VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Name: Mohammad Shaad

Reg no: 21BCE1542

Software Engineering Lab (BCSE301P)

Combined document

Submitted to: Prof. K.P. Vijayakumar

Slot: L39+L40

Index

Lab task 1: SRS

Use:

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil the needs of all stakeholders (business, users).

Advantages:

According to International Standard ISO/IEC/IEEE 29148:2011 (Systems and software engineering — Life cycle processes — Requirements engineering), the benefits of documenting the software requirements include:

- It provides a realistic basis for estimating product costs, risks and schedules.
- It provides an informed basis for deploying a product to new users or new operational environments.
- It provides a basis for product enhancement.
- It forces a rigorous assessment of requirements before design can begin and minimizes later redesign.
- It establishes the basis for agreement between the acquirers or suppliers on what the product is to do (in market driven projects, the user input may be provided by marketing).

Example: SRS for Online Railway Reservation System

Introduction:

Purpose:

The purpose of this source is to describe the railway reservation system which provides the train timing details, reservation, billing and cancellation on various types of
reservation namely,

- Confirm Reservation for confirm Seat.
- Reservation against Cancellation.
- Waiting list Reservation.
- Online Reservation.
- Tatkal Reservation.

Scope:

“Railways Reservation System” is an attempt to simulate the basic concepts of an online Reservation system. The system enables to perform the following functions:

 SEARCH FOR TRAIN

 BOOKING OF A SELECTED FLIGHT

 PAYMENT

 CANCELLATION

 Freight Revenue enhancement

 Passenger Revenue enhancement

 Improved & optimized service

Description:

Project Perspective:

The railways reservation system is proposed with the following,

- The computerization of the reservation system will reduce a lot of paperwork and hence the load on the airline administrative staff.
- The machine performs all calculations. Hence chances of error are nil.
- The passenger, reservation, cancellation list can easily be retrieved and any required addition, deletion or updation can be performed.
- The system provides for user-ID validation; hence unauthorized access is prevented.

Product Features:

The product features include searching for an appropriate train according to the passenger's schedule, selection of date, time, train number, stoppages, duration, place of departure as well as the place of arrival, passenger information, payment, cancellation and select the class such as first class, second seating, sleeper class, first, second and third AC, AC chair car etc. The software will also help a passenger find the live train status and check the PNR status.

User Characteristics:

- EDUCATIONAL LEVEL - At least user of the system should be comfortable with English language.
- TECHNICAL EXPERTISE - User should be comfortable using general purpose applications on the computer system.

Constraints:

The system will run under windows98 or higher platforms of the operating system.

Assumptions:

The software will require the user/passenger to have a valid username and password in order to use the software as well as have knowledge about the railway reservation system. To function, the software depends on the access to internet.

Functional Requirements:

Performance:

- User Satisfaction - The system is such that it stands up to the user expectations.
- Response Time - The response of all the operation is good. This has been made possible by careful programming.
- Error Handling - Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- Safety and Robustness - The system is able to avoid or tackle disastrous action. In other words, it should be foul proof. The system safeguards against undesired events, without human intervention.
- Portable -T he software should not be architecture specific. It should be easily transferable to other platforms if needed.
- User friendliness - The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

Hardware Interface:

The hardware requirements of this software basically include cache size, RAM size, memory restrictions etc. For example, cache size of 512 kb, RAM size of 128 kb and Hard disk drive of 80GB etc.

Software Interface:

The connection to internet is necessary. The primary requirements of the software include Windows XP, FrontPage and any windows-based OS with DOS.

Non-Functional Requirements

Performance:

The software should be reliable and available 24 x 7. In case of any hardware failure, backups of the database should be retrieved from the user and saved by the organiser.

Safety and Security Requirements:

The system should automatically log out the customer after a period of inactivity and not leave any cookies on the user's computer containing the user's password.

Lab task 2: WBS

Use:

A work breakdown structure (WBS) is a tool that can be used for projects, programs, and even initiatives to understand the work that has to be done to successfully produce a deliverable(s). The benefits of creating a WBS include: it defines and organizes the work required.

