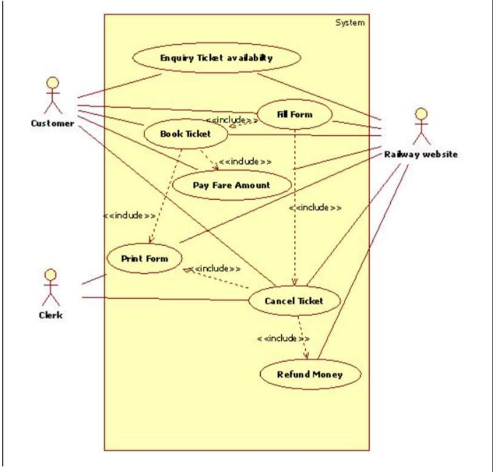
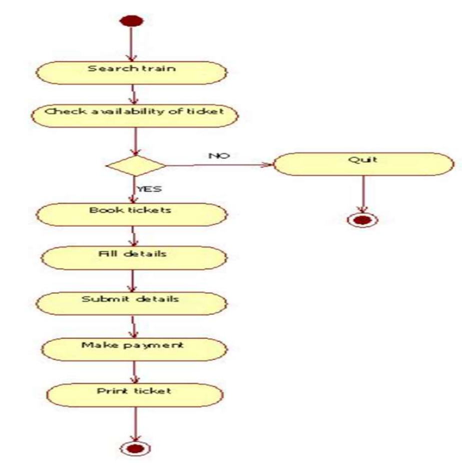
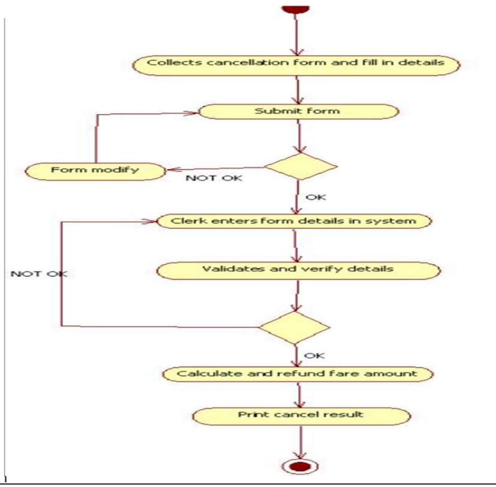
Use Case Diagram :- Railway Reservation



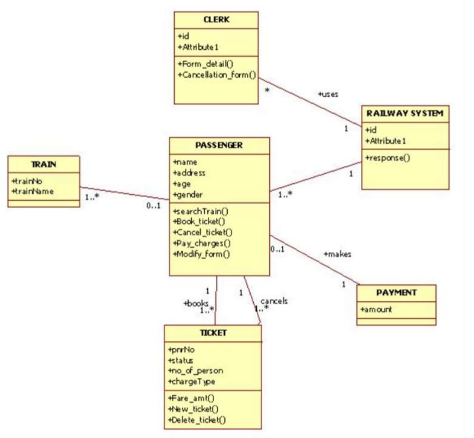
ACTIVITY DIAGRAM FOR BOOKING TICKET:



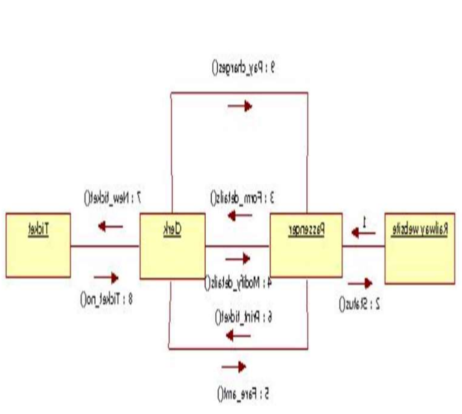
ACTIVITY DIAGRAM FOR CANCEL TICKET


Collects cancellation form and fill in details
Submit form
Form modify
NOT OK
OK
Clerk enters form details in system
NOT OK
Validates and verify details
OK
Calculate and refund fare amount
Print cancel result

