



DEPARTMENT OF  
COMPUTER SCIENCE AND ENGINEERING

---

**Title: Develop System Architecture (Part II: Low Level Design) for a specific project**

---

INTEGRATED DESIGN PROJECT II  
CSE 406



GREEN UNIVERSITY OF BANGLADESH

---

# 1 Objective(s)

- To define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other of a system.
- To design low level architecture for the given project (Part II).
- To understand the motive of low level design for a project.

## 1.1 Sub-Objective(s)

- To describe stakeholders' communication.
- To describe the functionalities performed by the system.
- To reuse the same components for systems with similar requirements in large scale.
- To represent the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them.

# 2 Problem analysis

Architectural design is concerned with understanding how a system should be organized and designing the overall structure of that system. In the model of the software development process, architectural design is the first stage in the software design process. It is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them. The output of the architectural design process is an architectural model that describes how the system is organized as a set of communicating components.

In practice, there is a significant overlap between the processes of requirements engineering and architectural design. Ideally, a system specification should not include any design information. This is unrealistic except for very small systems. Architectural decomposition is usually necessary to structure and organize the specification. Therefore, as part of the requirements engineering process, you might propose an abstract system architecture where you associate groups of system functions or features with large-scale components or sub-systems. You can then use this decomposition to discuss the requirements and features of the system with stakeholders.

Architectural design is a creative process where you design a system organization that will satisfy the functional and non-functional requirements of a system. Because it is a creative process, the activities within the process depend on the type of system being developed, the background and experience of the system architect, and the specific requirements for the system. It is therefore useful to think of architectural design as a series of decisions to be made rather than a sequence of activities.

During the architectural design process, system architects have to make a number of structural decisions that profoundly affect the system and its development process. Based on their knowledge and experience, they have to consider the following fundamental questions about the system:

- How will the system be distributed across a number of cores or processors?
- What architectural patterns or styles might be used?
- What will be the fundamental approach used to structure the system?
- How will the structural components in the system be decomposed into subcomponents?
- What strategy will be used to control the operation of the components in the system?
- What architectural organization is best for delivering the non-functional requirements of the system?
- How will the architectural design be evaluated?
- How should the architecture of the system be documented?

**Problem statement and motivation for the given project:** The project entitled Banking ATM system has a drastic change to that of the older version of banking system, customer feel inconvenient with the transaction method as it was in the hands of the bank employees. In our ATM system, the above problem is overcome here, the transactions are done in person by the customer thus makes the customers feel safe and

---

secure. Thus the application of our system helps the customer in withdrawing money, checking the balance and transaction of the amount with mini-statement and transferring the balance by validating the pin number therefore ATM system is more user friendly.

### 3 Methodology

You can design software architectures at two levels of abstraction, which we call architecture in the small and architecture in the large:

1. Architecture in the small is concerned with the architecture of individual programs. At this level, we are concerned with the way that an individual program is decomposed into components. This chapter is mostly concerned with program architectures.
2. Architecture in the large is concerned with the architecture of complex enterprise systems that include other systems, programs, and program components. These enterprise systems are distributed over different computers, which may be owned and managed by different companies.

The apparent contradictions between practice and architectural theory arise because there are two ways in which an architectural model of a program is used:

1. Low Level Design (Detailed Design): As a way of documenting an architecture that has been designed. The aim here is to produce a complete system model that shows the different components in a system, their interfaces, and their connections. The argument for this is that such a detailed architectural description makes it easier to understand and evolve the system.
2. High Level Design: As a way of facilitating discussion about the system design. A high-level architectural view of a system is useful for communication with system stakeholders and project planning because it is not cluttered with detail. Stakeholders can relate to it and understand an abstract view of the system. They can then discuss the system as a whole without being confused by detail. The architectural model identifies the key components that are to be developed so managers can start assigning people to plan the development of these systems.

