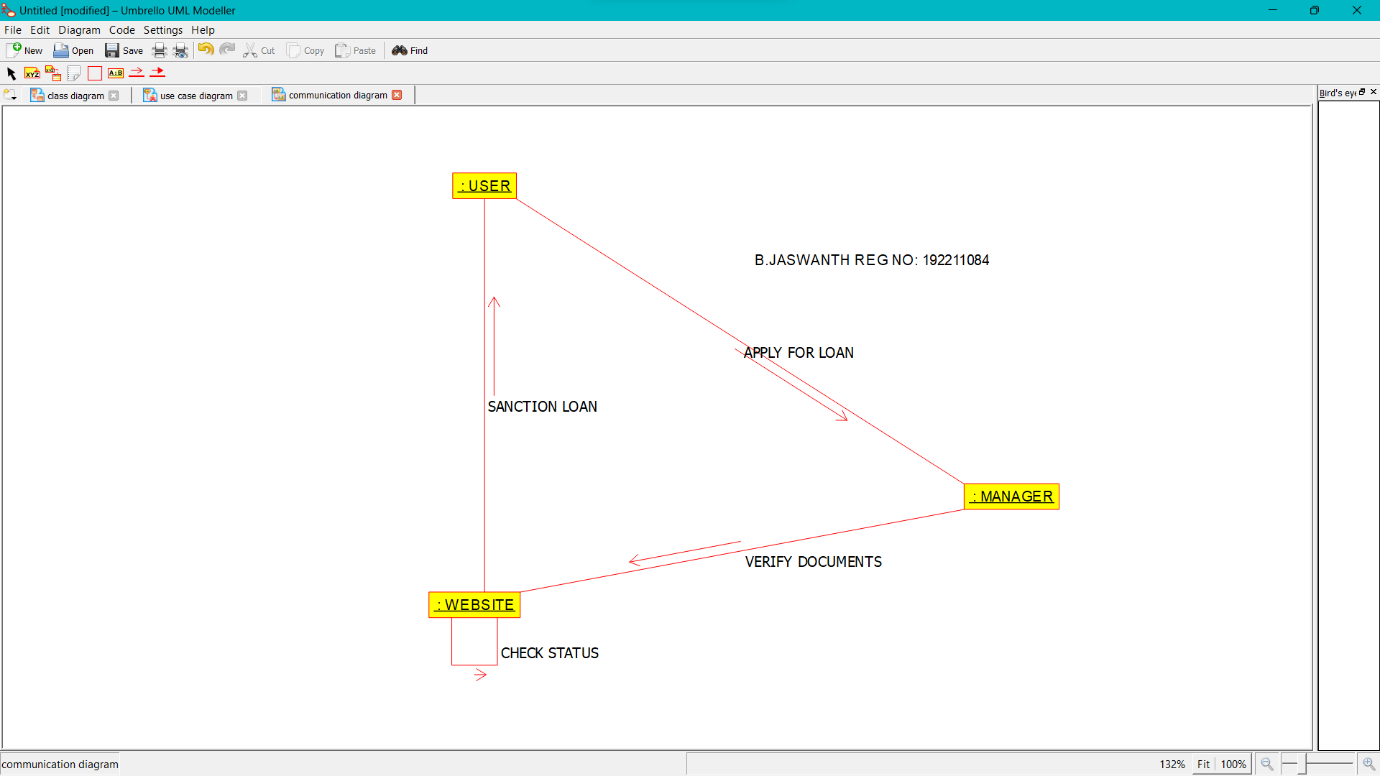


**PACKAGE DIAGRAM**

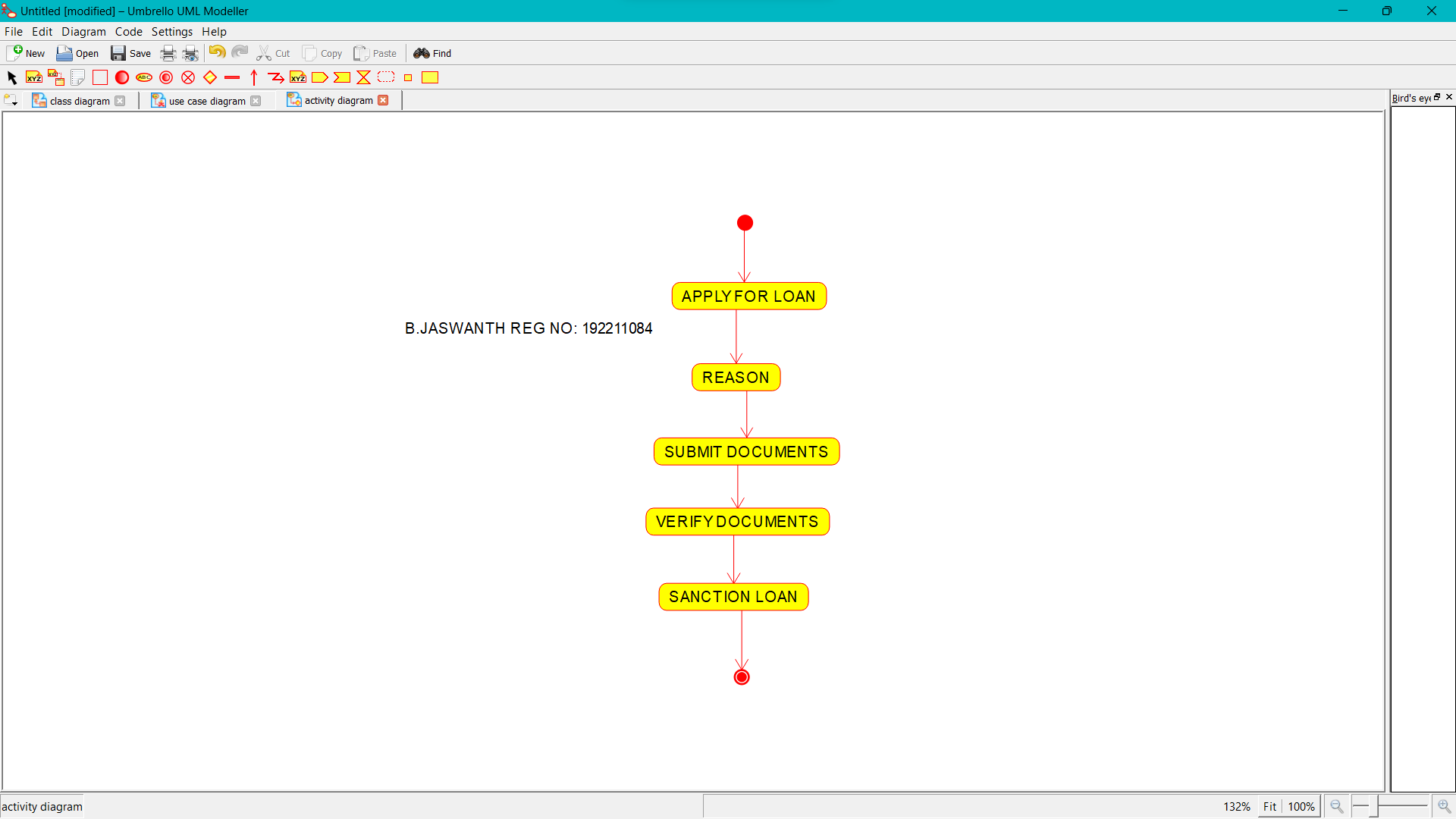


33. Develop a system using UML for Personal Loan Management System. The admin should add new customers and view the customer's records. The admin should check the defaulters. The customer can see his/her account detail and apply for a new loan by providing the required documents, and payslip. The admin should verify the document submitted and approve the loan after verifying the payback potential of the customer.