Advantages:

A work breakdown structure (WBS) is a tool that can be used for projects, programs, and even initiatives to understand the work that has to be done to successfully produce a deliverable(s). The benefits of creating a WBS include:

- it defines and organizes the work required
- it facilitates the quick development of a schedule by allocating effort estimates to specific sections of the WBS
- it can be used to identify potential scope risks if it has a branch that is not well defined
- it provides a visual of entire scope
- it can be used to identify communication points
- it provides a visual of impacts when deliverables are falling behind
- it can be used to show and assign accountabilities and responsibilities

Example: WBS for Online Railway Reservation System

WBS Number	Task Name	WBS Description	Must start by date	Must end by date	No. of Hours	No. of Days	Resource	Author of item
1	Plan the travel information	The website will need a catalog of the schedule of the trains and the stations they will visit.	10-Jan	20-Feb	50	40	Various websites and travel	Pranav
1.1	Develop database structure	Create various tables consisting of trains and their respective carts along with timings	25-Jan	31-Jan	10	6	MySQL and Microsoft Excel	Vishal
1.2	Load data into database	Create a database online using any web-server and add the fore-mentioned data into respective tables	01-Feb	20-Feb	20	20	Any backend programing language like PHP	Shivangi
1.2.1	Build train catalogue	Add reachable destinations, with the fare and approximate time of travel	01-Feb	20-Feb	20	20	MySQL Queries	Archith
1.2.1.1	Identify destination train station	Research about Prospective destination train station and locations	01-Feb	10-Feb	5	10	Any search engine like Google	Archith
1.2.1.2	Contact person at ticket counter	Create a network of people at ticket counter, so as to provide various details of the ticket	11-Feb	20-Feb	10	9	A mode of communication to visit them	Aryan
1.2.1.3	Sign collaborations	Sign deals assuring a profit margin to the person at ticket counter and at the same time, making it affordable for the	21-Feb	29-Mar	5	8	Agreements	Aryan
1.3	Sign agreement with banks	Sign agreement deals with banks and various governmental bodies to enable online payments	25-Feb	10-Mar	20	15	Agreements	Shivangi
1.3.1	Sign legal papers with banks	Make a deal with various banks such as Axis bank, HDFC bank and so on to manage the payments of the customers	25-Feb	28-Feb	5	3	Agreements	Shivangi
1.3.2	Get permission from income tax department	Get permission from income tax department to enable online payments through cards and even UPI	01-Mar	10-Mar	15	10		Vishal
2	Purchase Infrastructure	Includes buying of various items to ensure the smooth functioning	30-Mar	20-Apr	15	21	Sufficient investment	Pranav
2.1	Purchase Hardware	Purchase an office space and work-desks for employees to assist customers and sort queries	30-Mar	05-Apr	10	6	Labour force and office space	Pranav
2.2	Purchase Software	Purchase various servers, host the website and also buy specific tools to customize the website	06-Apr	20-Apr	5	14	Website to buy the respective tools	Pranav
3	Develop website	Work in the final phase of creating and porting the website	21-Apr	30-May	69	39	Web developers and respective languages	Guhan
3.1	Design interface	Create an on-paper design of the website to be implemented	21-Apr	01-May	40	10	Web designers/ UX designers	Guhan
3.2	Build web interface	Create the raw website and the functions it will execute	02-May	15-May	10	13	HTML, CSS, Javascript	Pranav
3.3	Test web interface	Test the website, with a few dummy users and take feedback	16-May	24-May	10	7		Guhan
3.4	Deploy web interface	Do the required changes and port the website online	25-May	30-May	9	5	Server	Pranav

Lab task 3: Analysis of Process Model

Use:

Process models are typically used to represent and analyze a series of activities that occur repeatedly and on a regular basis. Process models can be used to model the flow of work in or between people and departments in an organization, or the flow of activities in a computer system or application.

Advantages:

- This is because the first step in maximising productivity and reducing waste is to know how processes currently run. Process models can lead to:
- **Increased efficiency:** Since the purpose of a process model is to make the process better, it should ultimately result in increased efficiency. With better quality outputs, you can boost your bottom line.
- **Standardisation:** If different teams have to run the same process, having it visually displayed can aid in standardising the process.
- **Transparency:** With process models, everyone becomes aware of what processes exist and the reasons for them. This not only helps to hold responsible parties accountable, but it can also boost morale when employees understand the bigger picture of why they do what they do.
- **Agility:** Creating an environment where process improvement is a norm will help to shift the entire corporate culture. In a business setting that strives for greatness, it becomes easier to enact change management and adjust processes to keep up with the times.

Lab task 4: Class Diagram

Use:

Class diagrams are the blueprints of your system or subsystem. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design.