#### 3.1 Architectural views

It is impossible to represent all relevant information about a system's architecture in a single architectural model, as each model only shows one view or perspective of the system. It might show how a system is decomposed into modules, how the run-time processes interact, or the different ways in which system components are distributed across a network. All of these are useful at different times so, for both design and documentation, you usually need to present multiple views of the software architecture.

1. A logical view, which shows the key abstractions in the system as objects or object classes. It should be possible to relate the system requirements to entities in this logical view.
2. A process view, which shows how, at run-time, the system is composed of interacting processes. This view is useful for making judgments about nonfunctional system characteristics such as performance and availability.
3. A development view, which shows how the software is decomposed for development, that is, it shows the breakdown of the software into components that are implemented by a single developer or development team. This view is useful for software managers and programmers.
4. A physical view, which shows the system hardware and how software components are distributed across the processors in the system. This view is useful for systems engineers planning a system deployment.

#### 3.2 Required Software

There are several software available that can be used online and offline to draw these system architectures like as follows.

1. Visual Paradigm for UML 8.2 (online link: <https://online.visual-paradigm.com/>)
2. StartUML
3. Lucidchart and other drawing tools

---

## 4 Implementation

### 4.1 Low Level Design of The Banking ATM System

#### 4.1.1 Low Level Design of The Banking ATM System Based on Transaction Processing Application Architecture

Low Level Design in short LLD is like detailing HLD means it refers to component-level design process. It describes detailed description of each and every module means it includes actual logic for every system component and it goes deep into each modules specification. It is also known as micro level/detailed design. It is created by designers and developers. It converts the High Level Solution into Detailed solution. It is created second means after High Level Design. The low level design of system architecture of the Banking ATM system based on Transaction Processing Application architecture is shown in the following figures.

#### 4.1.2 Low Level Design of The Banking ATM System Using Abstract View of Context

The Fig.1 shows the trends to high level view to low level view using context based architecture. It shows how every part of a high level design can be converted to a detailed design. In the detained design all the functions are shown in very transparent way.

### 4.2 Part I (Low Level Design of The Banking ATM System)

The Fig.2, Fig.3, Fig.4, Fig.5, Fig.6, Fig.7 and Fig.8 show the low level view (detailed design) of the different sub modules of the ATM system using context based architecture.

You can go through the following link to find the detailed explanation of the high level and low level architecture of the Banking ATM system.