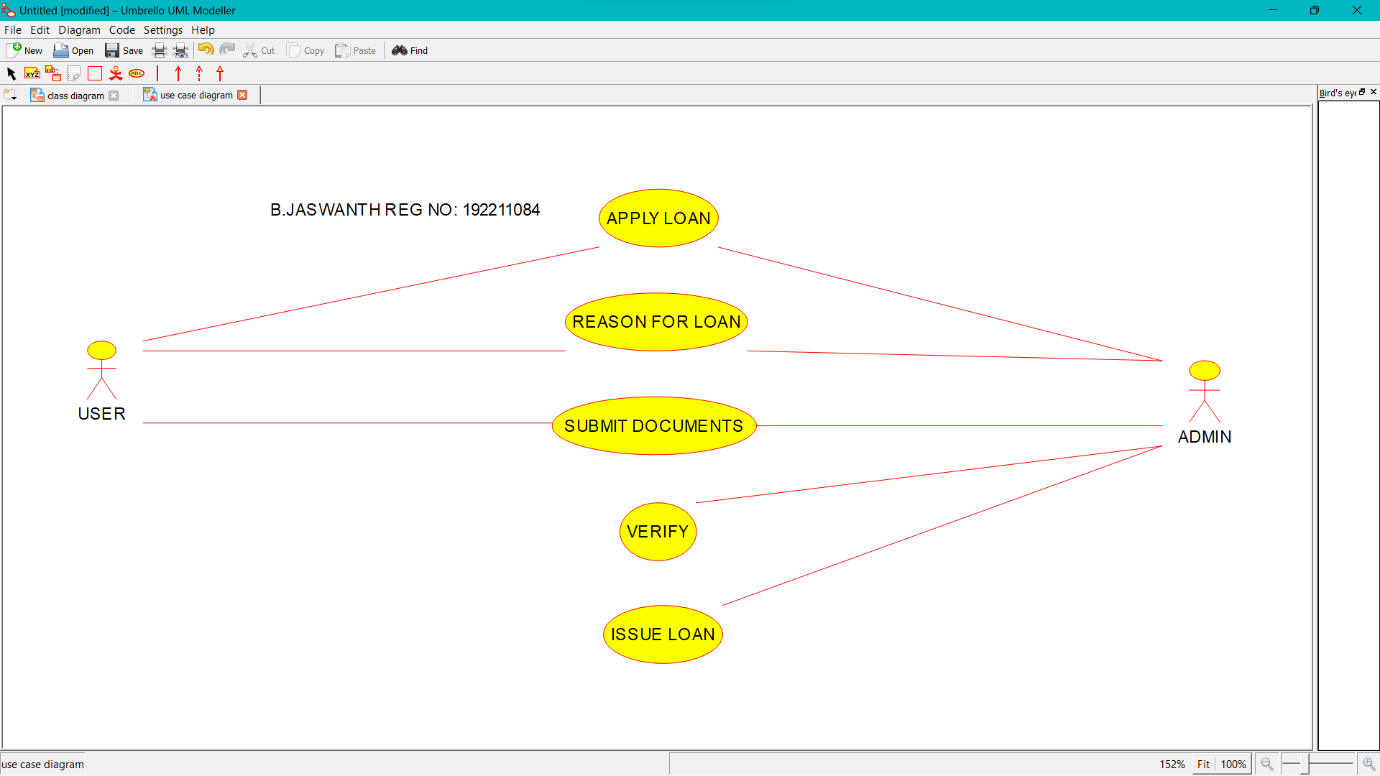
**COLLABORATION DIAGRAM**



**STATE CHART DIAGRAM**

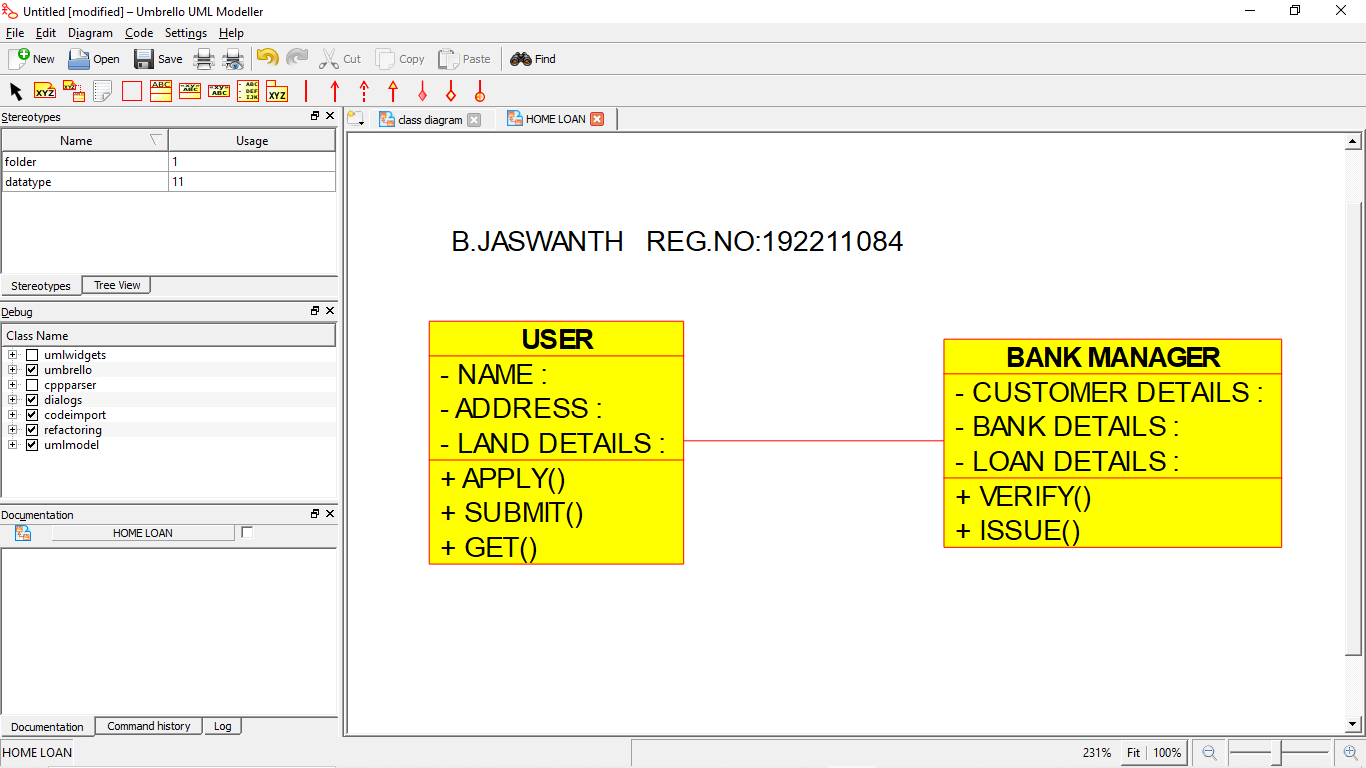


**USE CASE DIAGRAM**

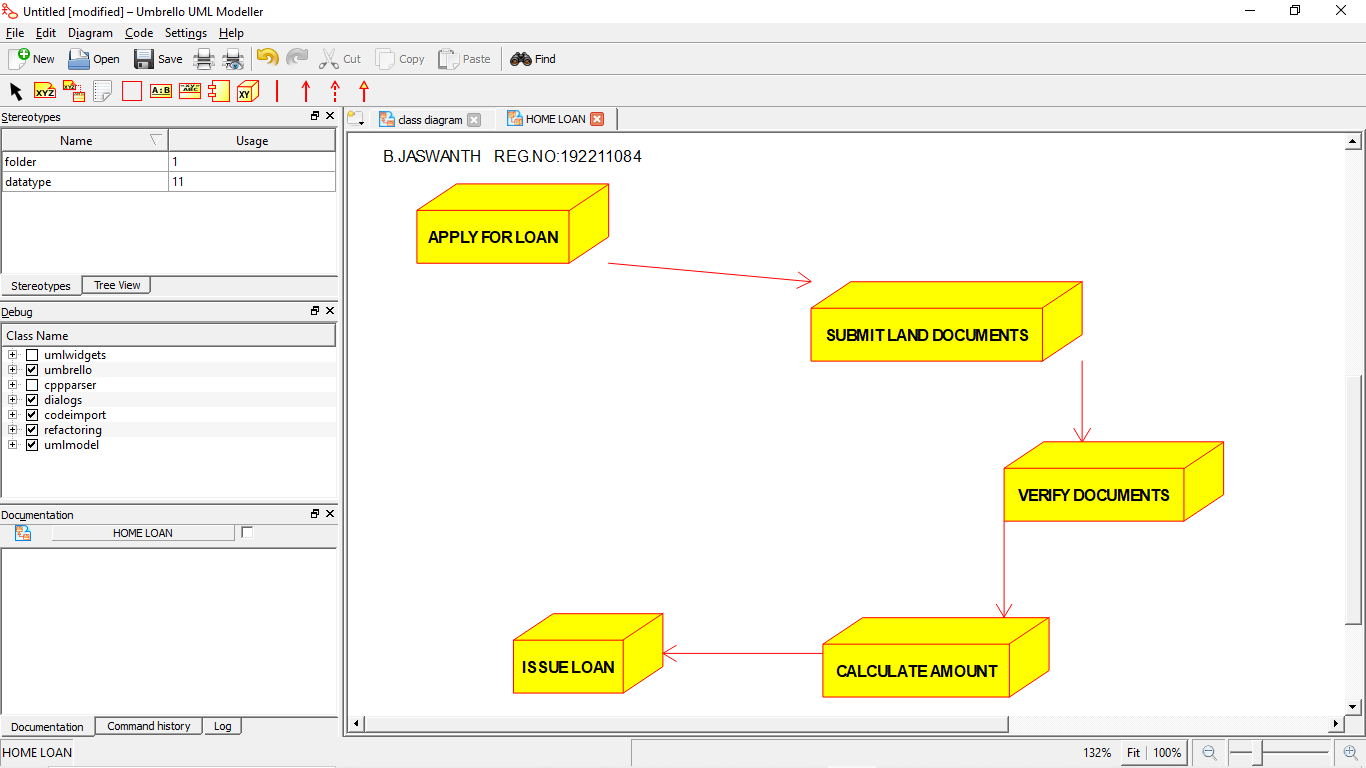


34. Develop a system using UML for Home Loan Management System. The admin should add new customers and view the customer's records. The admin should check the defaulters. The customer can see his/her account detail and apply for a new loan by providing the required documents. The admin should verify the document submitted and approve the loan after verifying the payback potential of the customer.