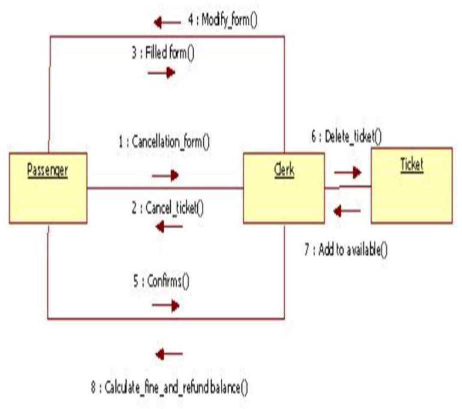
CLASS DIAGRAM FOR RAILWAY RESERVATION SYSTEM:



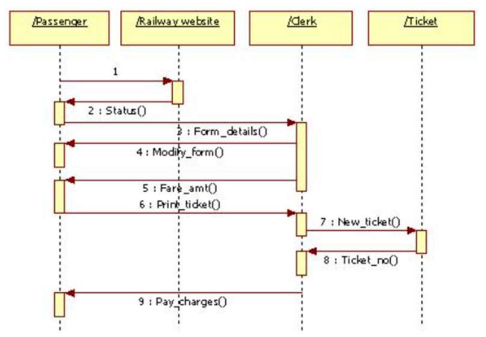
COLLABORATION DIAGRAM FOR BOOKING TICKET



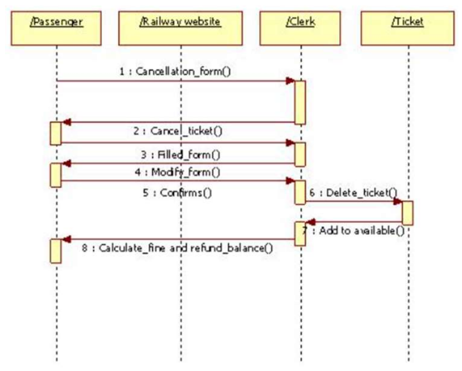
COLLABORATION DIAGRAM FOR CANCEL TICKET:



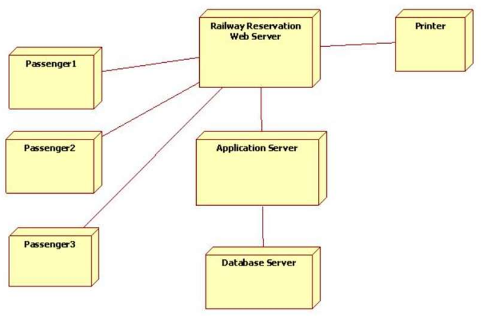
SEQUENCE DIAGRAM FOR BOOKING TICKET:



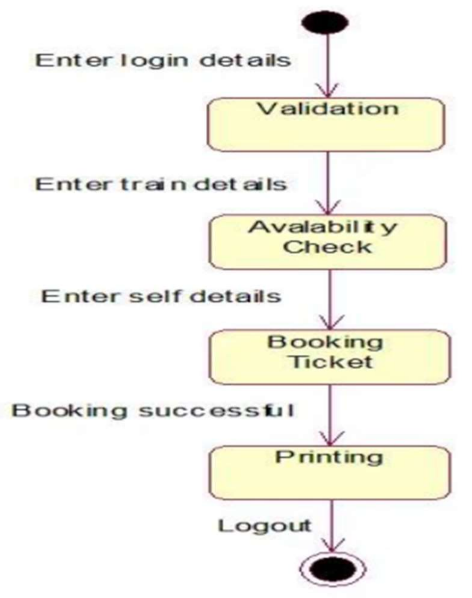
SEQUENCE DIAGRAM FOR CANCEL TICKET:



DEPLOYMENT DIAGRAM FOR RAILWAY RESERVATION SYSTEM:



State chart:



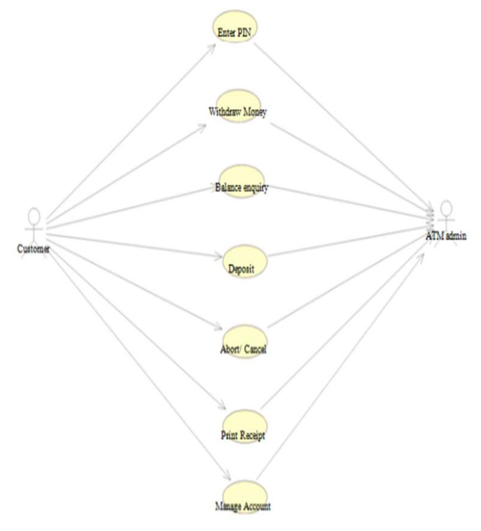
ATM Case Study

1 .Use Case Diagram for ATM System, with use case specification.

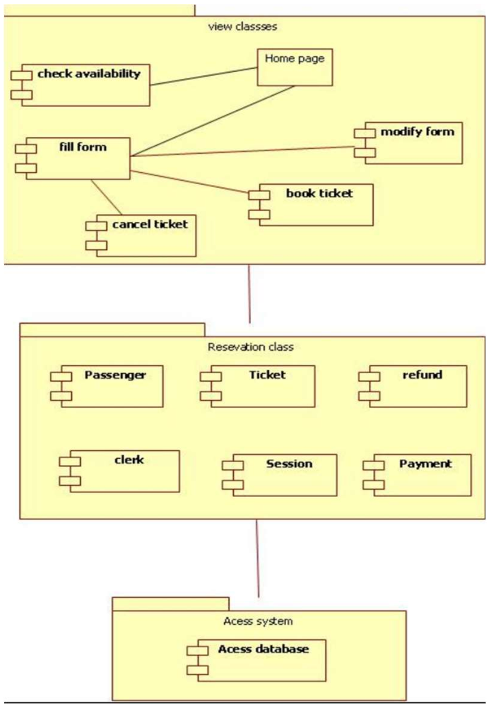
Step1: First an Actor is Created and named as User/Customer. Step2: Secondly a system is created for ATM.

Step3: A use case Enter PIN, Withdraw money is created and connected with user as association relationship.

Step4: Similarly various use cases like Deposit money, Balance Enquiry, Manage Account etc are created and appropriate relationships are associated with each of them

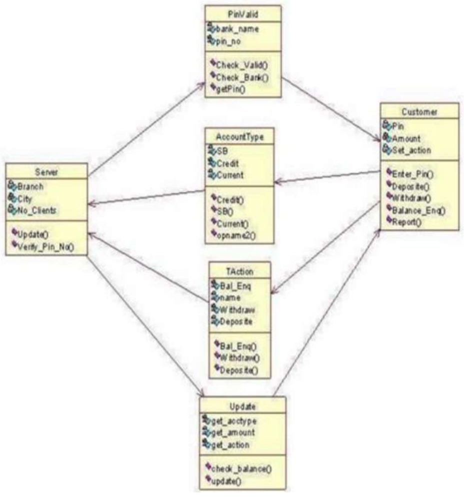


COMPONENT DIAGRAM FOR RAILWAY RESERVATION SYSTEM :

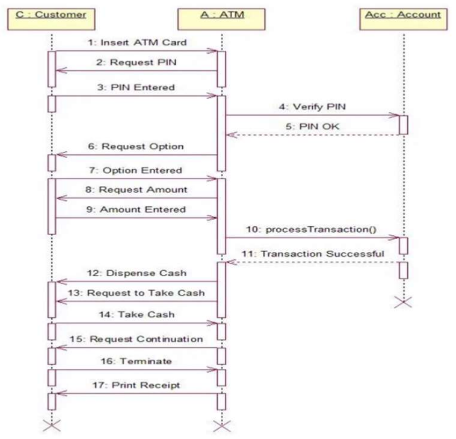


2. Identify the classes for the ATM system and draw the Class Diagram

Step1: First Classes are created.
Step2: Named as PinValid, Account Type, Transaction, Update, Server, Customer classes are created.
Step3: Appropriate relationships are provided between them as association.