<https://link.springer.com/content/pdf/bbm%3A978-3-642-56209-9%2F1.pdf>

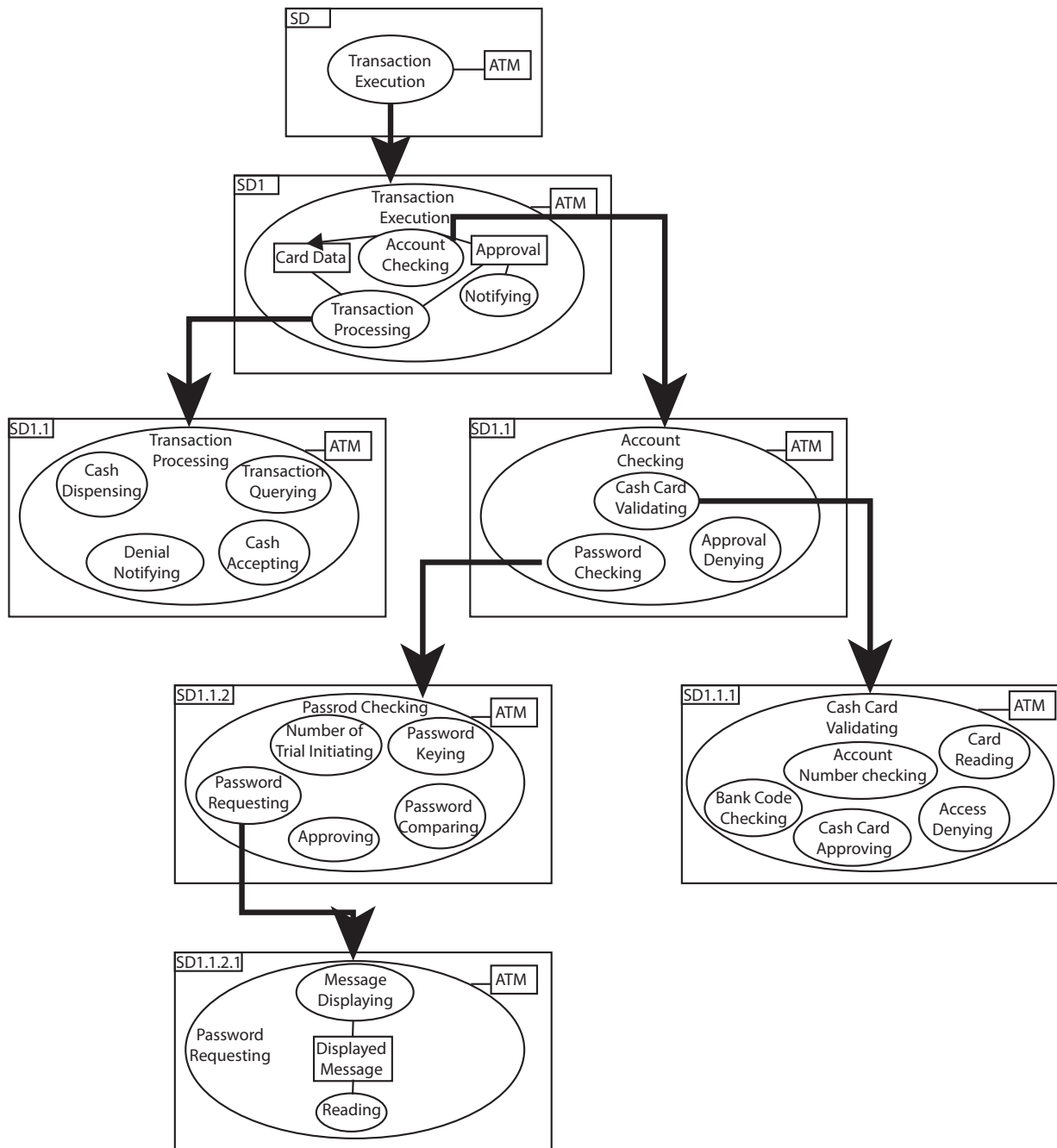


Figure 1: Trends to high level view to low level view using context based architecture of A Banking ATM system