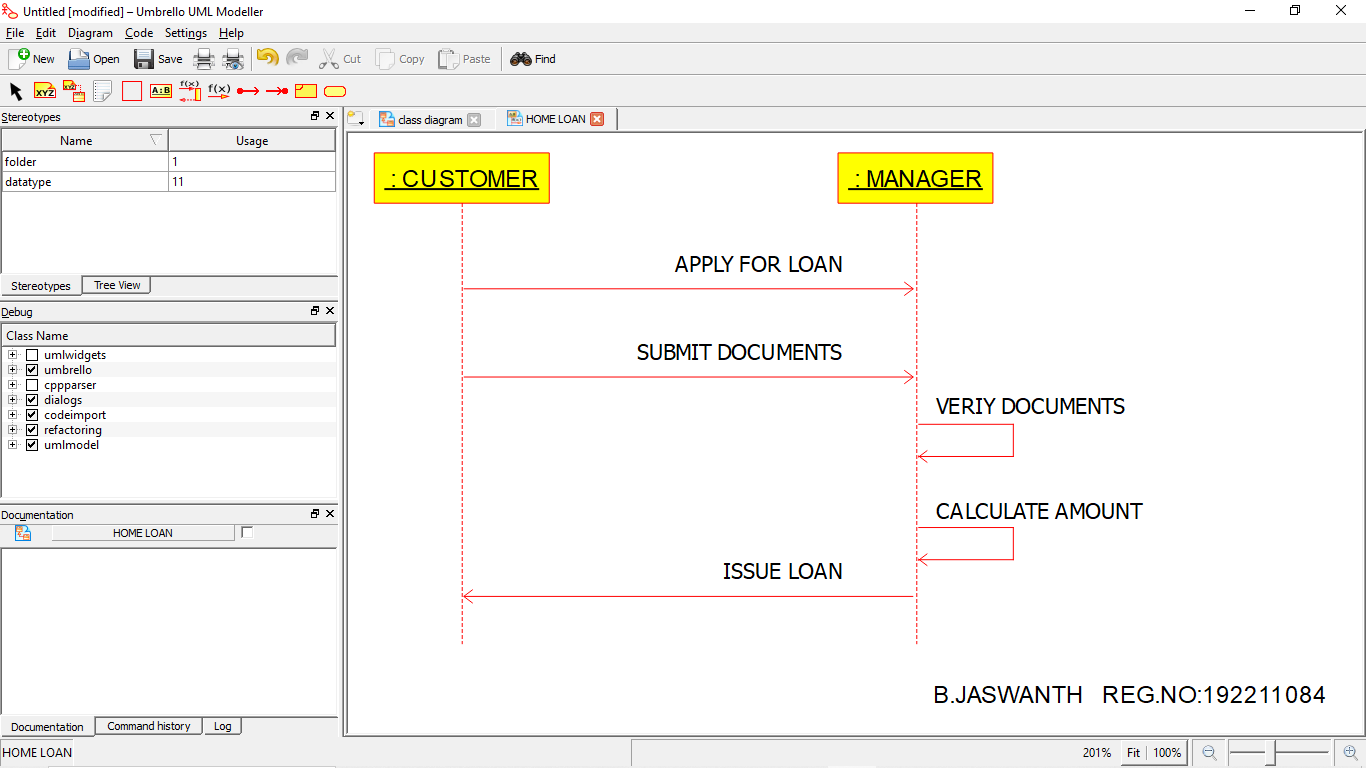
**CLASS DIAGRAM**



**DEPLOYMENT DIAGRAM**

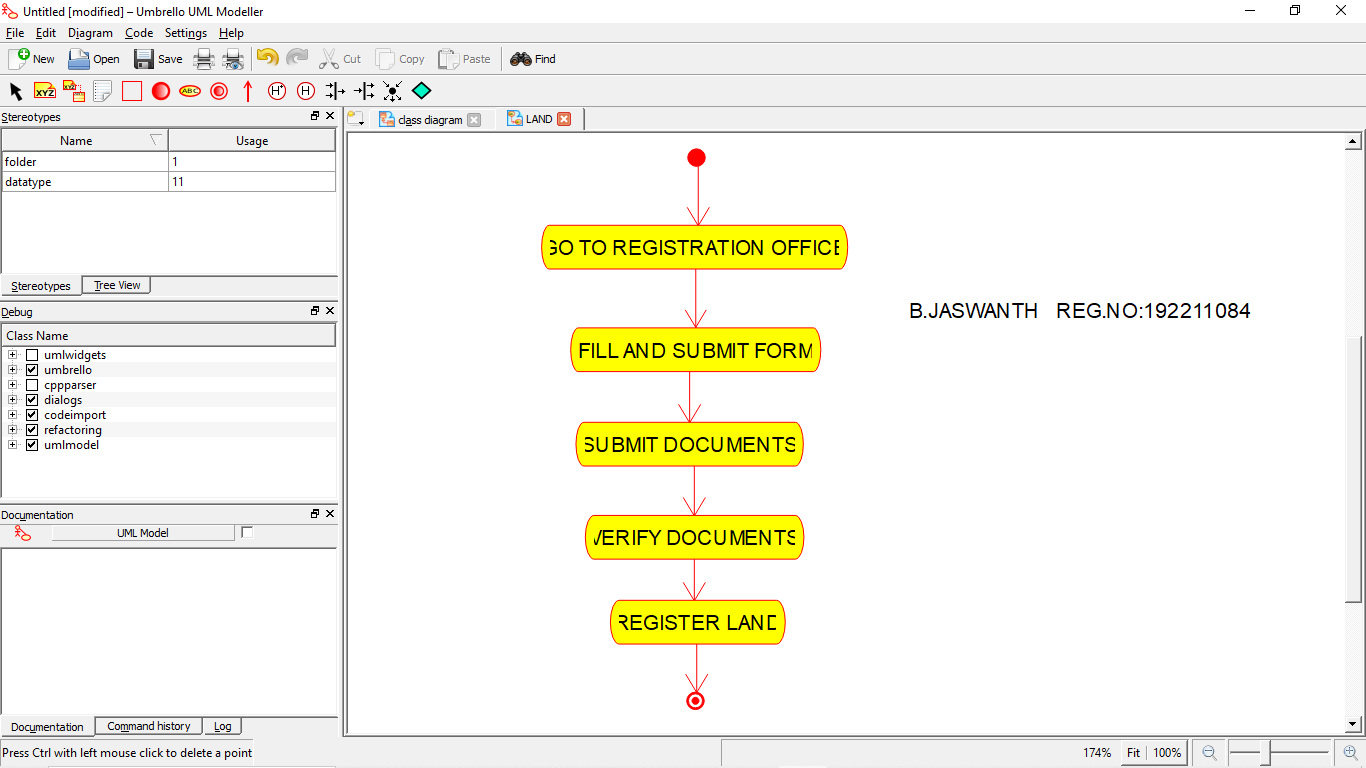


**SEQUENCE DIAGRAM**

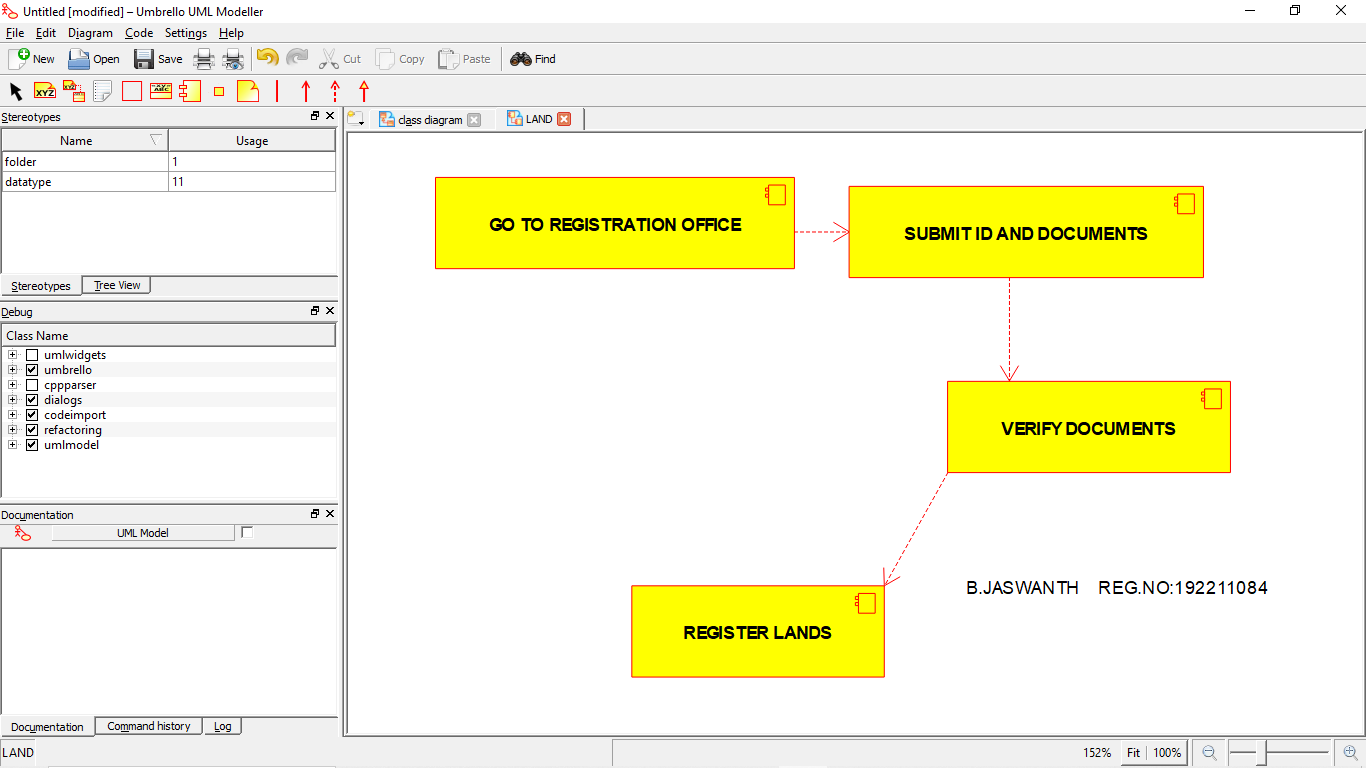


35. Develop a system using UML for Land Registration System. The system should maintain the database of the land detail and the owner of those lands. The system also maintains the citizen’s details having the lands in their name. Whenever the land is registered, the land owner's detail should be updated the same as the citizen's detail too. The amount transacted for the registration and the tax paid for it is also to be updated.