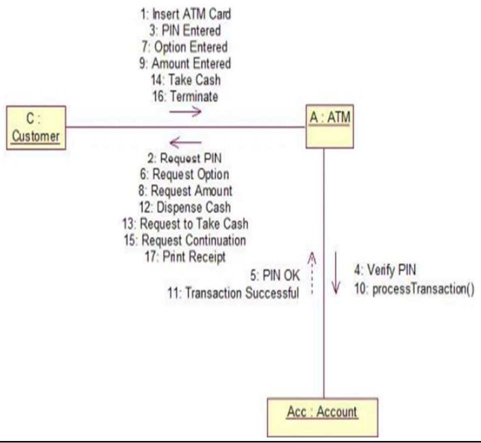


3. Draw the Sequence for each of the use cases Identified.
Step1: First An actor is created and named as user.
Step2: Secondly an object is created for Atm.
Step3: Timelines and lifelines are created automatically for them.
Step4: In sequence diagram interaction is done through time ordering of messages. So appropriate messages are passed between user and ATM is as shown in the figure.

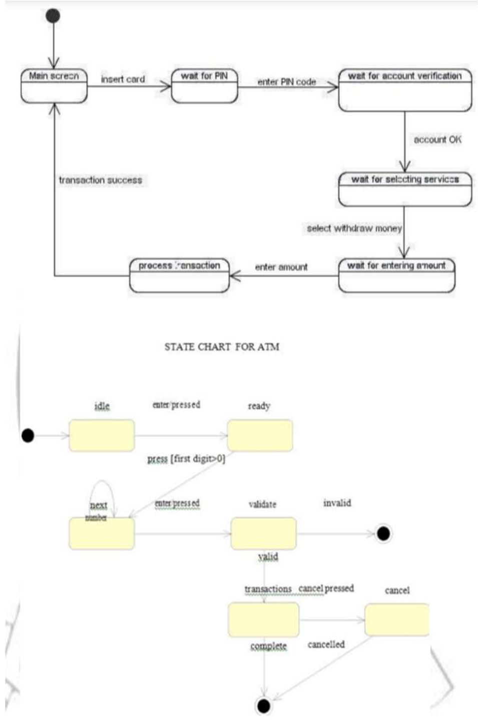


Collaboration diagram:

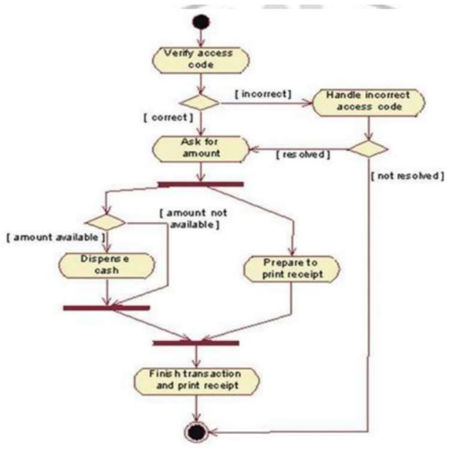
Step1: First an actor is created and named as user.
Step2: Secondly an object is created for ATM.
Step3: In collaboration diagram interaction is done through organization.
Step4: So appropriate messages are passed between user and ATM as shown in the figure.



Draw the State chat Diagram for the system.

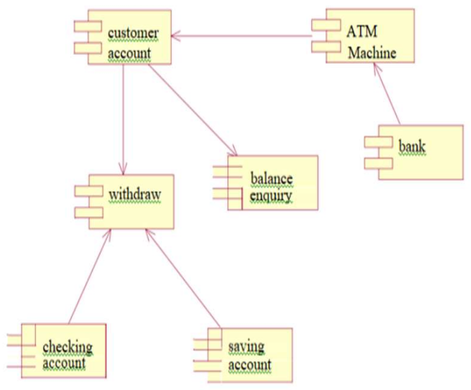


Draw the Activity Diagram for the system
Step1: First initial state is created.
Step2: After that it goes to the action state insert card.
Step3: Next it undergoes transition to the state enter pin
Step4: In this way it undergoes transitions to the various states.
Step5: Use forking and joining wherever necessary

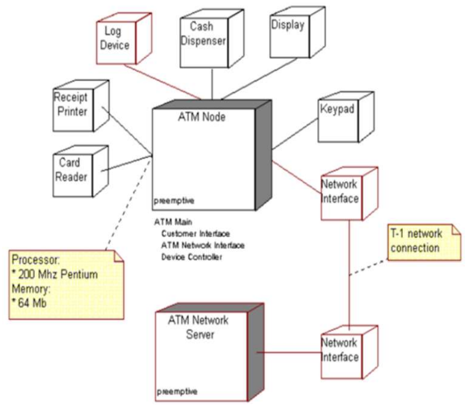


Draw the Component Diagram for the system

Step1: First user component is created.
Step2: ATM system package is created.
Step3: In it various components such as withdraw money, deposit money, check balance, transfer money etc. are created.
Step4: Association relationship is established between user and other components.



Draw the deployment Diagram for the system



Version Control System GIT :
Git is a Version Control System or VCS. VCS is basically software designed to record changes within one or more files over time. It allows us to undo or to cancel all made or pending changes within one or more files. If we're working on a project with many files, VCS enables us to control the whole project.

1. Working Locally with GIT :
One of the main goal of version control is to save snapshots of your directory. We call those snapshots commits. To each snapshot is associated some metadata: the date the snapshot was taken, who took it, what files where modified, the changes that were made on those files etc. Git will enable you to track the changes made to the files, revert the entire project to a previous snapshot, review changes made over time, view who modified a file.
..Now that we know why we want to create a snapshot, let's see how to do this. Right now, nothing in the project is tracked. First let's create some directories and files in our directory.

touch README

You now have a new file in the project folder. As mentioned before, git does not track this file yet. First, you need to tell git this file exists:

git add README

Now you can commit it:

git commit -m "Added a README file"

At this point, you have one tracked file, and an initial commit. Each file in the working directory can be in one of two states: tracked or untracked. Any file you have not explicitly added at some point is untracked. Tracked files can themselves be in different states: unmodified, modified or staged.
You can check the status of each file in your directory with the command git status. This command will display all untracked files, and modified and staged files in your directory

ouch AUTHORS git status

Note as the README file is not listed, while the AUTHORS file is there. Now let's add the AUTHORS file. The AUTHORS file is now tracked, and staged:

git commit -m "Added the AUTHORS file"

And here is the second commit! Sometimes, you may want to look at the changes you've made to a modified file

git diff

To look at the changes you've made in the staged files, simply use:

git diff –cached

Now, let's try to rm AUTHORS:

rm AUTHORS CONTRIBUTORS git status

You can see that the AUTHORS file is in red, and marked as deleted. Yet, running git add on this file doesn't move the change in the staging area. To remove a file that has been tracked with git, use the git rm command. In a similar fashion, git mv can be used to move a file. Because to err is human, you may want to cancel some stages. Two scenarios may occur: (1) you have staged a file you do not want to commit (2) you have made some changes on a file you want to cancel. First, let's assume you've staged a file you want to onstage

touch TODO git
add TODO

To unstage it, run:

git reset HEAD TODO

The syntax is git reset HEAD . We will explain what HEAD is later on. In the second case, you've modified a file and you want to cancel the changes. To do so, use git checkout . If you run git status, you can notice git reminds you what command to use for which action.

How to setup Git on premises Hardware :
To use Git on the command line, you will need to download, install, and configure Git on your computer. You can also install GitHub CLI to use GitHub from the command line. For more information, see "About GitHub CLI." If you want to work with Git locally, but do not want to use the command line, you can download and install the GitHub Desktop client. For more information, see "Installing and configuring GitHub Desktop." If you do not need to work with files locally, GitHub lets you complete many Git-related actions directly in the browser, including:
1. **Download and install the latest version of Git.**
Note: If you are using a Chrome OS device, additional setup is required:
a. Install a terminal emulator such as Termux from the Google Play Store on your Chrome OS device.
b. From the terminal emulator that you installed, install Git. For example, in Termux, enter apt install git and then type y when prompted.
Set your username in Git.
Set your commit email address in Git.
Authenticating with GitHub from Git
When you connect to a GitHub repository from Git, you will need to authenticate with GitHub using either HTTPS or SSH.
Note: You can authenticate to GitHub using GitHub CLI, for either HTTP or SSH. For more information, see gh auth login.
Connecting over HTTPS (recommended)
If you clone with HTTPS, you can cache your GitHub credentials in Git using a credential helper. For more information, see "About remote repositories" and "Caching your GitHub credentials in Git." Connecting over SSH If you clone with SSH, you must generate SSH keys on each computer you use to push or pull from GitHub. For more information, see "About remote repositories" and "Generating a new SSH key and adding it to the ssh-agent." Next steps You now have Git and GitHub all set up. You may now choose to create a repository where you can put your projects. Saving your code in a repository allows you to back up your code and share it around the world
. • Creating a repository for your project allows you to store code in GitHub. This provides a backup of your work that you can choose to share with other developers. For more information, see "Create a repository.".
• Forking a repository will allow you to make changes to another repository without affecting the original. For more information, see "Fork a repo."
• Each repository on GitHub is owned by a person or an organization. You can interact with the people, repositories, and organizations by connecting and following them on GitHub. For more information, see "Be social."
• GitHub has a great support community where you can ask for help and talk to people from around the world. Join the conversation on GitHub Community.

Install and setup Jenkins
Install Jenkins on Windows
1. Browse to the official Jenkins download page. Under the Downloading Jenkins section is a list of installers for the long-term support (LTS) version of Jenkins. Click the Windows link to begin the download.
2. Once the download is complete, run the jenkins.msi installation file.
3. The setup wizard starts. Click Next to proceed.
4. Select the install destination folder and click Next to continue.
5. Under the Run service as a local or domain user option, enter the domain username and password for the user account you want to run Jenkins with. Click Test Credentials to verify the login data, then click Next to proceed.
Using the Run service as LocalSystem option doesn't require you to enter user login data. Instead, it grants Jenkins full access to your system and services. Note that this is a less secure option and is thus not recommended.
Note: When selecting the Run service as local or domain user, the user in question must have the required permission to log on as a service. If this is not the case, an error message stating the account cannot be verified appears. In the event of this error, perform the following steps to resolve the issue:
1. Make sure you are logged in as a user with administrative privileges.
2. In the Administrative Tools, open Local Security Policy.
3. In the Local Security Policy window, expand Local Policy in the left-hand panel and select User Rights Assignment.
4. In the right-hand side panel, right-click Log on as service and select Properties.
5. Click the Add User or Group... button. 6. Use the Select Users or Groups dialogue to add the current user and click OK. 7. Save changes by clicking the OK button in the Properties window.
6. Enter the port number you want Jenkins to run on. Click Test Port to check if the selected port is available, then click Next to continue. 7. Select the directory where Java is installed on your system and click Next to proceed. 8. Select the features you want to install with Jenkins and click Next to continue. 9. Click Install to start the installation process. 10. Once the installation is complete, click Finish to exit the install wizard.