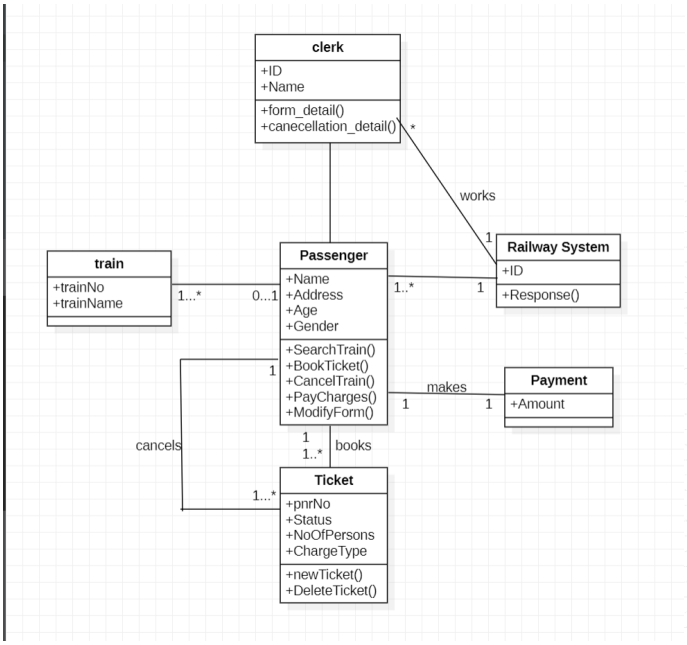
Advantages:

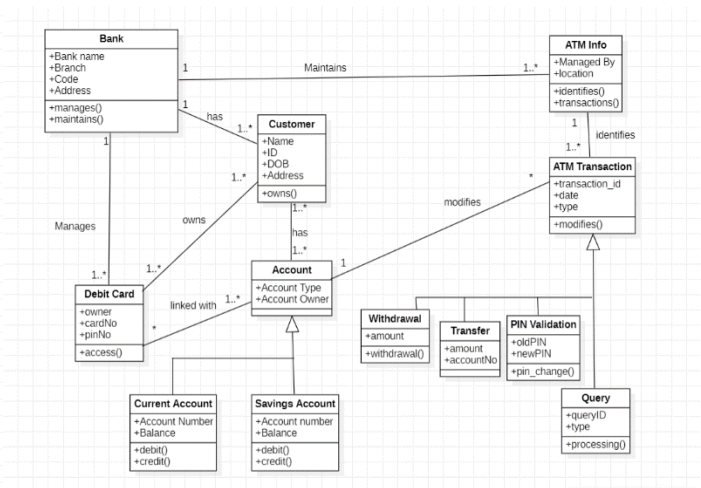
A class diagram could be implemented in different phases of a project and is the heart of the UML. A representation of reality is created by the class diagram by appearing on the domain model during analysis. The software modelling is done during the design phase, whereas the code is generated during the implementation phase. The foundation of software products is the class diagrams which are an essential part of any project.

Notations:

- Box: top section is used to name the class, the second one is used to show the attributes of the class, the third section is used to describe the operations performed by the class.
- Free line: describes the relation between the two classes and also denotes the entity relations such as one-to-one, one-to-many, many- to-one & many-to-many.

Example:





Lab task 5: Use Case Diagram

Use:

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

Advantages:

The reasons why an organization would want to use case diagrams include:

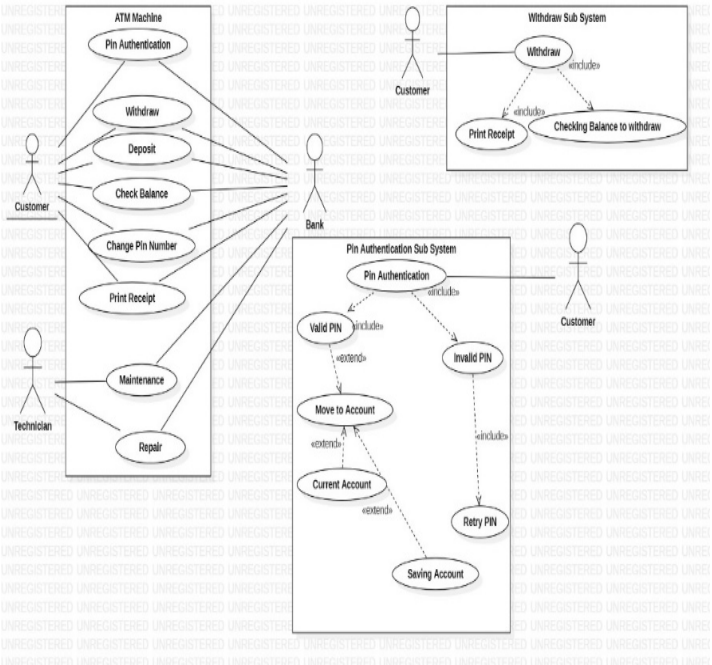
Represent the goals of systems and users.