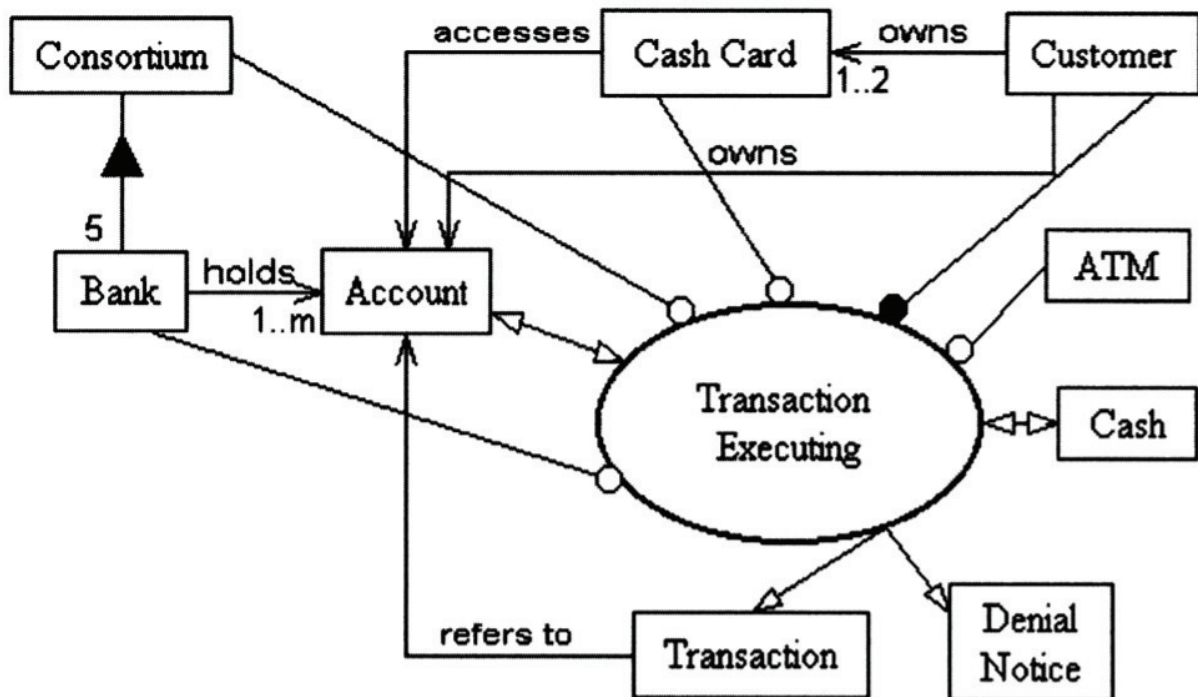


Figure 2: SD- Low level view of the system architecture for the SD (system diagram) of the ATM system

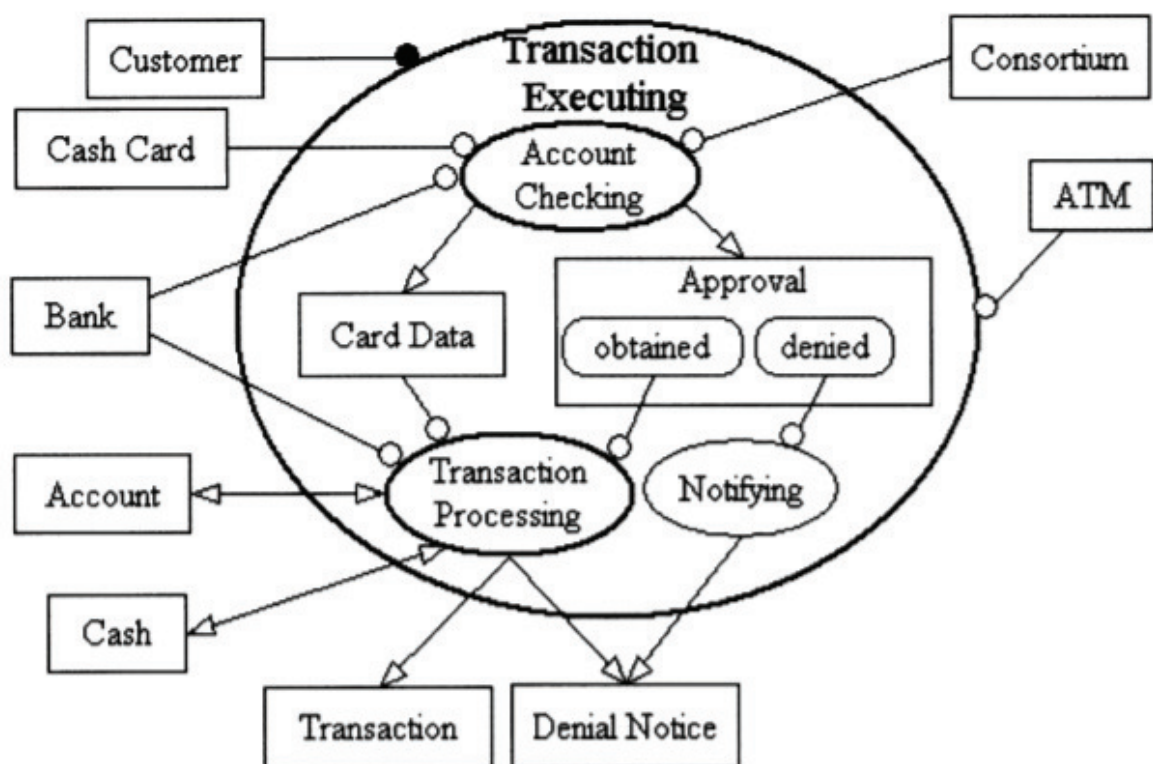


Figure 3: SD1- Low level view of the system architecture for Transaction Executing from SD zooms in SD1 into Account Checking, Transaction Processing, and Notifying, as well as Approval and Card Data

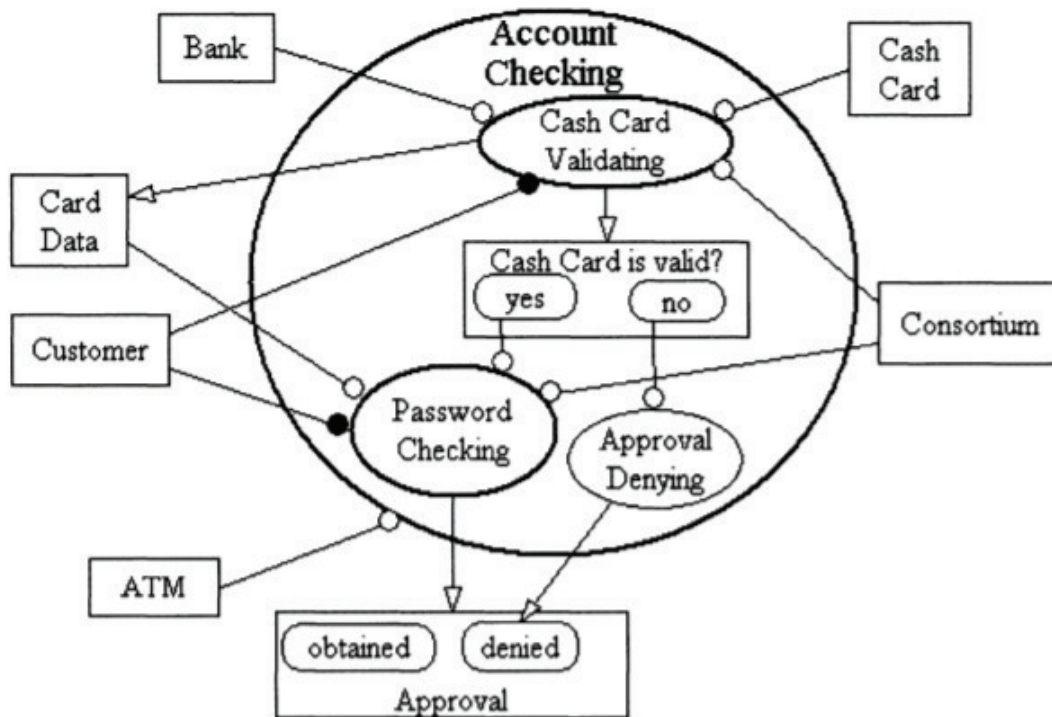


Figure 4: SD1.1- Low level view of the system architecture for Account Checking from SD1 zooms in SD1.1 into Cash Card Validating, Password Checking, and Approval Denying, as well as "Cash Card is Valid?"

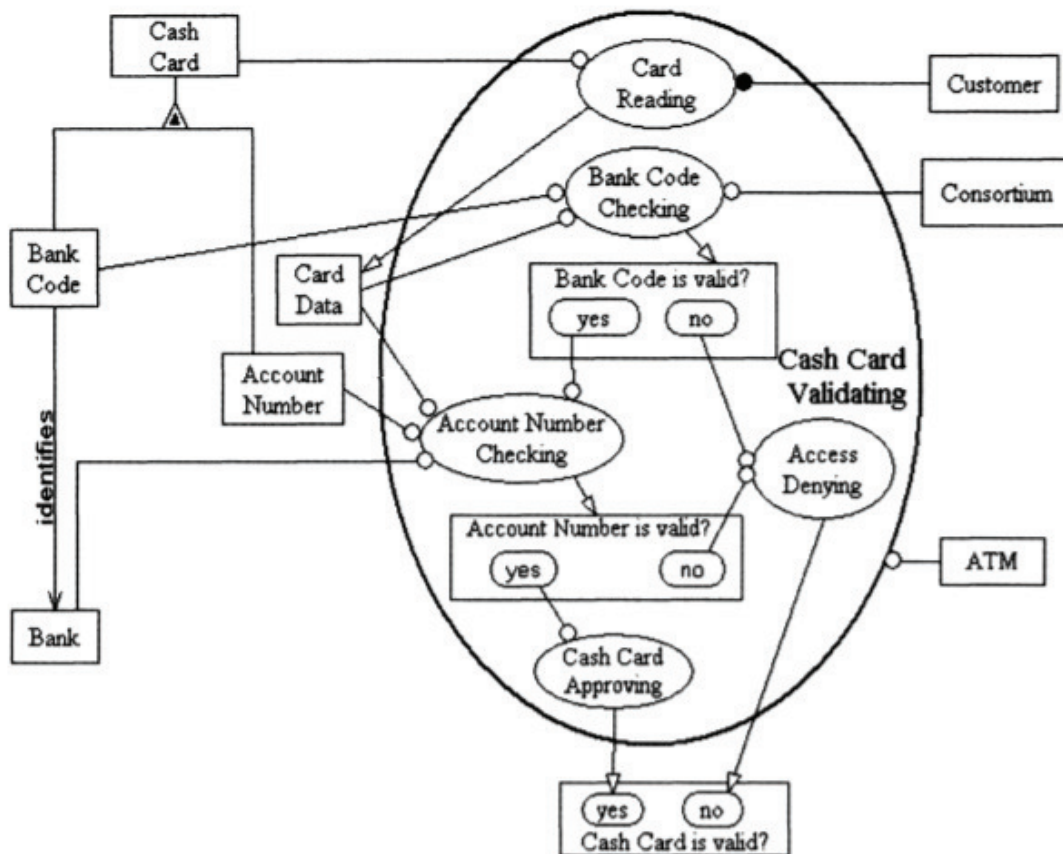


Figure 5: SD1.1.1- Low level view of the system architecture for the operation of Cash Card from SD1.1 unfolds in SD1.1.1 to exhibit Bank Code and Account Number



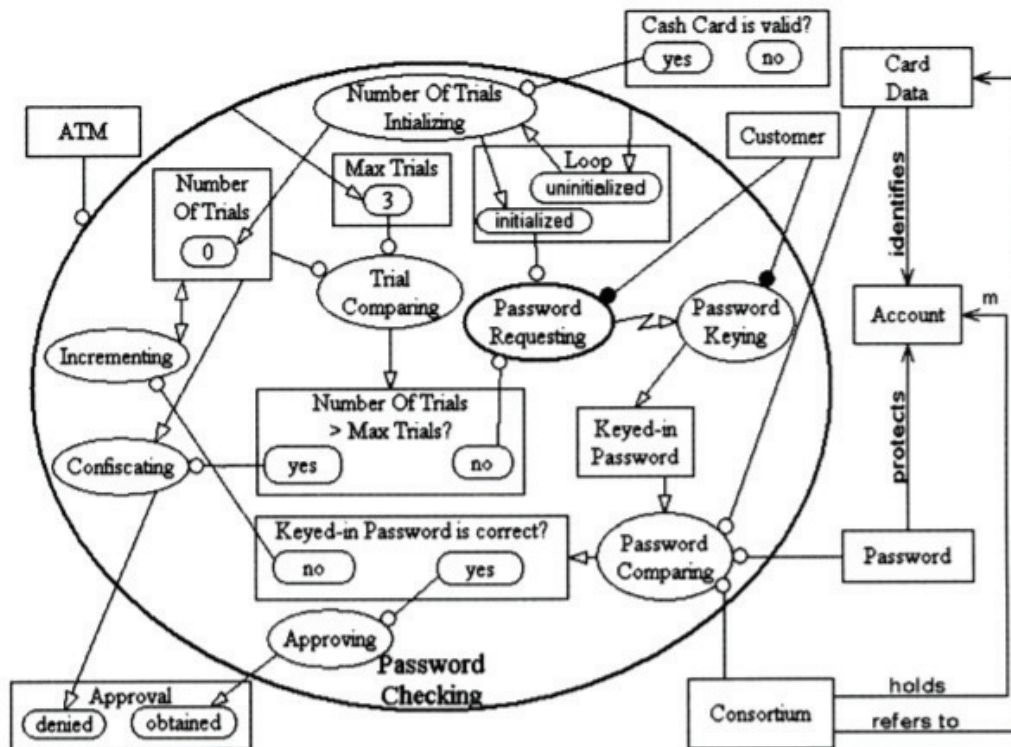


Figure 6: SD1.1.2- Low level view of the system architecture for the operation of the Account from SD1.1 unfolds in SD1.1.2 to relate to Password

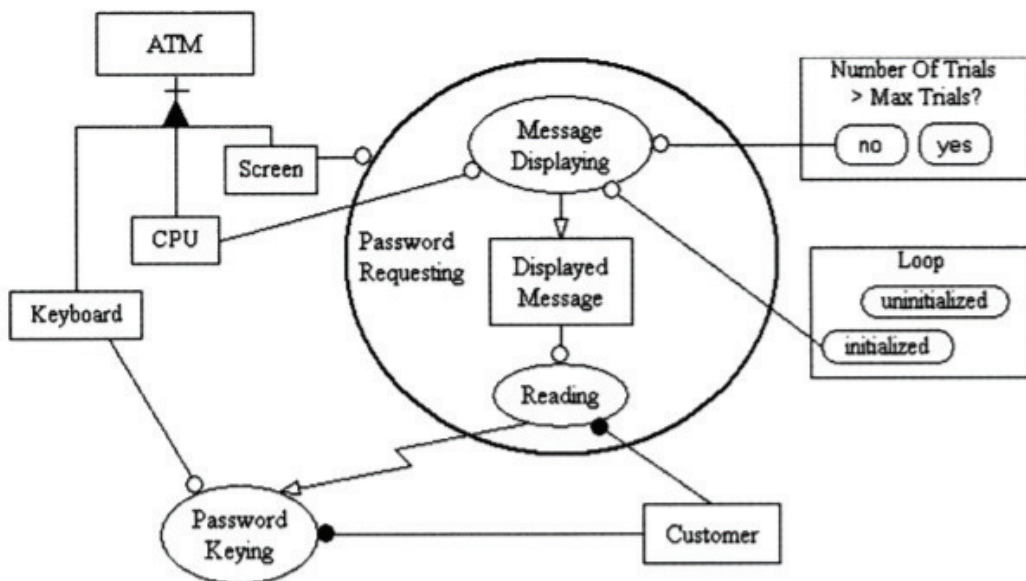


Figure 7: SD1.1.2.1- Low level view of the system architecture for the operation of the Password Requesting from SD1.1.2 zooms in SD1.1.2.1 into Message Oisplaying and Reading, as weil as Oisplayed Message