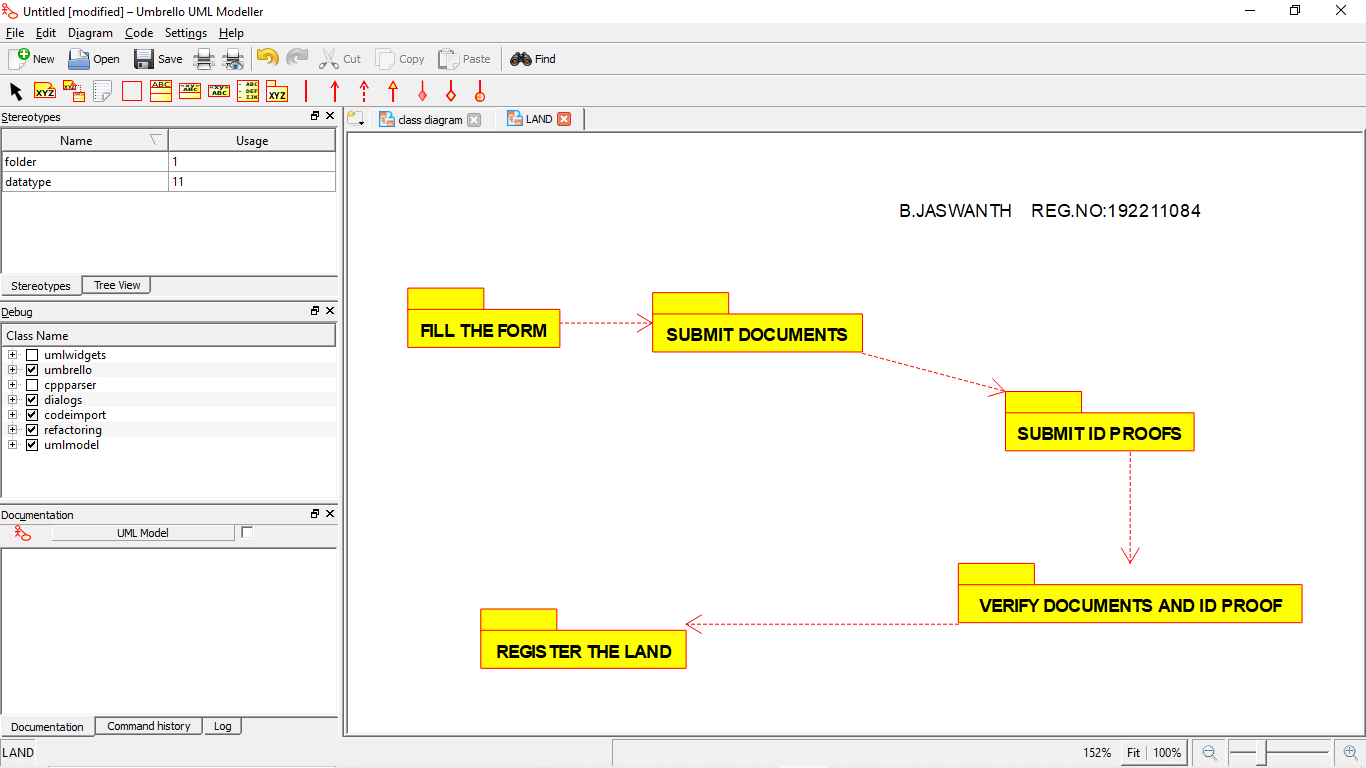
**ACTIVITY DIAGRAM**



**COMPONENT DIAGRAM**

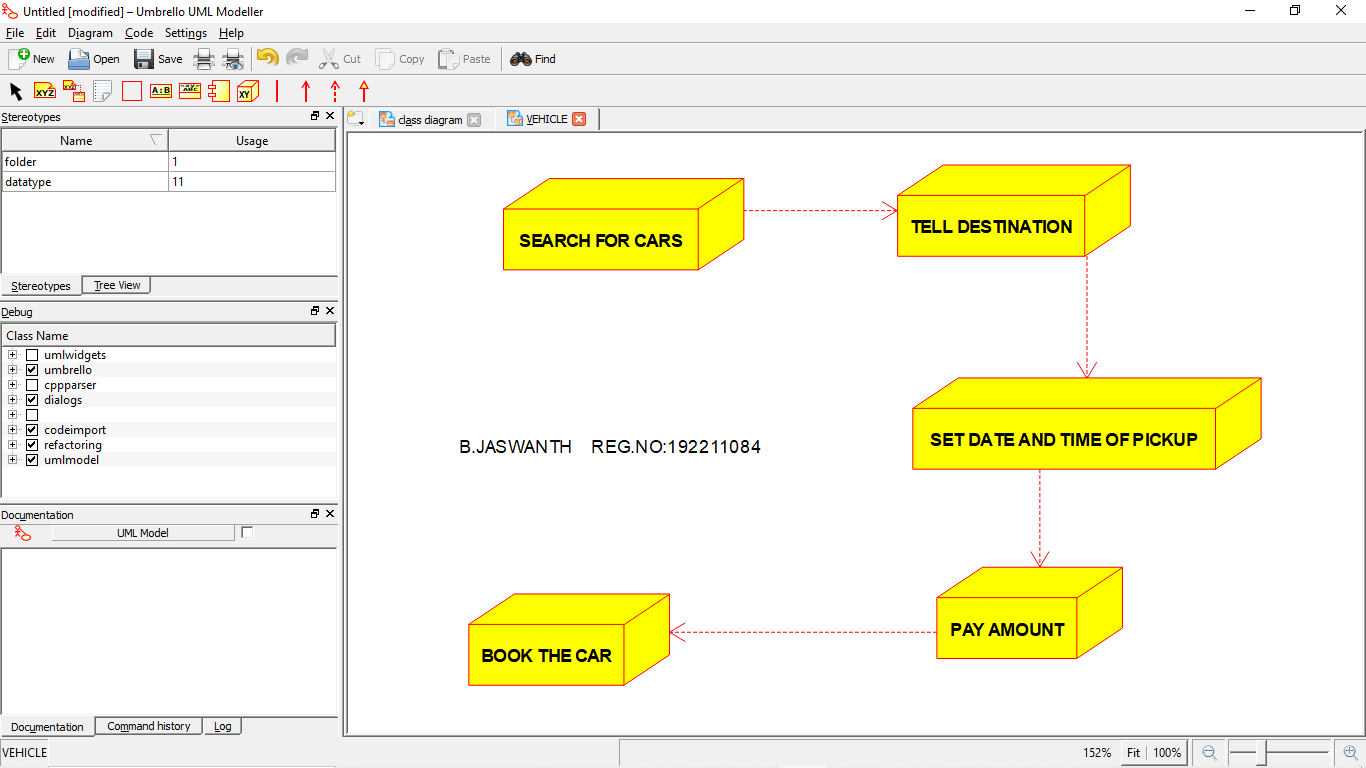


**PACKAGE DIAGRAM**

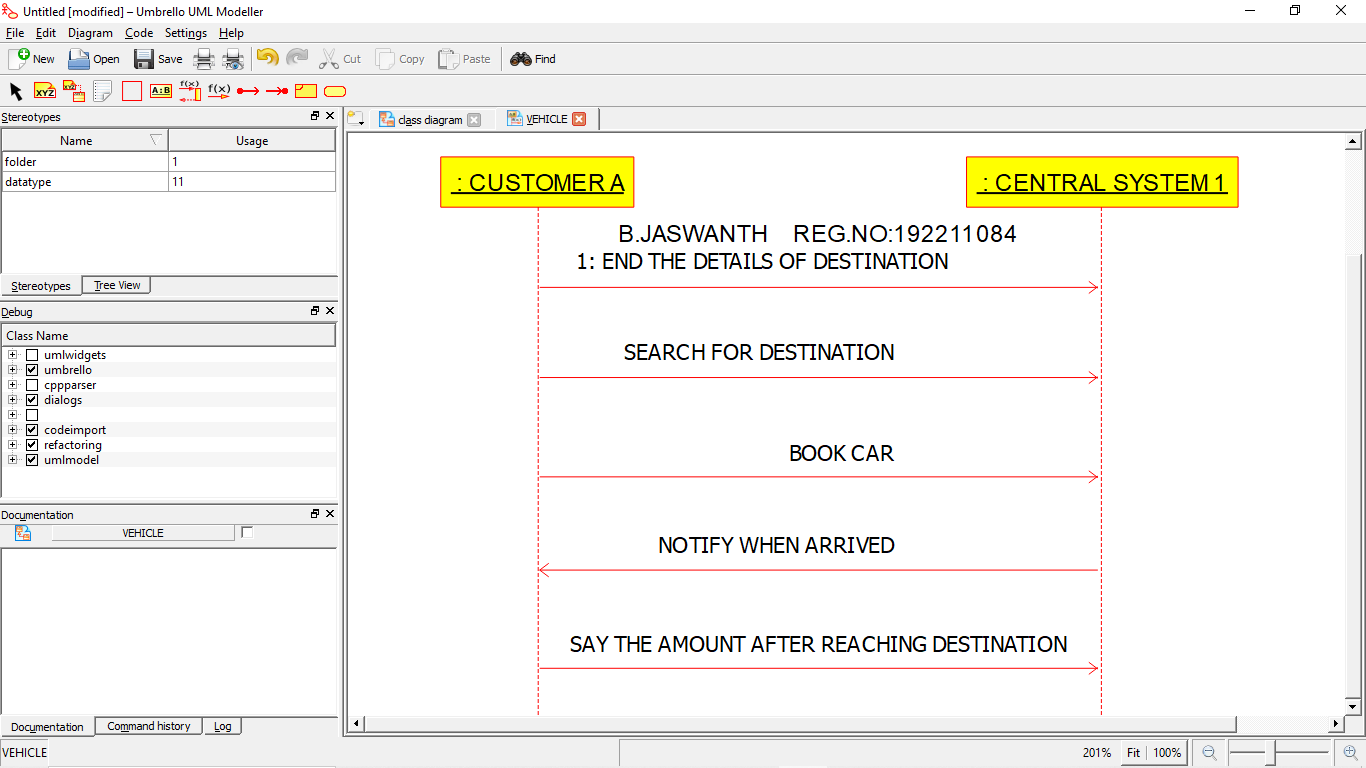


36. Develop a system using UML for Car/Bike Vehicle Booking System. The system should support the renting of different automobiles by the customer. Each vehicle should have a unique access detail. The system should be able to send a notification whenever the reservation is approaching the pickup date, as well as when the vehicle is nearing the due date. The system should maintain a vehicle log to track all events related to the vehicles.