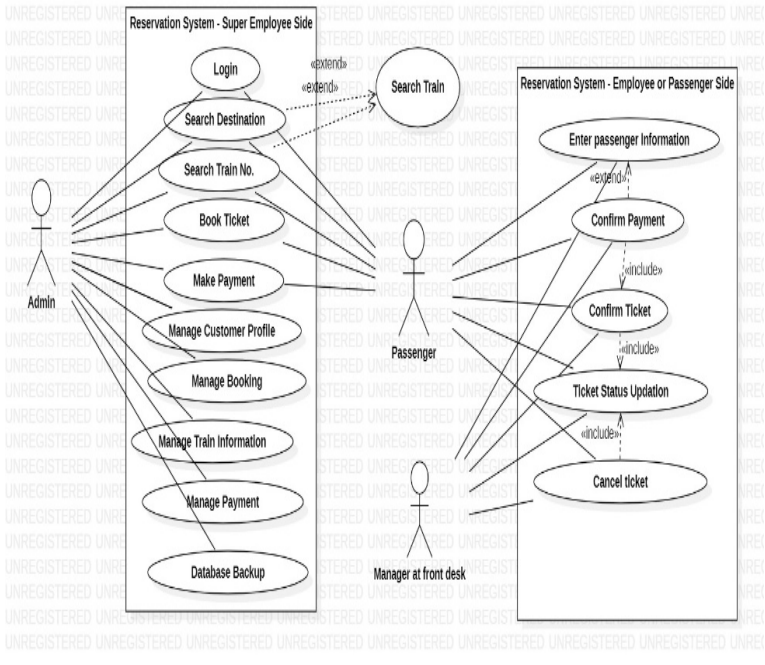
- Specify the context a system should be viewed in.
- Specify system requirements.
- Provide a model for the flow of events when it comes to user interactions.
- Provide an outside view of a system.

Notations:

- Ellipse: denotes the use case
- Arrowhead: shows the direction of control
- Rectangle: use case subject
- Actor: model element that interacts with a system

Example:





Lab task 6: Activity Diagram

Use:

Describe the steps performed in a UML use case. Illustrate a business process or workflow between users and the system. Simplify and improve any process by clarifying complicated use cases. Model software architecture elements, such as method, function, and operation.

Advantages:

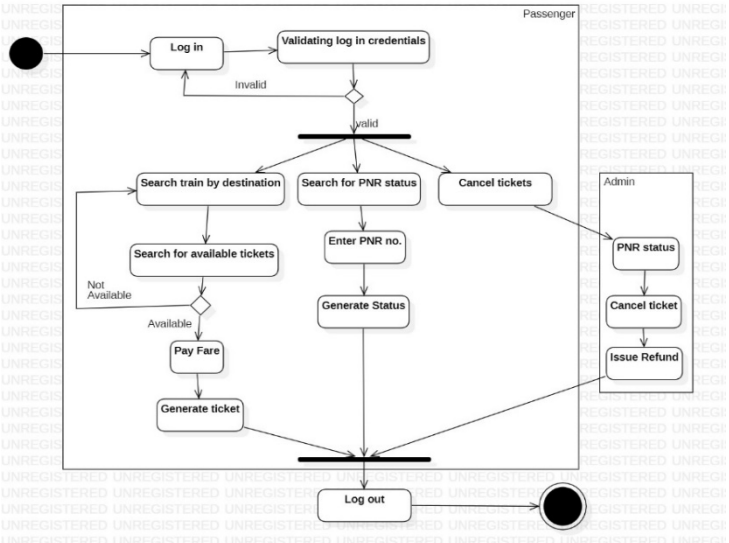
Benefits of activity diagrams:

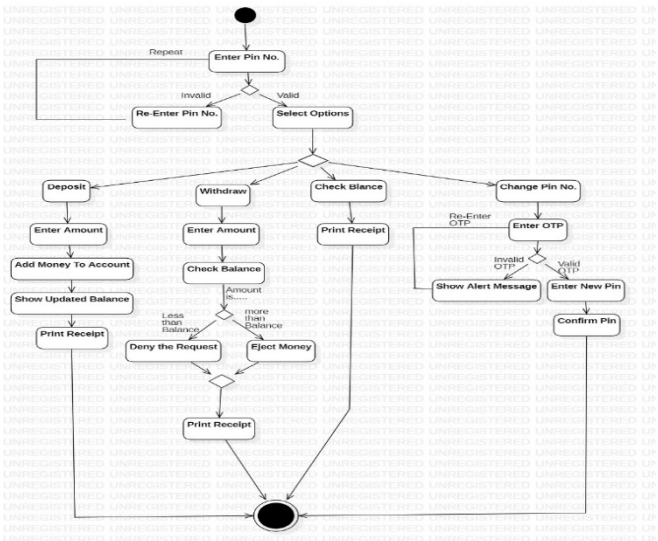
- Demonstrate the logic of an algorithm.
- Describe the steps performed in a UML use case.
- Illustrate a business process or workflow between users and the system.
- Simplify and improve any process by clarifying complicated use cases.

Notations:

- Circle: initial state
- Concentric circles: final state
- Arrowhead: control flow
- Fork: a control node that splits a flow into multiple concurrent flows
- Join: represents the merging of two or more concurrent flows into a single outgoing flow
- Merge: diamond-shaped symbol with two or more edges entering it and a single activity edge leaving it
- Decision: control node that accepts tokens on one or two incoming edges and selects one outgoing edge from one or more outgoing flows

Example:





Lab task 7: Sequence Diagram

Use:

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them.

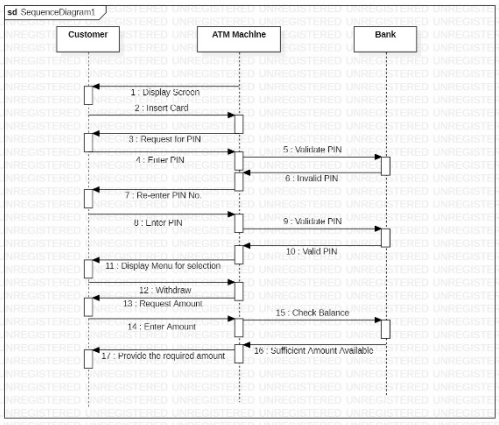
Advantages:

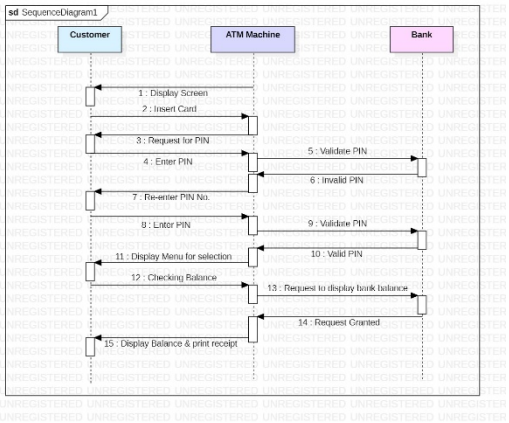
The sequence diagram is a good diagram to use to document a system's requirements and to flush out a system's design. The reason the sequence diagram is so useful is because it shows the interaction logic between the objects in the system in the time order that the interactions take place.

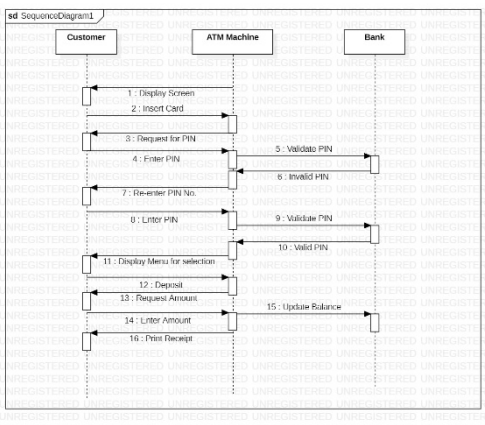
Notations:

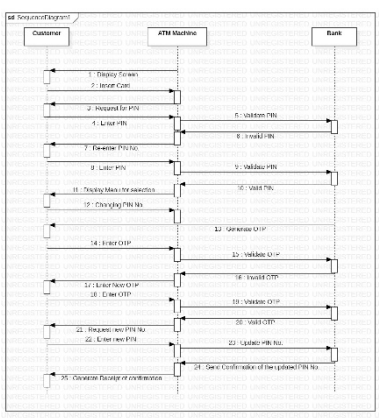
- Lifeline: represents the passage of time as it extends downward
- Message: a communication between objects that conveys information with the expectation that action will ensue
- Self-Message: reflects a new process or method invoked within the calling lifeline's operation
- Reply Message: represented by a dashed line with a lined arrowhead, these messages are replies to calls

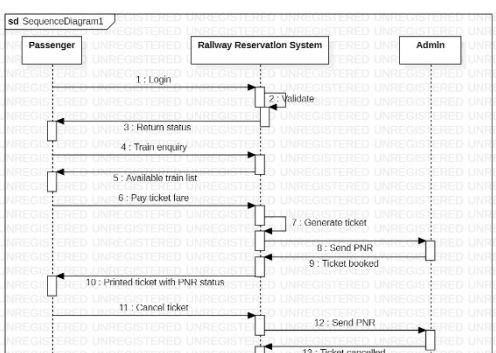
Example:











Lab task 8: State Chart Diagram

Use:

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

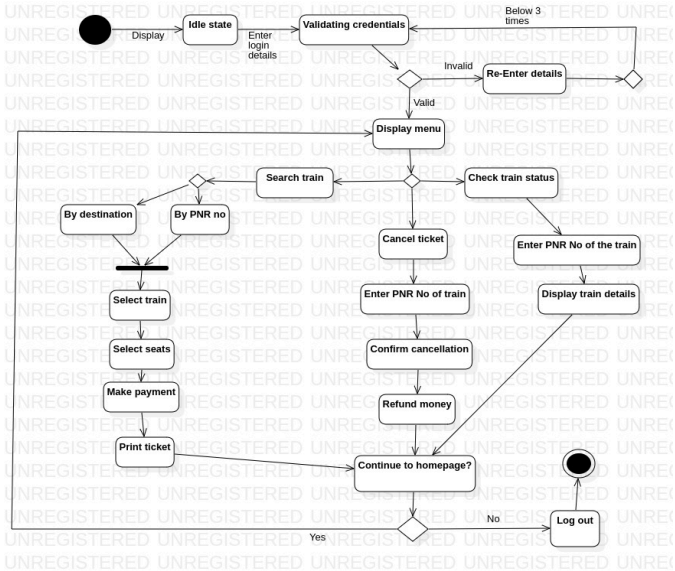
Advantages:

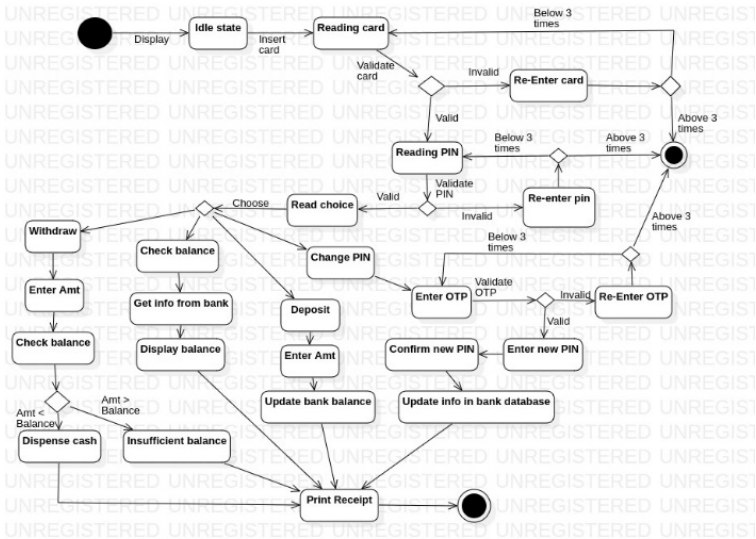
State diagrams enable you to describe the behaviour of objects during their entire life span. In addition, the different states and state changes as well as events causing transitions can be described. On other words: State diagrams make the system behaviour visible.