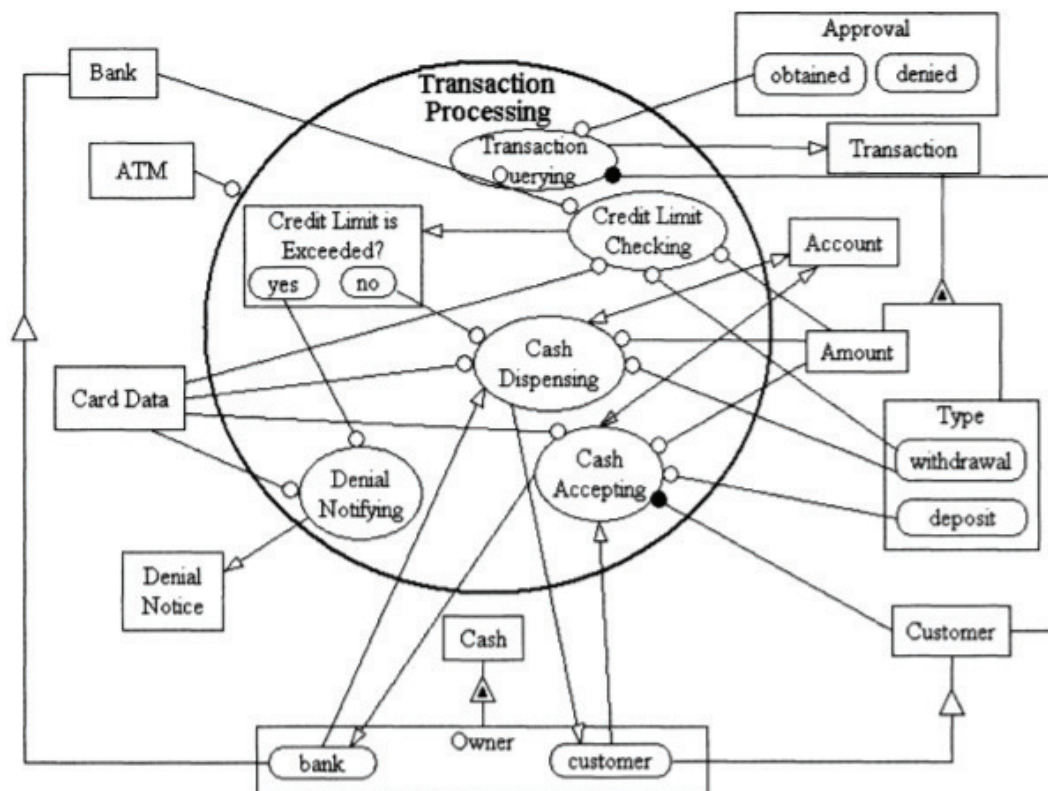


Figure 8: SD1.2- Low level view of the system architecture for the Transaction from SD1 unfolds in SD1.2 to exhibit Amount and Type

## 5 Discussion & Conclusion

Based on the focused objective(s) to understand about low level design of the system architectures including different patterns and views, the additional lab exercise made me more confident towards the fulfilment of the objectives(s).

## 6 Lab Task (Please implement yourself and show the output to the instructor)

1. Draw a low level architecture (detailed view) of the given project 'Library Management System'.

### 6.1 Problem analysis

A library database system is an infrastructure that allows users to search books and book content, add/remove, and download selected books. The problem faced is that library users require an efficient method to find a specific book or keyword(s) within a book given a continuously expanding library. Some scenarios of the 'Library Management System' are as follows:

1. User who registers himself as a new user initially is regarded as staff or student for the library system.
  - (i) For the user to get registered as a new user, registration forms are available that is needed to be fulfilled by the user.
  - (ii) After registration, a library card is issued to the user by the librarian. On the library card, an ID is assigned to cardholder or user.
2. After getting the library card, a new book is requested by the user as per their requirement.
3. After, requesting, the desired book or the requested book is reserved by the user that means no other user can request for that book.
4. Now, the user can renew a book that means the user can get a new due date for the desired book if the user has renewed them.

- 
5. If the user somehow forgets to return the book before the due date, then the user pays fine. Or if the user forgets to renew the book till the due date, then the book will be overdue and the user pays fine.
  6. User can fill the feedback form available if they want to.
  7. Librarian has a key role in this system. Librarian adds the records in the library database about each student or user every time issuing the book or returning the book, or paying fine.
  8. Librarian also deletes the record of a particular student if the student leaves the college or passed out from the college. If the book no longer exists in the library, then the record of the particular book is also deleted.
  9. Updating database is the important role of Librarian.

## 7 Policy

Copying from internet, classmate, seniors, or from any other source is strongly prohibited. 100% marks will be *deducted* if any such copying is detected.