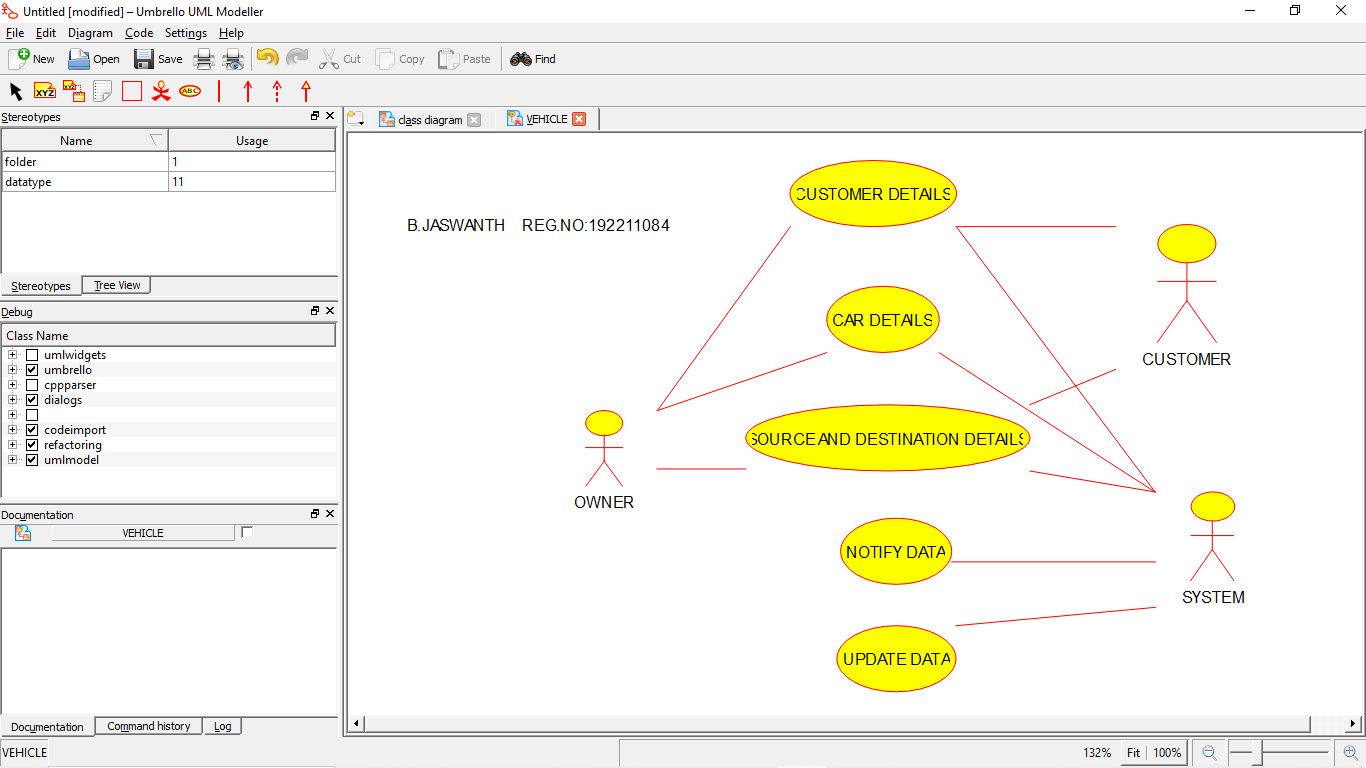
**DEPLOYMENT DIAGRAM**



**SEQUENCE DIAGRAM**

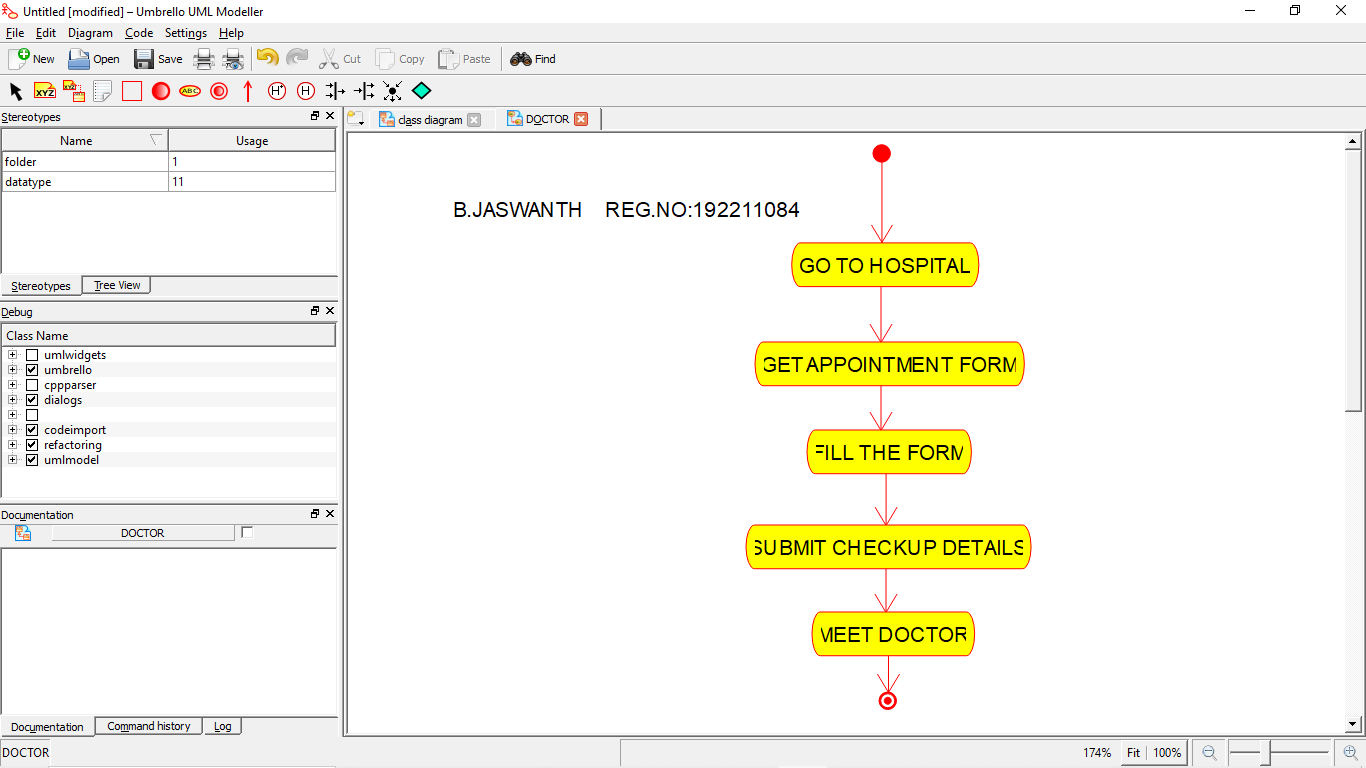


**USE CASE DIAGRAM**

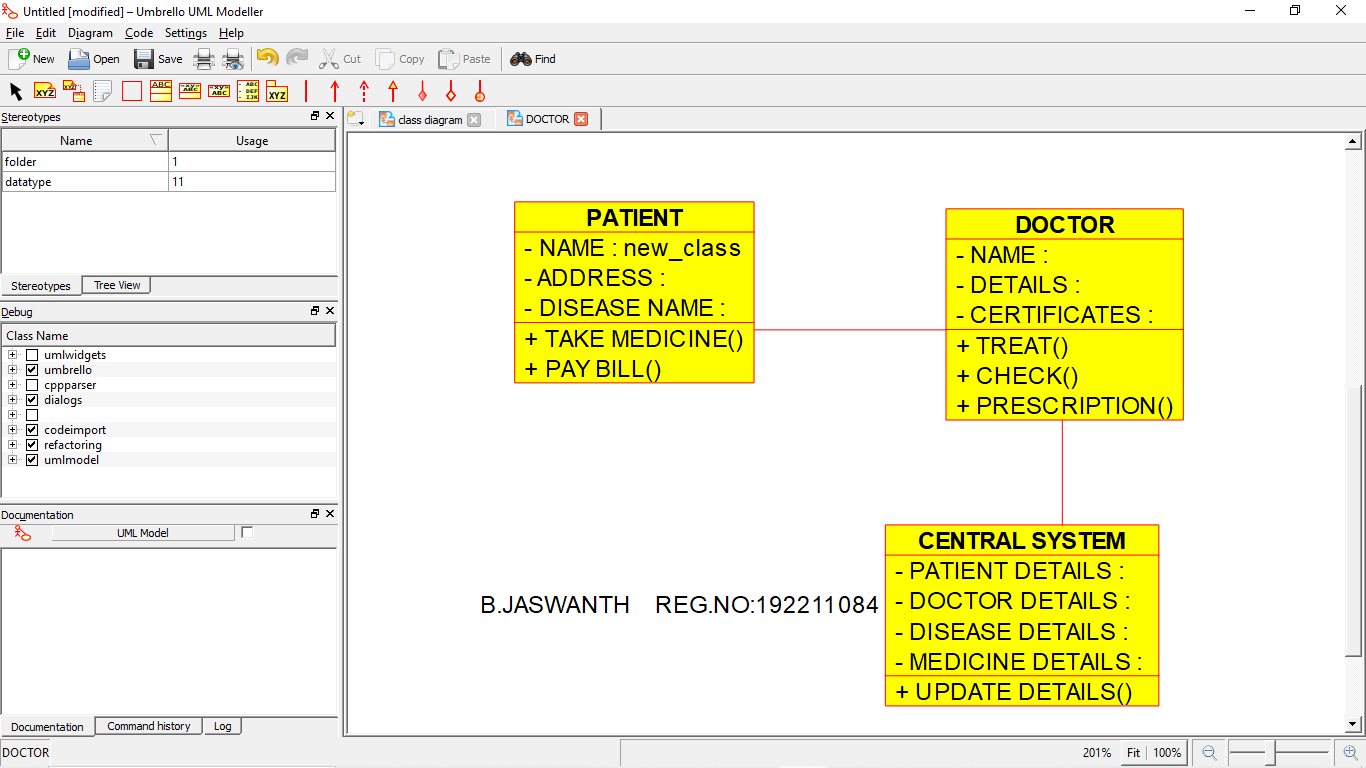


37. Develop a system using UML for Doctor Appointment System. The central system should manage doctors’ availability and appointments for the given date. The patient can book the doctor’s appointment based on the doctor’s availability. When the booking is confirmed, it should be updated with the patient and doctor.

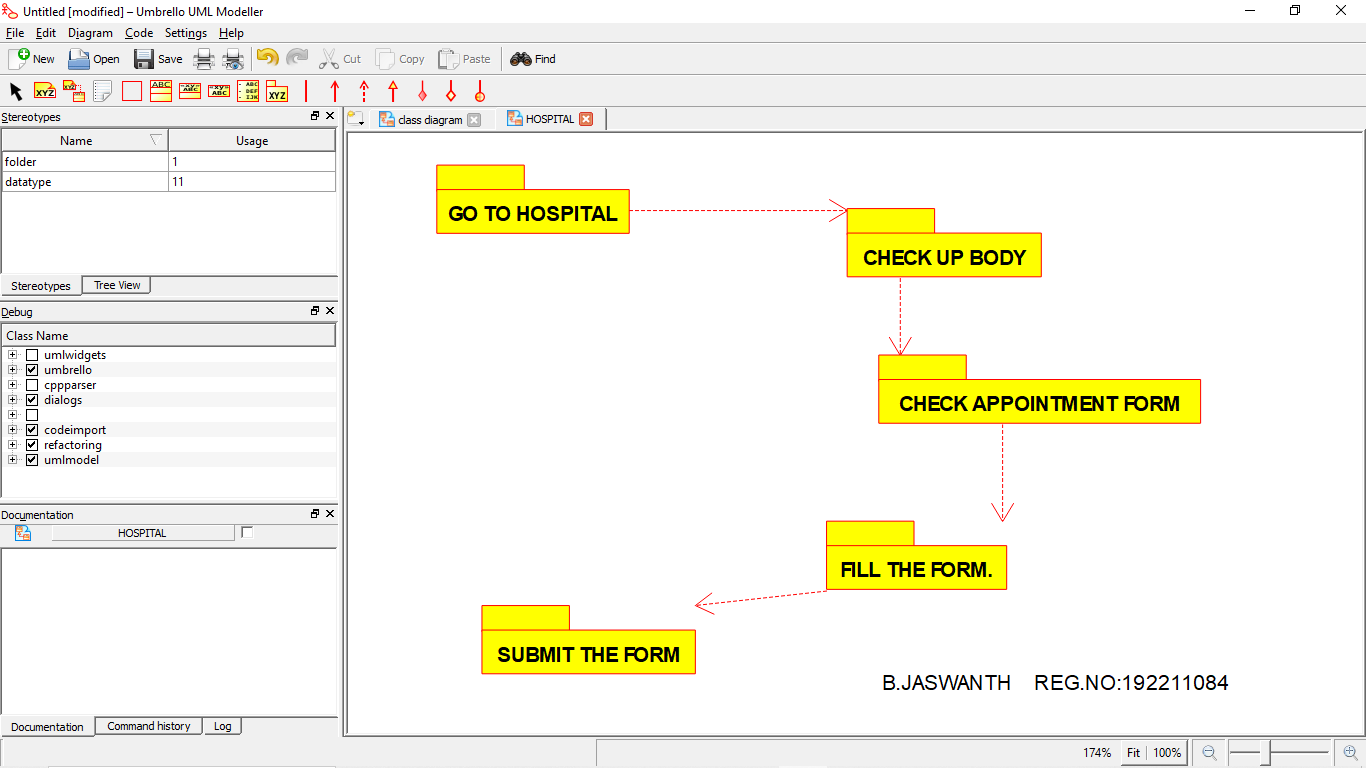
**ACTIVITY DIAGRAM**



**CLASS DIAGRAM**

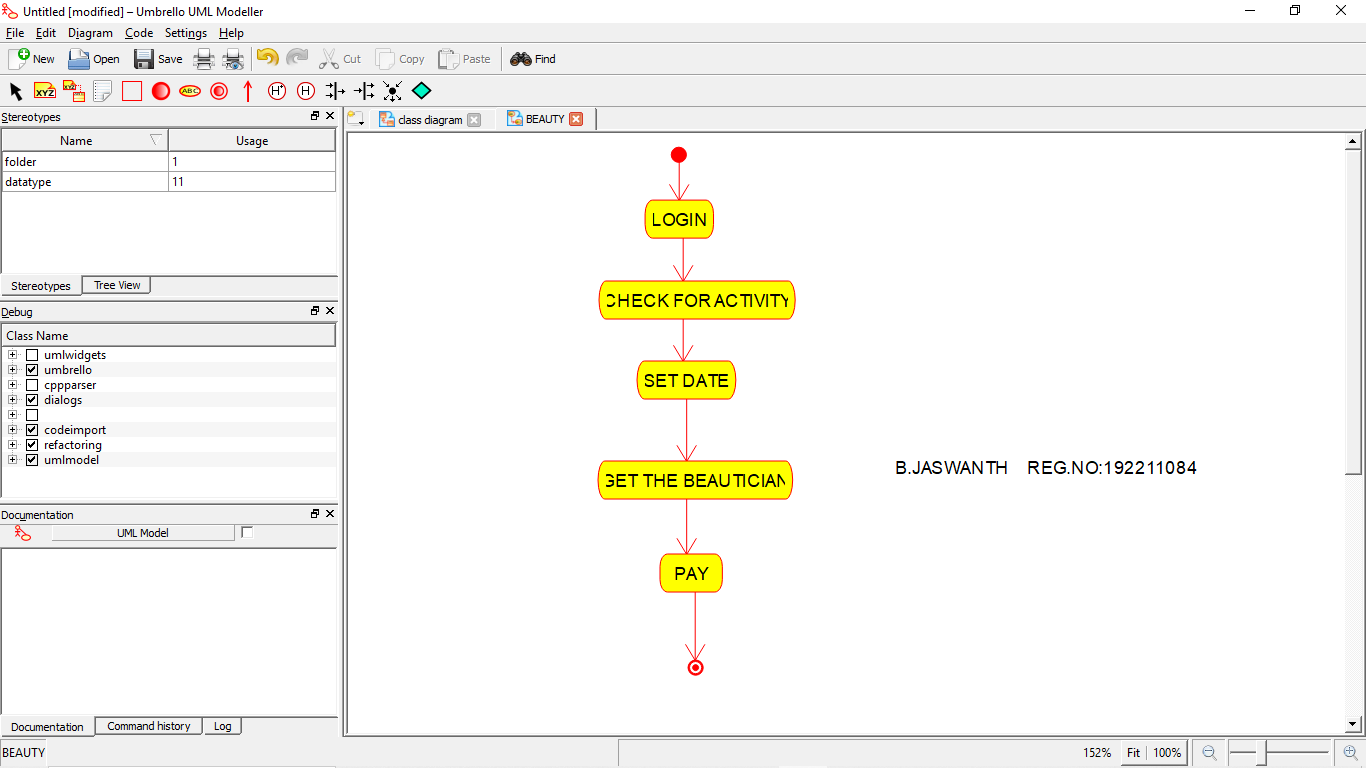


**PACKAGE DIAGRAM**

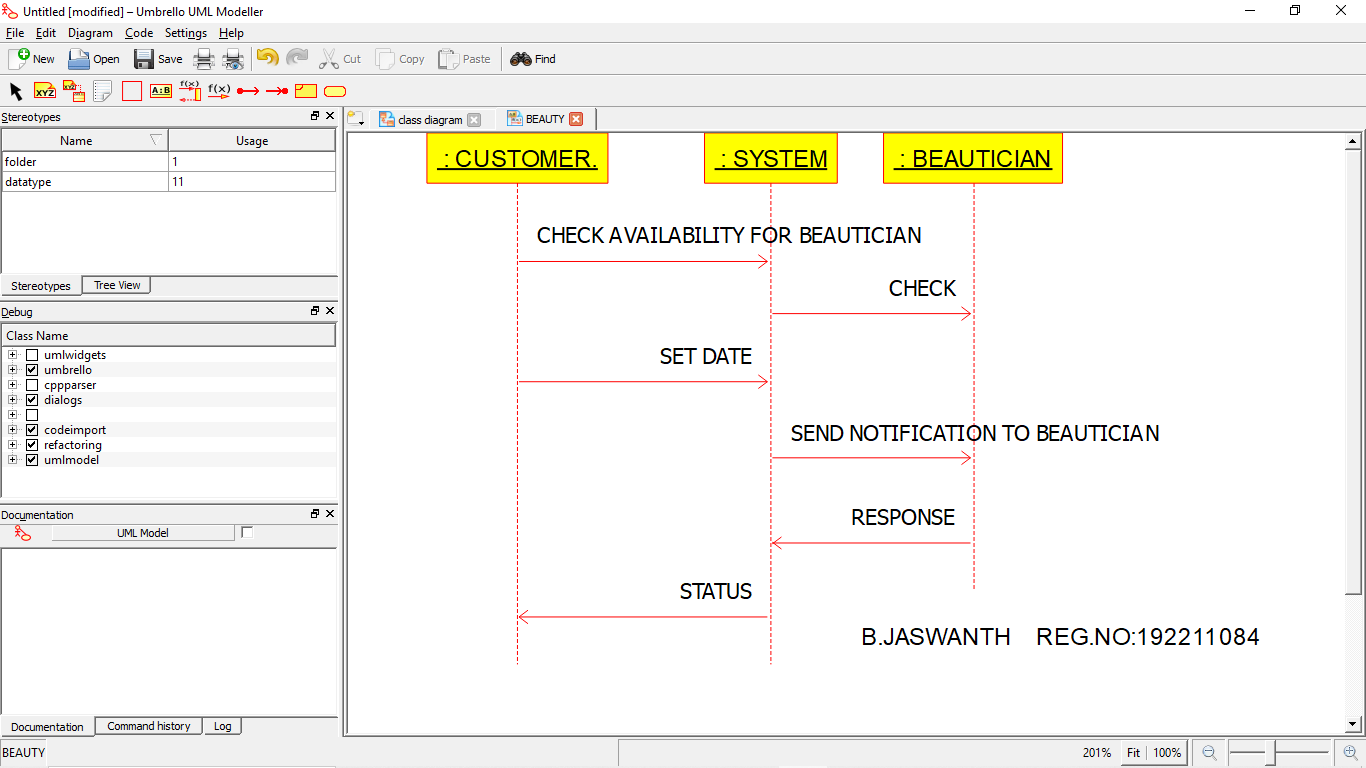


38. Develop a system using UML for Salon/Beauty Parlour Reservation System. The central system should manage beauticians’ availability and appointments for the given date. The customer can book the beautician appointment based on availability. When the booking is confirmed, it should be updated for the patient and beautician.

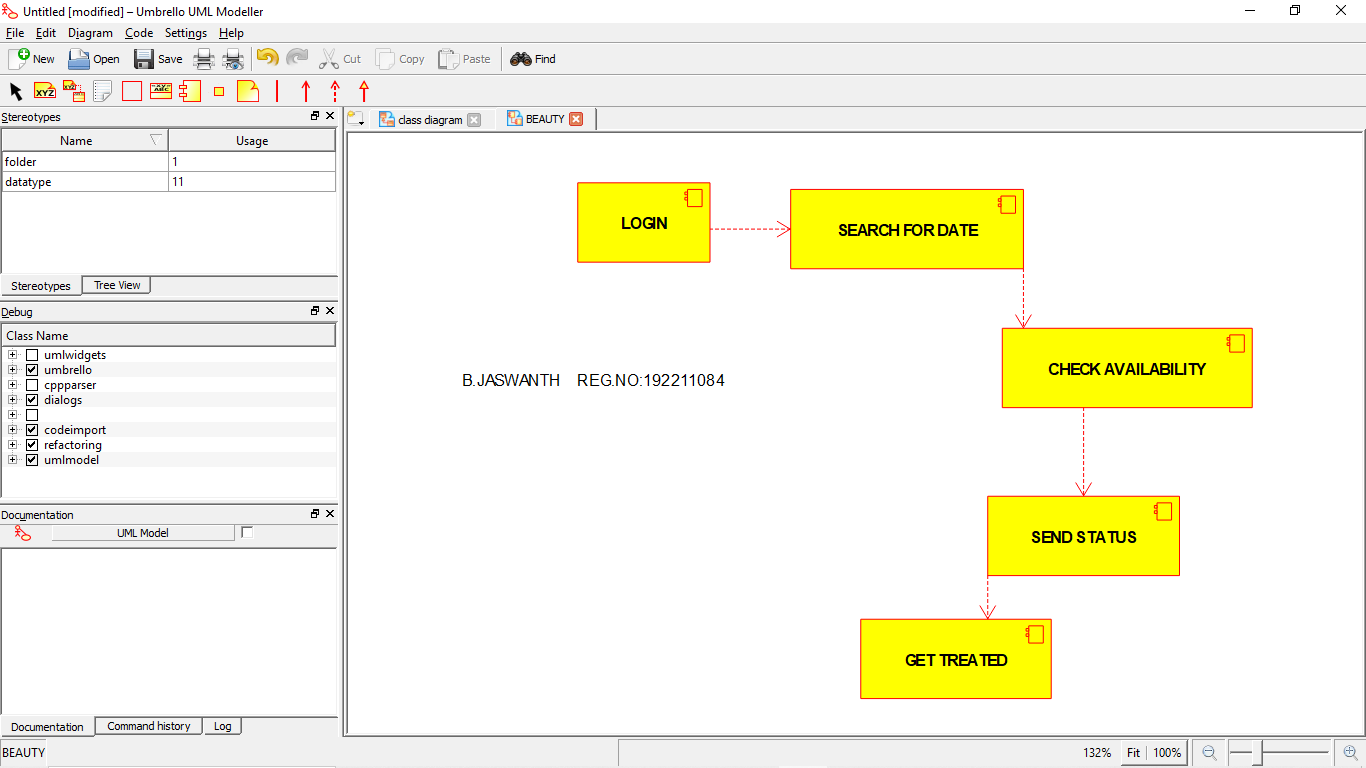
**ACTIVITY DIAGRAM**



**SEQUENCE DIAGRAM**

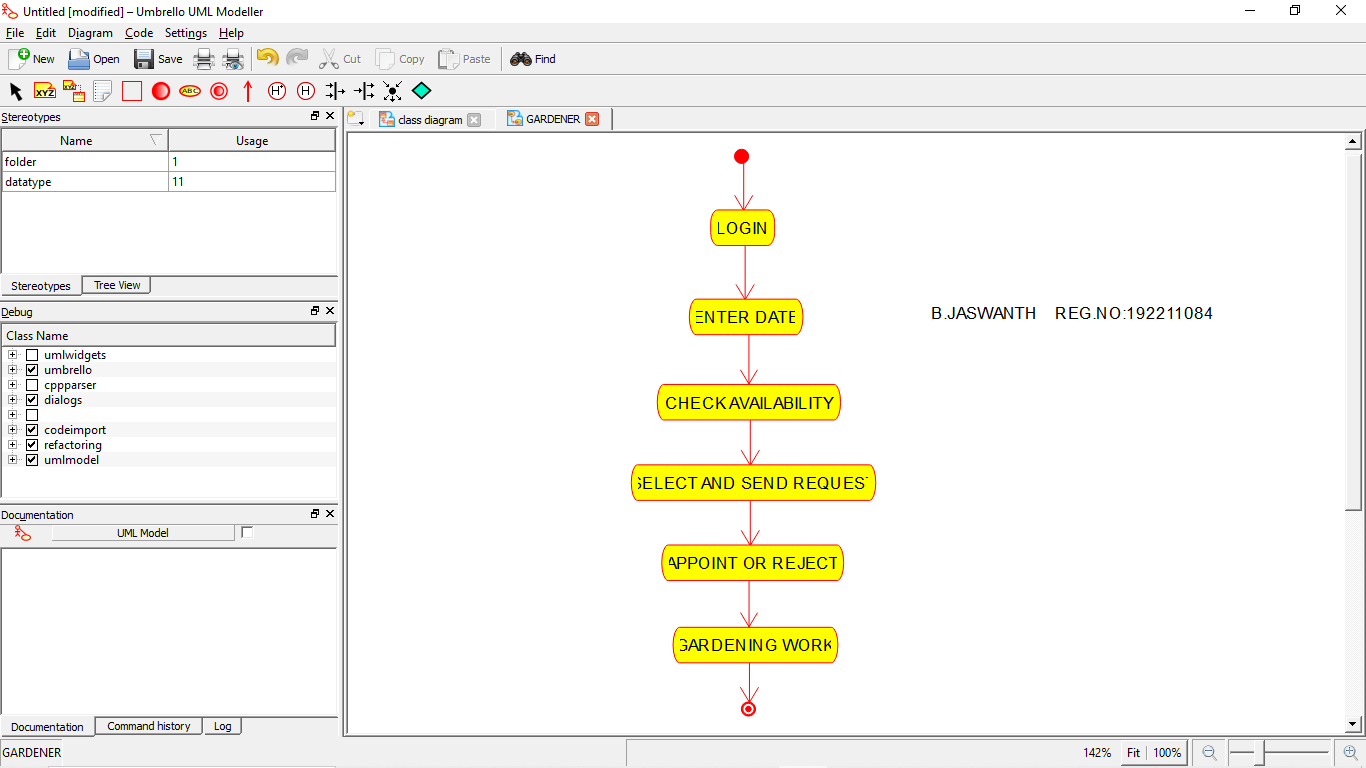


**COMPONENT DIAGRAM**

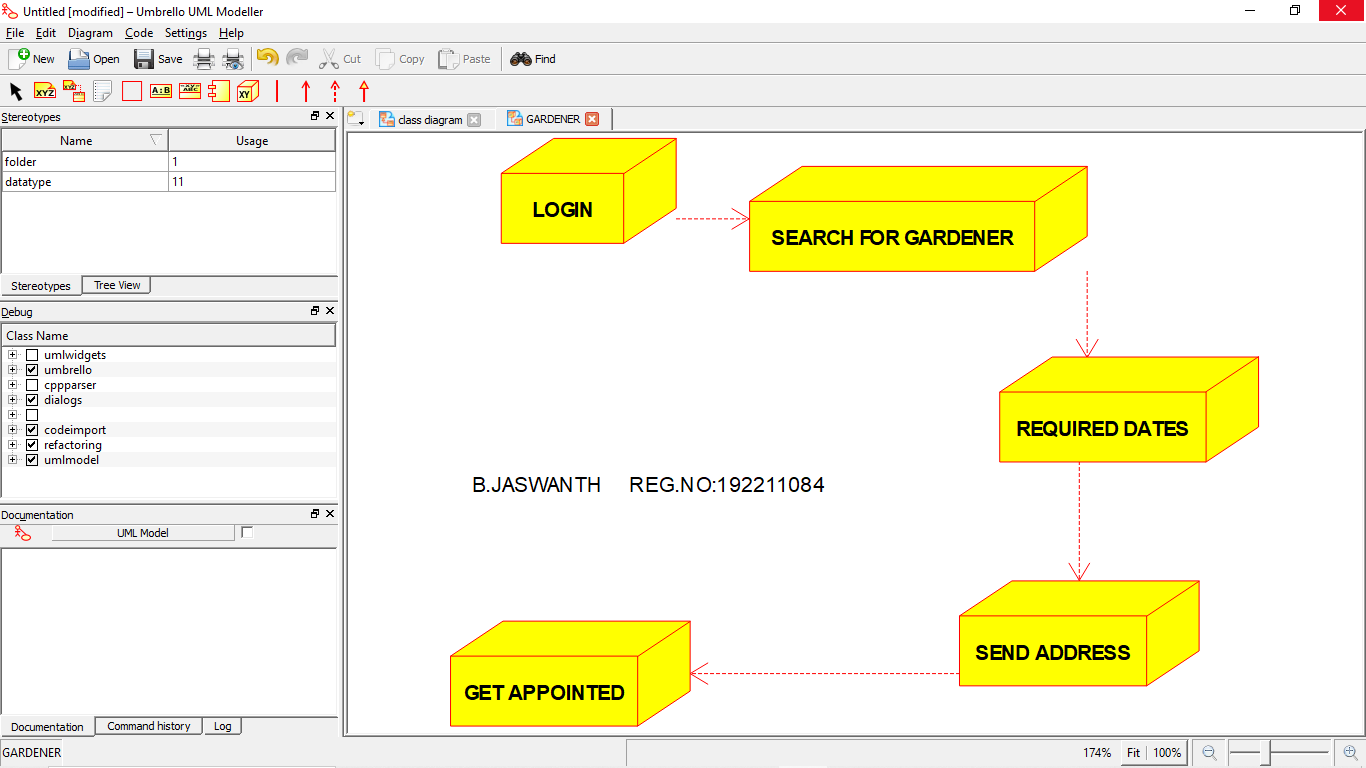


39. Develop a system using UML for Gardener Booking System. The central system should manage the list of gardeners and their availability and appointments for the given date. The customer can book the gardener’s appointment based on availability. When the booking is confirmed, it should be updated for the gardener and beautician.

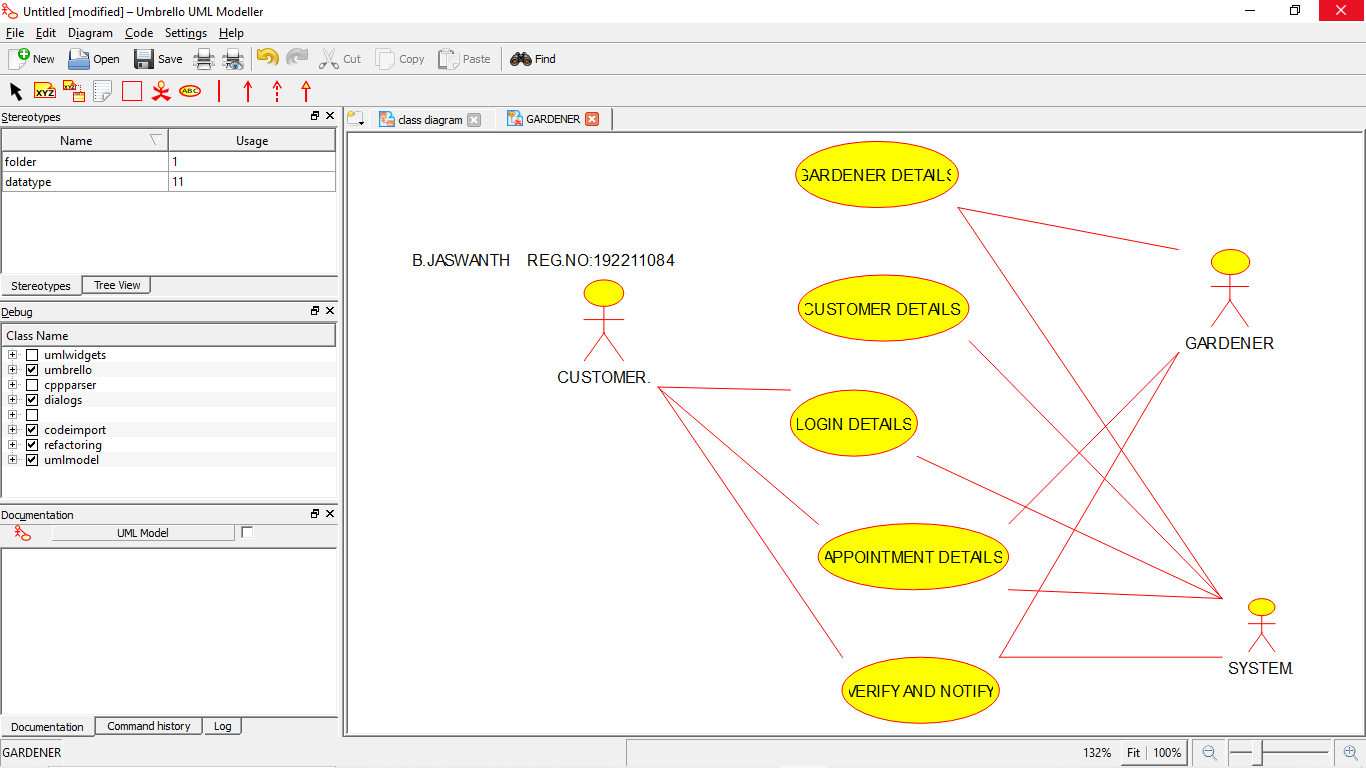
**ACTIVITY DIAGRAM**



**DEPLOYMENT DIAGRAM**

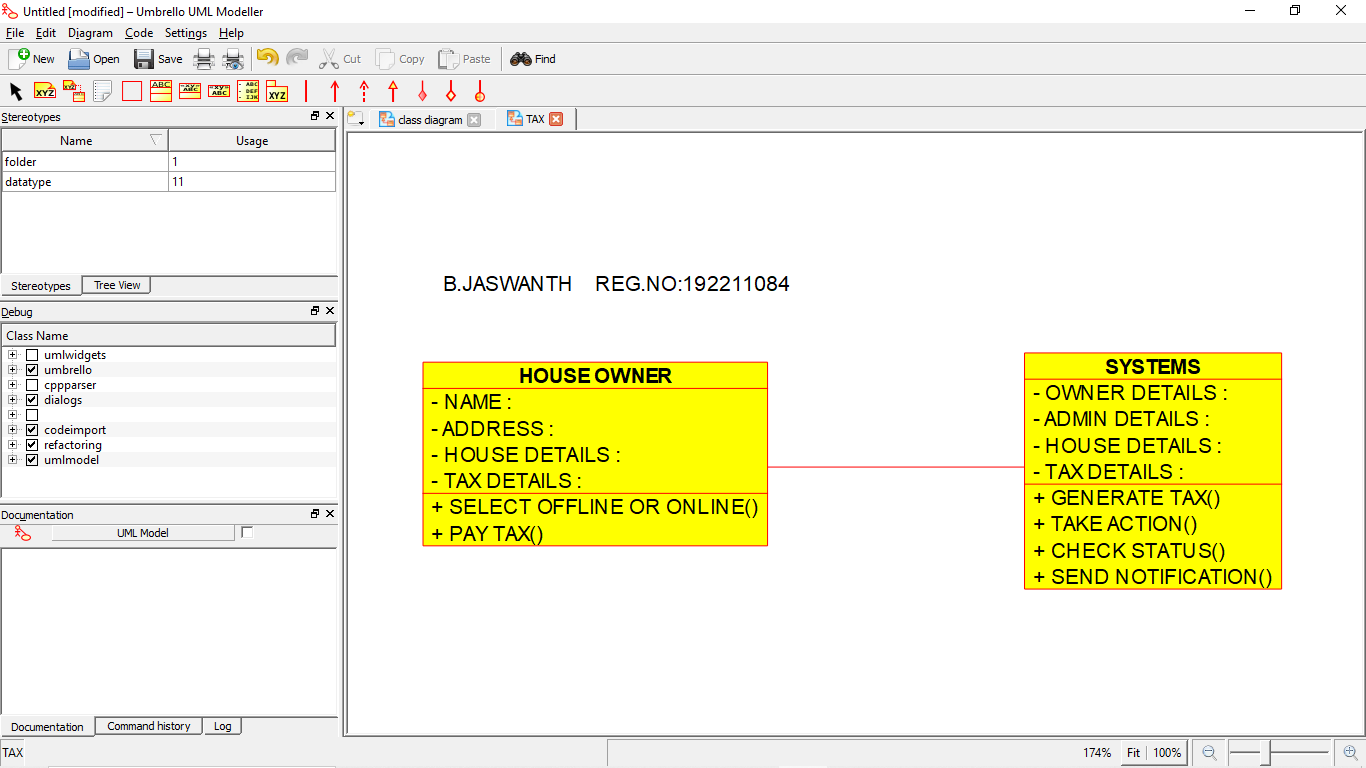


**USE CASE DIAGRAM**

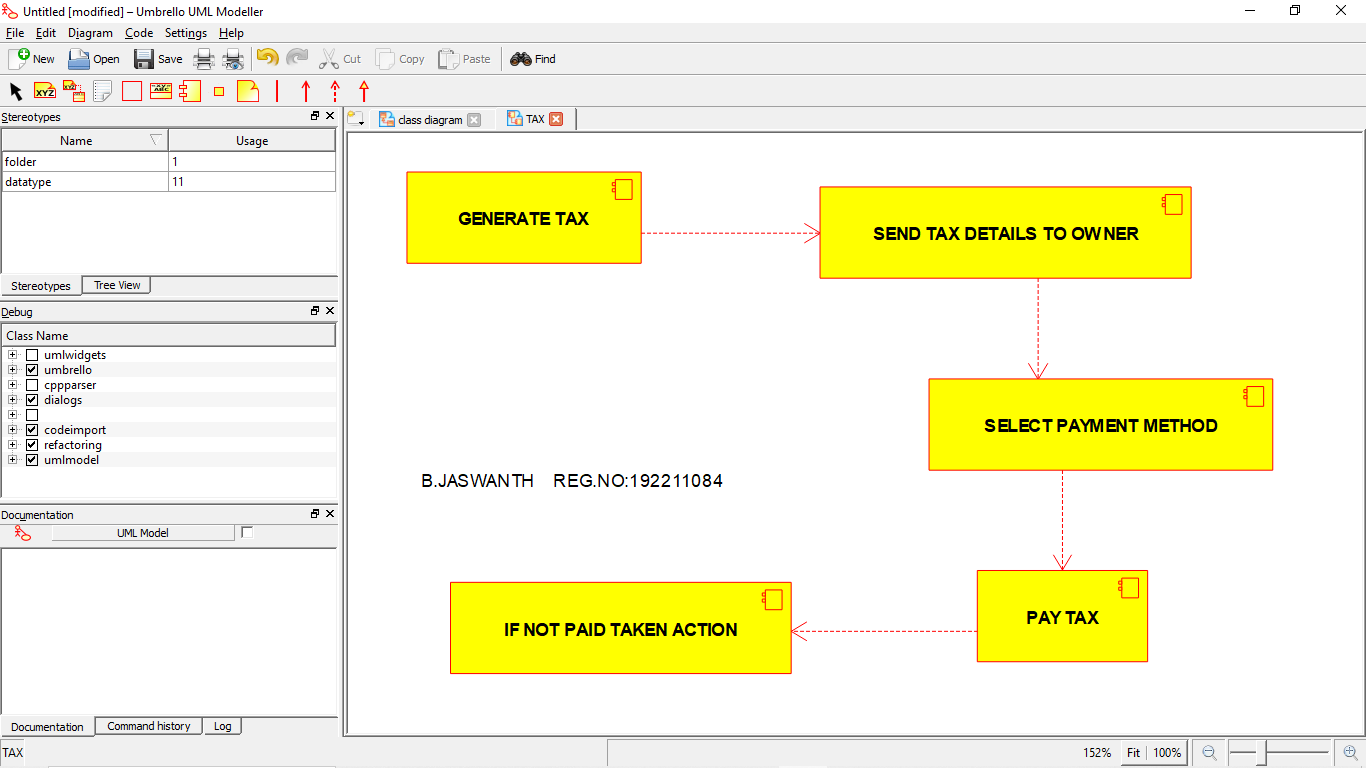


40. Develop a system using UML for HOME Tax Collection System. The system should generate a home tax for all the homes in the stipulated area it covers and the same should be sent to the owner of the home. The owners should be allowed to pay the tax online or offline. When the deadline for the bill is nearer, a reminder message should be sent to the owner. When the deadline is crossed, it should be intimated to theowner and the admin to take corresponding actions.

CLASS DIAGRAM



COMPONENT DIAGRAM



PACKAGE DIAGRAM