Notations:

- Simple State
- Circle: initial state
- Concentric circles: final state
- Choice: used to select a particular option from a list of options
- Join: represents the merging of two or more concurrent flows into a single outgoing flow
- Fork: a control node that splits a flow into multiple concurrent flows

Example:





Lab task 9: Dataflow Diagram

Use:

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

Advantages:

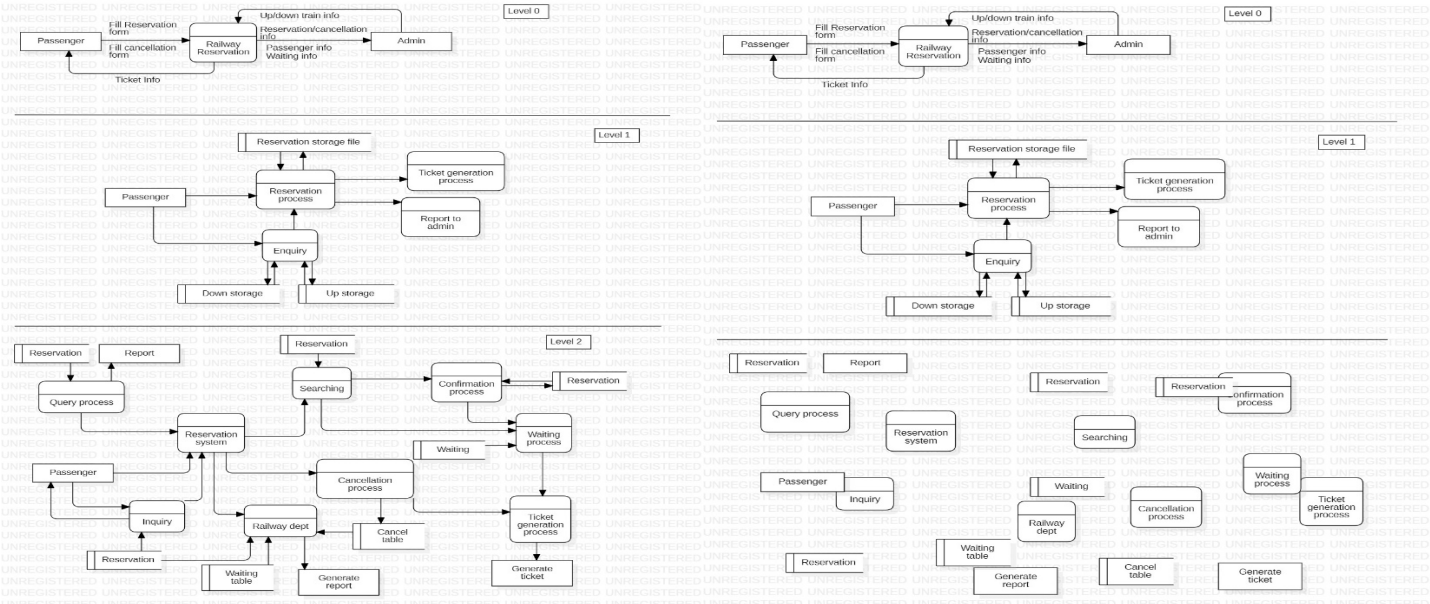
Advantages of data flow diagram:

- It aids in describing the boundaries of the system.
- It is beneficial for communicating existing system knowledge to the users.
- A straightforward graphical technique which is easy to recognise.
- DFDs can provide a detailed representation of system components.

Notations:

- External Entity: represents sources of data to the system or destinations of data from the system
- Process: process (function, transformation) is part of a system that transforms inputs to outputs
- Data Store: represents data that is not moving (delayed data at rest)
- Data Flow: indicates the direction of flow

Example:



Lab task 10: ER Diagram

Use:

Database troubleshooting: ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it's going wrong. Business information systems: The diagrams are used to design or analyze relational databases used in business processes.

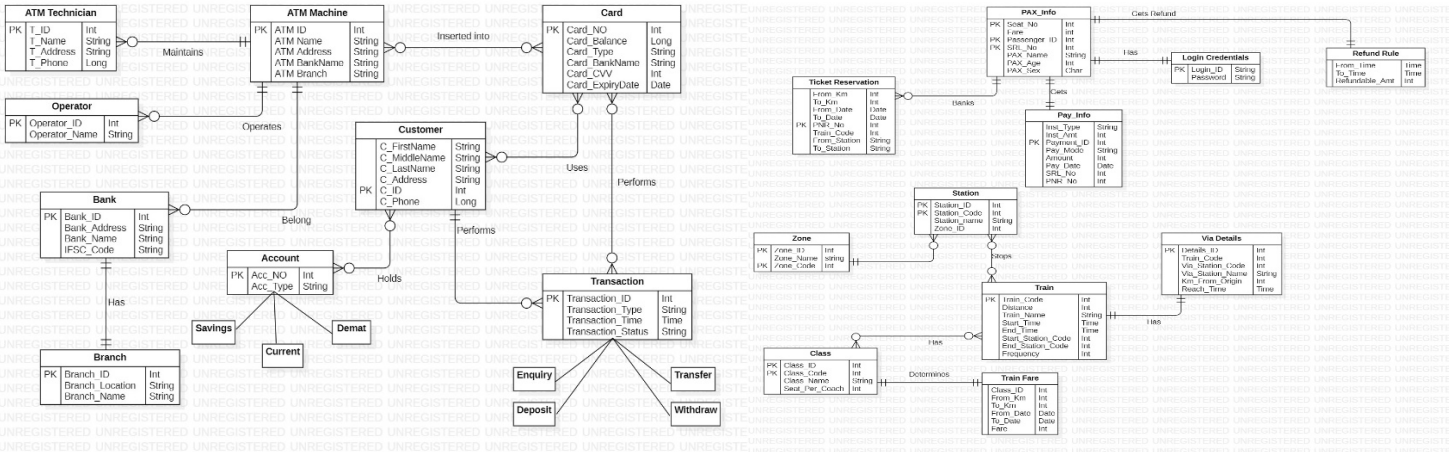
Advantages:

- **Conceptually it is very simple:** ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.
- **Better visual representation:** ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.
- **Effective communication tool:** It is an effective communication tool for database designer.
- **Highly integrated with relational model:** ER model can be easily converted into relational model by simply converting ER model into tables.
- **Easy conversion to any data model:** ER model can be easily converted into another data model like hierarchical data model, network data model and so on.

Notations:

- Rectangles: represents entity types
- Relationships: One-to-one, one-to-many, many-to-many

Example:



Lab task 11: JUnit Testing

Use:

JUnit testing is used to test the behavior of methods inside classes we have written. We test a method for the expected results and sometimes exception-throwing cases—whether the method is able to handle the exceptions in the way we want.

Advantages:

Advantages of JUnit:

- The JUnit framework is open source.
- It provides text-based command lines as well as AWT-based and Swing-based graphical test mechanisms.
- It has some annotations to utilize test functions.
- It has a test runner to test running applications.
- It allows you to write code.

Example:

Sample code:

```
public class Calculation {

   public static int findMax(int arr[]){
      int max=0;
      //int max=arr[0];
      for(int i=1;i<arr.length;i++){
         if(max<arr[i])
            max=arr[i];
      }
      return max;
   }
}
```

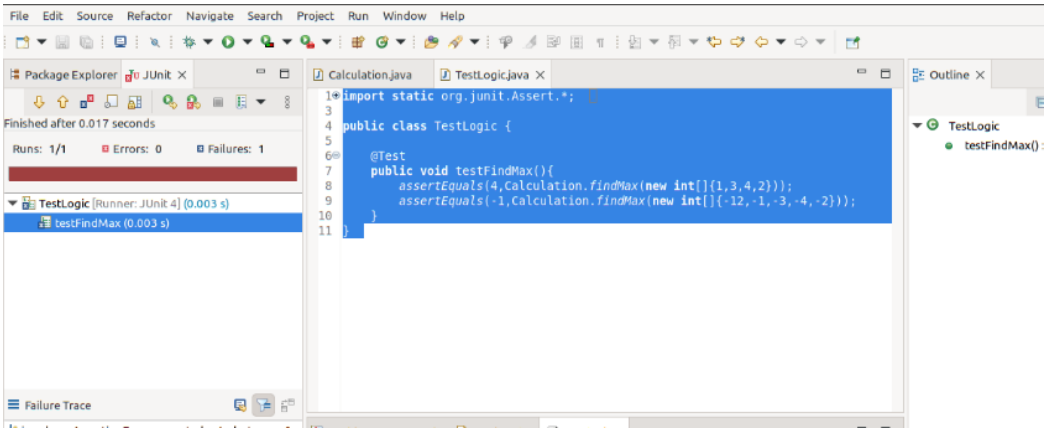
Junit code:

```
import static org.junit.Assert.*;
import org.junit.Test;

public class TestLogic {

   @Test
   public void testFindMax(){
      assertEquals(4,Calculation.findMax(new int[]{1,3,4,2}));
      assertEquals(-1,Calculation.findMax(new int[]{-12,-1,-3,-4,-2}));
   }
}
```

failure example:



Passing example: