Project Guide

Java Application Development

Project Guide - Java Application Development

APTECH LIMITED
Contact E-mail: ov-support@onlinevarsity.com

Edition 1 - 2013

Dear Learner,

We congratulate you on your decision to pursue an Aptech Worldwide course.

Aptech Ltd. designs its courses using a sound instructional design model – from conceptualization to execution-incorporating the following key aspects: –

- Scanning the user system and needs assessment

Needs assessment is carried out to find the educational and training needs of the learner.

Technology trends are regularly scanned and tracked by core teams at Aptech Ltd. TAG* analyzes these on a monthly basis to understand the emerging technology training needs for the Industry.

An annual Industry Recruitment Profile Survey# is conducted during August - October to understand the technologies that Industries would be adapting in the next 2 to 3 years. An analysis of these trends and recruitment needs is then carried out to understand the skill requirements for different roles and career opportunities.

The skill requirements are then mapped with the learner profile (user system) to derive the learning objectives for the different roles.

- Needs analysis and design of curriculum

The learning objectives are then analyzed and translated into learning tasks. Each learning task or activity is analyzed in terms of knowledge, skills and attitudes that are required to perform that task. Teachers and domain experts do this jointly. These are then grouped in clusters to form the subjects to be covered by the curriculum.

In addition, the society, the teachers, and the industry expect certain knowledge and skills that are related to abilities such as learning-to-learn, thinking, adaptability, problem solving, positive attitude etc. These competencies would cover both cognitive and affective domains.

A precedence diagram for the subjects is drawn, where the prerequisites for each subject are graphically illustrated. The number of levels in this diagram is determined by the duration of the course in terms of number of semesters etc. Using the precedence diagram and the time duration for each subject, the curriculum is organized.

- Design and development of instructional materials

The content outlines are developed by including additional topics that are required for the completion of the domain and for the logical development of the competencies identified. evaluation strategy and scheme is developed for the subject. The topics are arranged/organized in a meaningful sequence.

The detailed instructional material – Training aids, Learner material, reference material, project guidelines, etc. are then developed. Rigorous quality checks are conducted at every stage.

- Strategies for delivery of instruction

Careful consideration is given for the integral development of abilities like thinking, problem solving, learning-to-learn etc. by selecting appropriate instructional strategies (training methodology), instructional activities and instructional materials.

The area of IT is fast changing and nebulous. Hence considerable flexibility is provided in the instructional process by specially including creative activities with group interaction between the students and the trainer. The positive aspects of web based learning –acquiring information, organizing information and acting on the basis of insufficient information are some of the aspects which are incorporated in the instructional process.

- Assessment of learning

The learning is assessed through different modes – tests, assignments and projects. The assessment system is designed to evaluate the level of knowledge and skills as defined by the learning objectives.

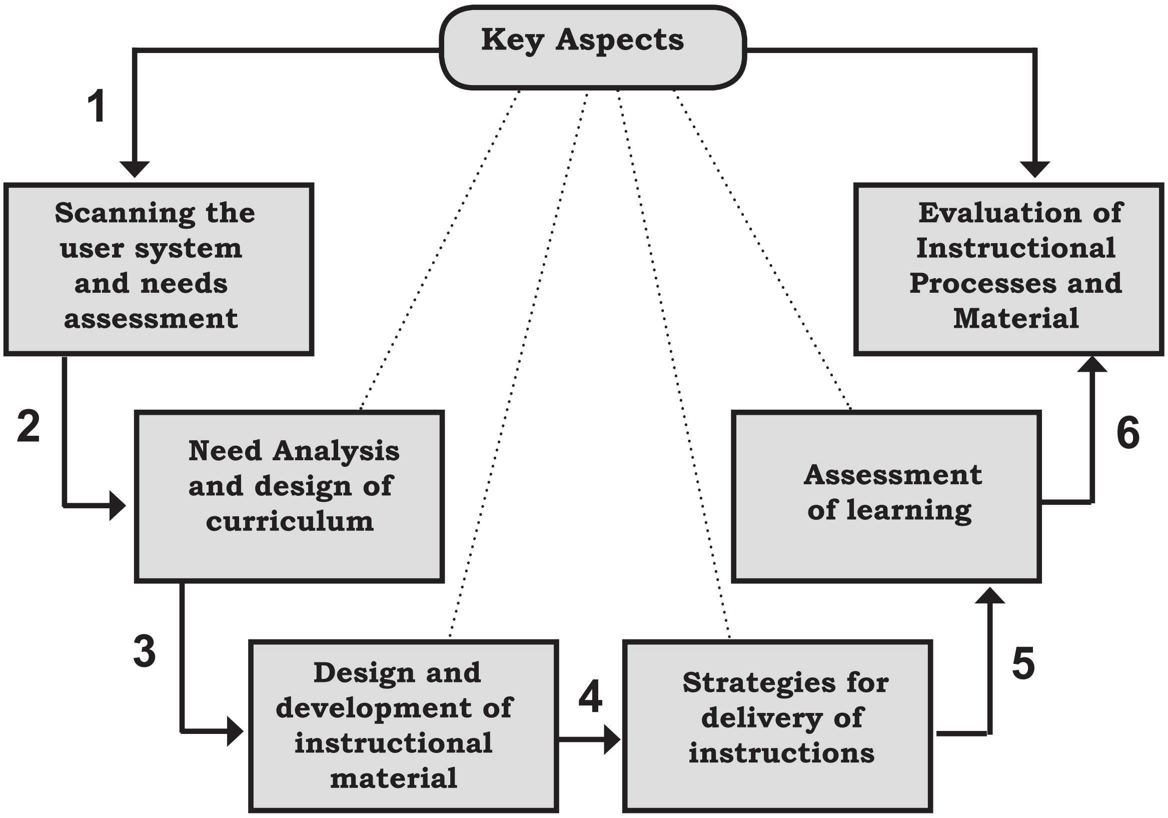
- Evaluation of instructional process and instructional materials

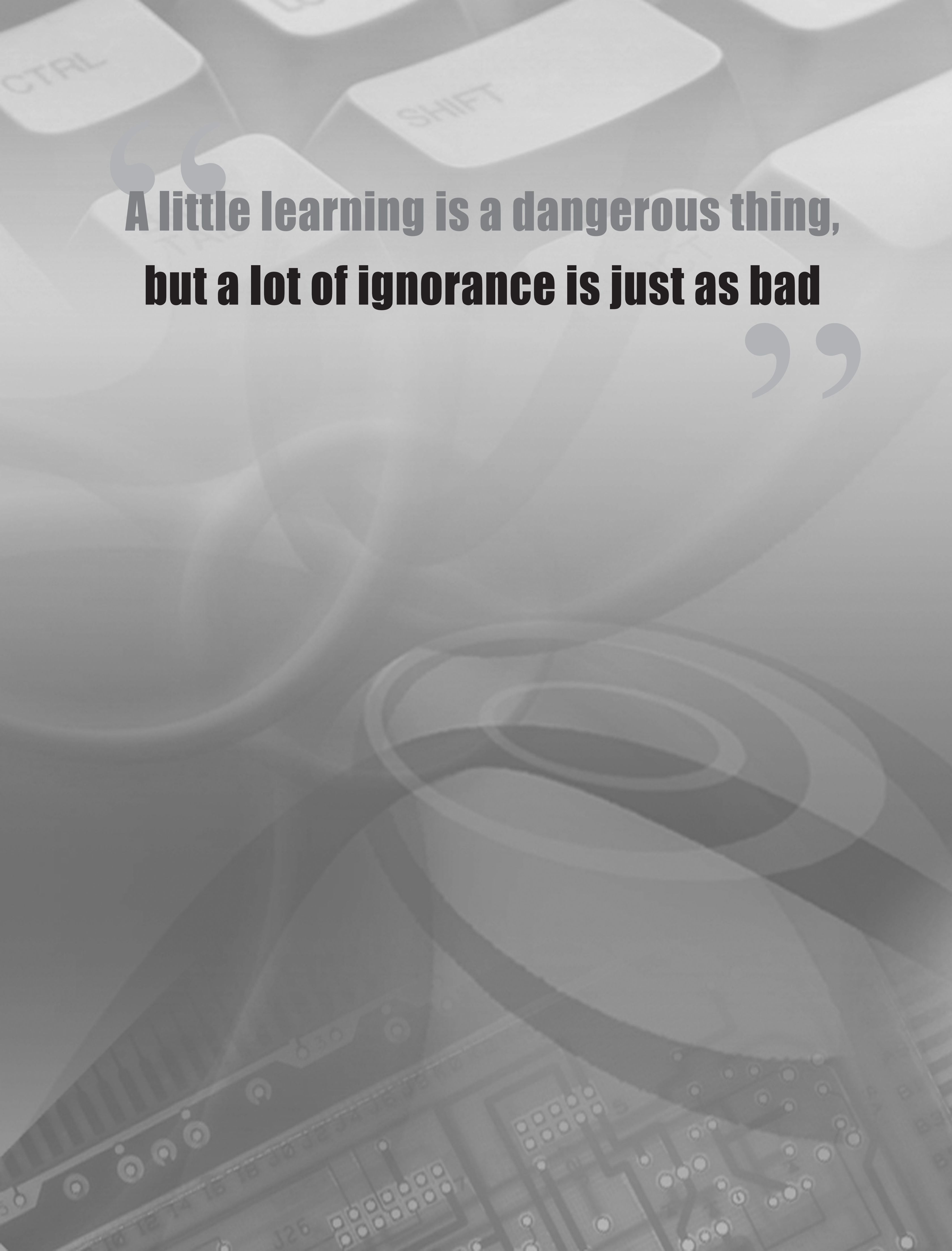
The instructional process is backed by an elaborate monitoring system to evaluate - on-time delivery, understanding of a subject module, ability of the instructor to impart learning. As an integral part of this process, we request you to kindly send us your feedback in the reply pre-paid form appended at the end of each module.

*TAG – Technology and Academics Group comprises members from Aptech Ltd., professors from reputed academic institutions, senior managers from industry, technical gurus from software majors and representatives from regulatory organizations/forums.

Technology heads of Aptech Ltd. meet on a monthly basis to share and evaluate the technology trends. The group interfaces with the representatives of the TAG thrice a year to review and validate the technology and academic directions and endeavors of Aptech Ltd.

Aptech New Products Design Model


Key Aspects
1
Scanning the user system and needs assessment
2
Need Analysis and design of curriculum
3
Design and development of instructional material
4
Strategies for delivery of instructions
5
Assessment of learning
6
Evaluation of Instructional Processes and Material



“A little learning is a dangerous thing,

but a lot of ignorance is just as bad”

Preface

Today, the influence of IT spans over a wide area of Commerce and Industry. The process started with the conversion of support systems like payroll, inventory, accounts management, HR management, and so on.

This Project Guide provides all the necessary information to students in developing a Java application that will help retailers manage their business by enabling them to add, modify and view customer and product details.

Students have to understand the workflow and design the project accordingly.

The best way to learn something is to apply its principles and test it. Similarly, the best way to evaluate the tool knowledge of the students is to apply it through project work. The degree of success of the project depends on the strength of its Guide. This Project Guide has been prepared following the best practices in the Industry and helps you to have the experience of going through a LIVE project. It teaches you the essentials of successful development of IT projects.

The Project Guide will help you to:

- Analyze a project
- Design the specifications of the project
- Develop the solution
- Maintain disciplined documentation for the work done

This Project Guide reiterates the commitment of Aptech in keeping up its tradition of providing innovative, career oriented professional education. This ensures that modules are based on the Project Based Learning concept.

Religiously following the given systematic approaches in this book would prepare you to get the real life experience of handling projects. This is because the practices listed here have been extracted from the current industry norms. Thus, such an exercise would prepare you for joining the Software Development Industry.

The knowledge and information in this book is the result of a concerted effort of the Design Team, which is continuously striving to bring you the best and the latest in Information Technology. The process of Design has been a part of the ISO 9001 certification for Aptech-IT Division, Education Support Services. As a part of Aptech's quality drive, this team does extensive research and curriculum enrichment to keep it in line with industry trends.

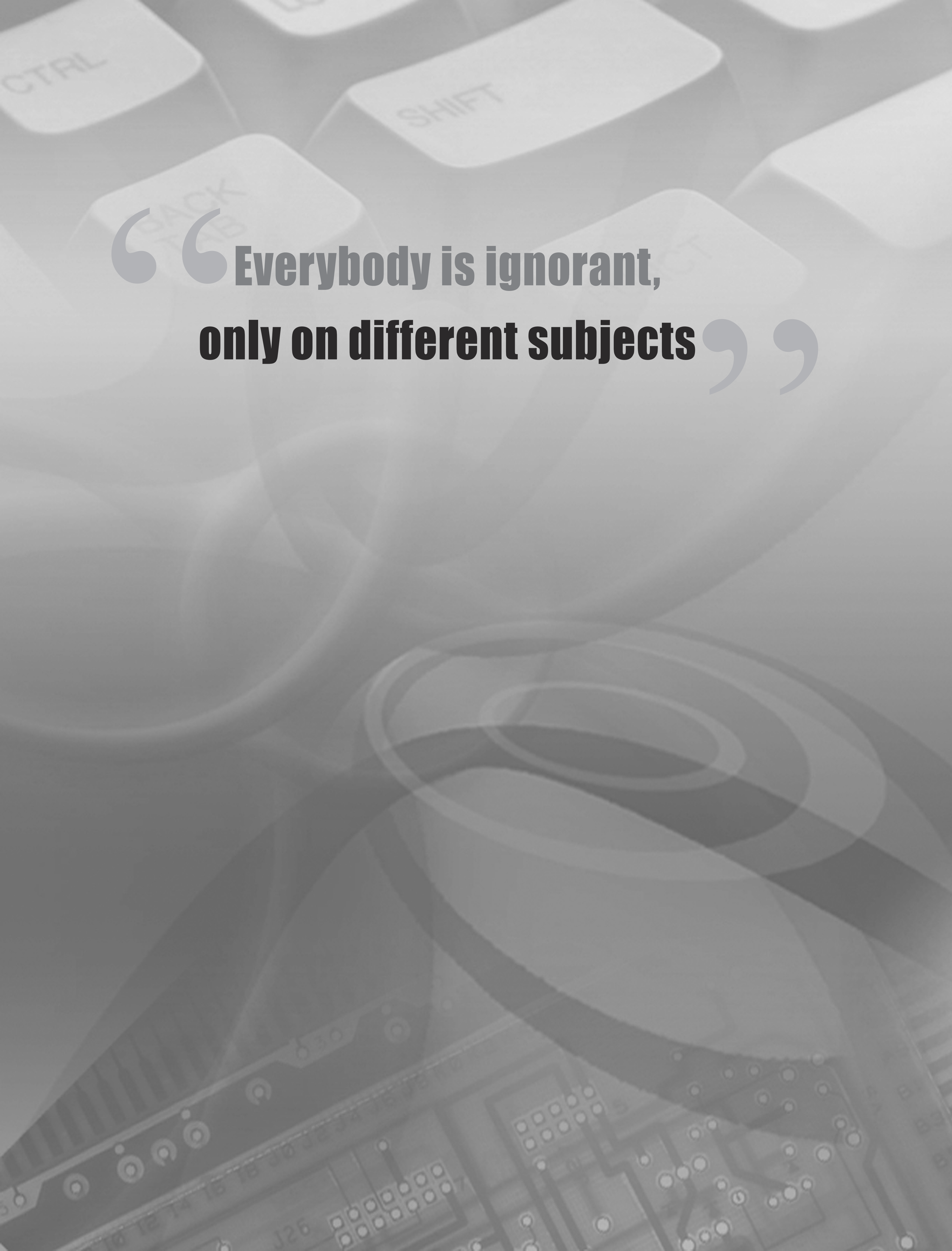
We will be glad to receive your suggestions. Wishing you the very best.

Design Team

“Learning is a treasure that will follow its owner everywhere”

Contents

“Everybody is ignorant,
only on different subjects”



Java Application Development

1.1 How to effectively use this Project Guide

In this project based learning approach an entire project has to be developed throughout the lab sessions. The modules are based on a case study. You will code the solution for that case study throughout the labs of all the modules. Therefore, it is important to complete thoroughly and successfully the labs of all the modules. To do this, it is necessary to refer to the Project Guide as and when required.

The following guidelines will help to understand how to use Project Guide throughout your course:

1. Read the Case Study and understand the business scenario
2. Solve the lab questions so that different components of the application are ready
3. Document the design of the corresponding Process, Component, Interface, and Function as soon as you complete your lab exercise
4. Check whether your code follows the common standards
5. Identify the areas where common coding standards can be implemented and design the common standards for coding
6. Integrate the entire project that you have developed throughout all lab sessions

1.2 Case Study

With the rise in retail business, every retailer has to maintain a single record of every product whether it is being bought or sold. In addition, their operations have to be efficient and documented properly. Many organizations are deploying industry standard quality management methodologies based on ISO 9001 and /or any other quality standards relevant to their type of industry. There is a need to offer a software solution to offset cost by substituting paper based expenses with increased control, efficiency, productivity, and live documentation.

This software solution will enable to add, modify, and view customer details and product details. It will also help to find out product sales and transactions. Also, the software should be capable of storage, retrieval, and manipulation of data in appropriate database files.

The users for this system are retailers.

The application should implement the following functions:

- The ‘Admin‘ is the person who will be operating from the Retailer’s Shop. Admin can register himself/herself with the application. This registration will enable him/her to log in to the system and use the application.

- Create an Admin file that will store the login details of the Admin person such as Login ID, Name, Organization ID, and Password.

- Customers are the people on whose behalf the ‘Admin/Retailer’ buys or sells products. Before doing any transaction, the customer details should be present in the application. Retailer person is responsible for getting all the customer details and adding it to the customer file.

- Retailer is responsible for getting all the customer details and adding it to the customer file.

- Only an ‘Admin/Retailer’ user would be able to register a customer. The details of the customer would be stored in the system in a customer database file. Once registered, the customer details can be used to perform transactions through the ‘Sales’ module.

- Once the customer has registered, the application should generate a unique customer id containing the Organization ID and an auto incremented number for uniquely identifying the customer.

- When retailer selling products to customers the data should be updated in database files so respective notifications can be shown for current outstanding balance and Product quantity.

- All the product details are stored in a Product Details database file. Product details will include Product Id, Product Name, Description, Product Quantity, and Price Per Unit.

- In the organization, product details are added by the administrator and the files will be circulated to all.

- The ‘Admin’ user should login before using the application. The ‘Admin’ user would input Login ID and Password.

- For adding a customer, ‘Admin’ user would input Customer Details. Customer number/id would be system generated and added along with the ‘Admin’ number of the user who created the customer. Before saving customer information into the file, the application must prompt if the ‘Admin’ user wants to save this customer information. Before saving, the application should check if the customer already exists.

- The application should display the current price of the products. As every products price varies as per the Whole sale rate, retailers rate and market rate ‘Admin’ should able to input the quantity of selected products the customer wants to buy.

- If the customer buys products successfully, then the Customer ID, Product Name, Amount, Price per Unit, Quantity of Product, Date of Transaction, and Type of Transaction would be recorded in the Transaction File.

- ‘Admin’ user will input the quantity of selected products that the customer wants to buy. ‘Admin’ user cannot sell products for the customer in the system if:

  - A notification will be provided which gives specification of current quantity of products.

  - Quantity of Products that the customer wants to buy is less than the Quantity of Products that is present in the Transaction file.

Proposed Solution

The proposed solution of the project is a desktop application.

The entry point of all customer requests will be a login screen which will be responsible validating the users. The application will allow the Admin to login into the system with provided Login ID and Password. After Login Admin can add Customer details and also product details. Admin will be responsible for managing the Products of the customer. Managing customer’s Products will include creation of customer’s account, buying Products, and selling Products.

Database tables will be used to store information related to sales, clients, and Products info.

Leading retailers are moving beyond standalone and disconnected retail software systems and touch points to focus on linking cross-channel customer interactions with foundational business systems to provide a single, unified view of customers and the business. The unification of business systems and the data that feeds them makes it possible for retailers to deliver a transparent, channel-agnostic, personalized customer experience.

Regal Soft is a retail management software system that brings together POS (Point Of Sale), General Ledger, Bills, and Invoice details, e-commerce merchandising and order management, financials, and warehouse management into a single, centrally-managed solution. This software solution is an end-to-end suite designed especially for multi-channel retailers that will enable enhanced customer service across channels while driving growth and increased revenue.

Project Specifications

The application is to be developed for Windows platform using Java SE 7, NetBeans 7.1, and SQL Server 2012.

The application should perform the following functions:

- Store information for managing the products sales and purchase in database files

- Sales
  - Register a new Admin user
  - Allow login for registered Admin
- The following options should be available to the Admin
  - Products management
  - Client/Customer information management
  - Generate Bill, Invoices
- Customer
  - Register a customer
  - Search for a particular customer information
  - View detailed customer information. The information should include:
    - Customer information:
      - Customer ÍD, Customer Name, Gender, Phone, Address, Email, Product Sales for this customer, customer’s balance (Outstanding)
    - Customer Product list
      - Account Number, Product name, Amount, Buying Price, Quantity, Market Rate Of Product, Retailers Rate Of Product, Wholesale Rate of product, Update customer information
- Products
  - Buy product
  - Sell product
  - View Products information

Hints for developing the Solution:

Create a desktop application that would have menu options that would do the following:

- Allow options to perform different tasks of allowing admin to login into the system, which allows to add new customer details also new product details. View Notifications and view bills and invoice details.

- Allow to view reports.

- All operations on the customer module will be done by the Admin person.

1.3 Team Building

A Project Leader leads this team. A Project Leader essentially decides which tasks are to be performed by each team member and how much time should be allotted to each Project Development Phase.

A team typically consists of:

- **Analyst**

  The Analyst studies the requirements of the system and defines the problem. The Analyst identifies the needs of the system and determines the inputs, outputs, and processes involved in transforming those inputs into outputs.

- **Designer**

  The Designer creates a blueprint of the system in terms of the database structure, screens, forms and reports.

- **Developer/Programmer**

  The Developer builds the user interface according to the specifications prepared by the Designer. Next, the Developer builds a prototype of the system. After receiving client approval on the prototype, the Developer adds the necessary code to make the prototype a full-fledged system.

- **Tester**

  The Tester tests the working of an application by first testing each module for its functionality. Test data is used to check if the module is able to process it without causing any errors. Test data may be live data extracted from existing records in the system or dummy data. The Tester then also verifies the integrated application’s functionality with test data.

- **Implementation Engineer**

  The implementation engineer ports the completed application to the Client's computers. The implementation engineer ensures that the installation process has been carried out accurately, and hands over the system to the Client.

- **Maintenance Engineer**

  The Maintenance engineer is responsible for taking care of maintaining the system that has been built. Maintenance includes extending troubleshooting support, and performing software upgrades in case of changes in the external system.

> Though each role is huge in itself, the same team member may perform more than one role in a team. For example, one person could assume the role of both Analyst and Designer.

The project team interacts with the client (customer for whom the project is being developed). For your project, the teacher takes on the role of the client. Hence, all the interactions that the project team performs with the client, you will perform with the Teacher.

> For the ***Retailers Software Application,*** the roles of the Implementation Engineer and Maintenance Engineer are beyond the scope of this project. Since each one of you has to learn the various roles, the members of the Project Group will perform all the roles of Analyst, Designer, Programmer, and Tester together. Each of you will not be assigned one role and be limited to performing that role only.

1.4 File Design

File structure of the different files is given in the following sections:

- **Customer Details**

  The Customer_Master file stores the customer details.

- **Admin**

  The Sales_Person file stores the details of all the sales people of the Organization belonging to a particular Region.

- **Product Details**

  The Customer_Products file stores the Product details for the customer to enable him/her to perform transactions.

- **Product Sales**

  The Product_Sales file stores the transaction details of the mutual funds of all the customers.

- **Customer Bill Details**

  This will store Bill details of the Customer.

1.5 Architecture of the Application

Figure 1.1 displays the application architecture.

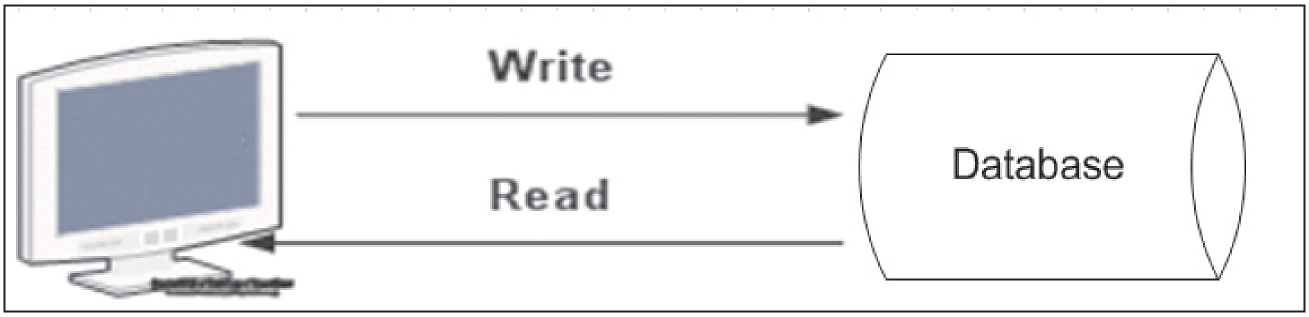


Figure 1.1: Application Architecture

Data Flow Diagram

Figure 1.2 displays the context level diagram of the application.

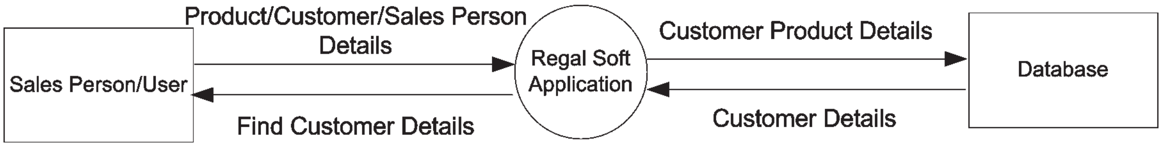


Figure 1.2: Context Level Diagram

Figure 1.3 displays the first level of Data Flow Diagram.

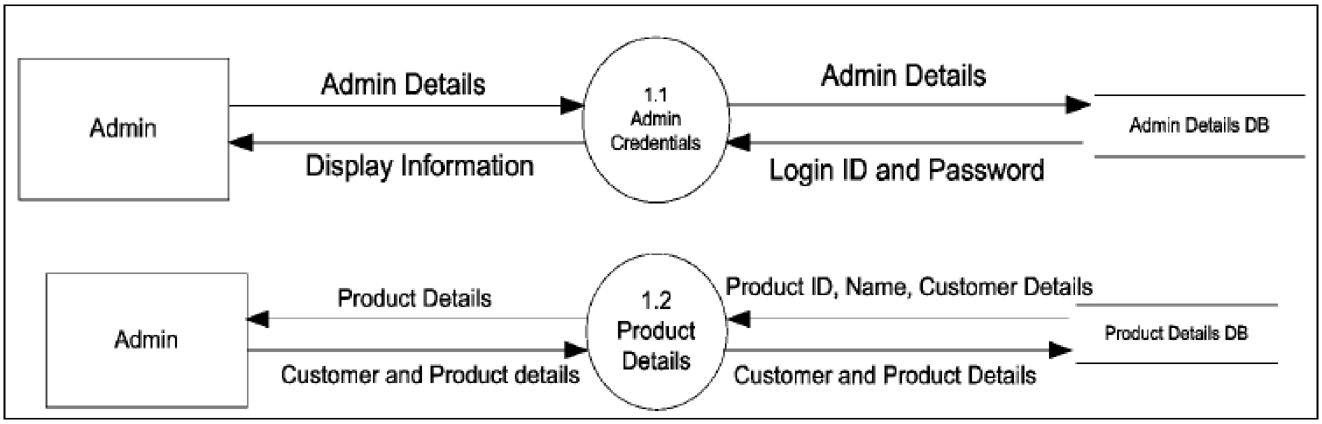


Figure 1.3: Admin and Products First Level DFD

Figure 1.4 displays the first level of data flow diagram of Product Transaction.

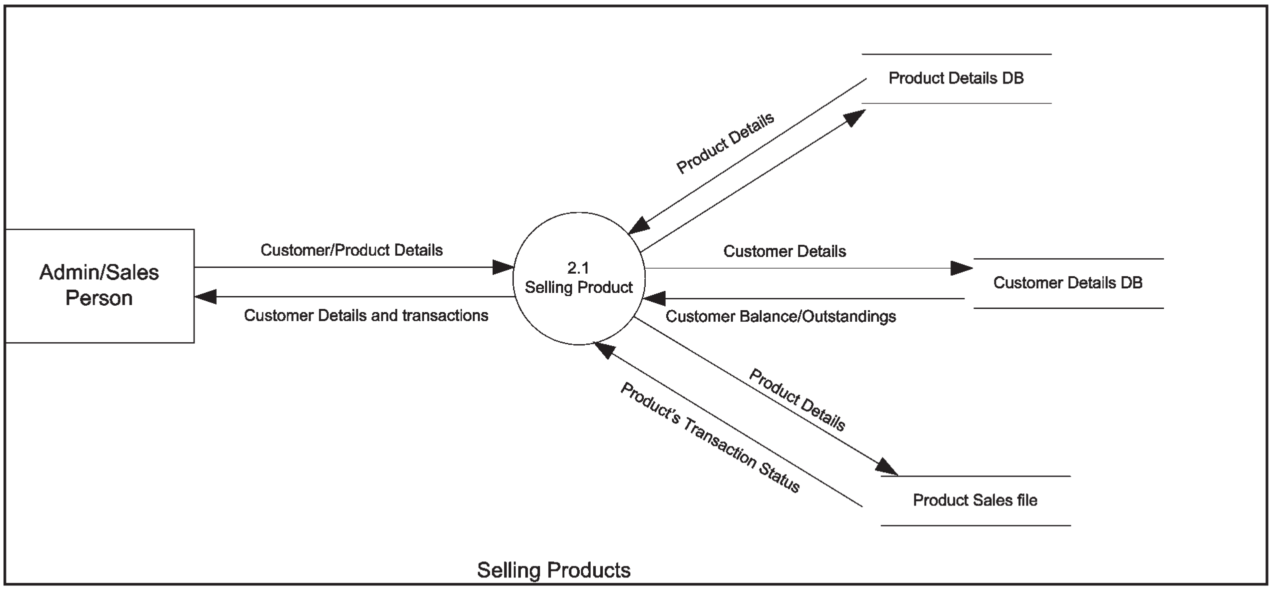


Figure 1.4: Selling Products

Use Case

Figure 1.5 shows the use case of the application.

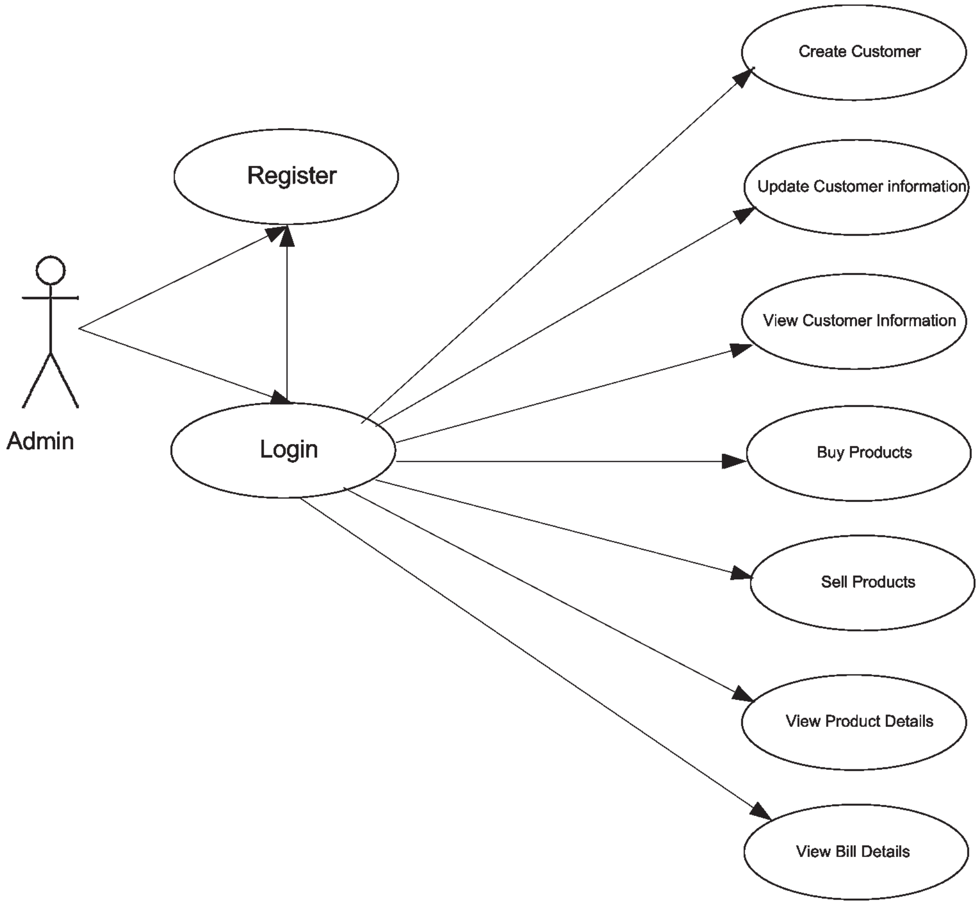


Figure 1.5: Use Case

Login

The application flow for login screen is shown in figure 1.6.

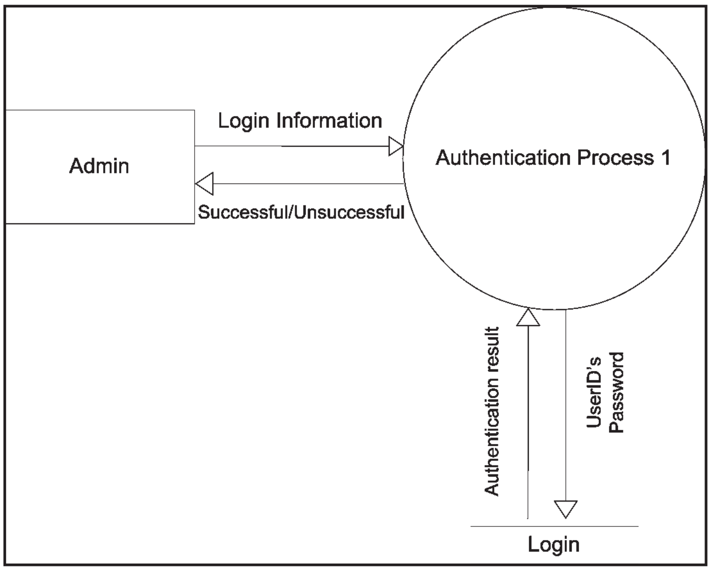


Figure 1.6: Login

Application Screen

Figure 1.7 shows the different activities associated with the application.

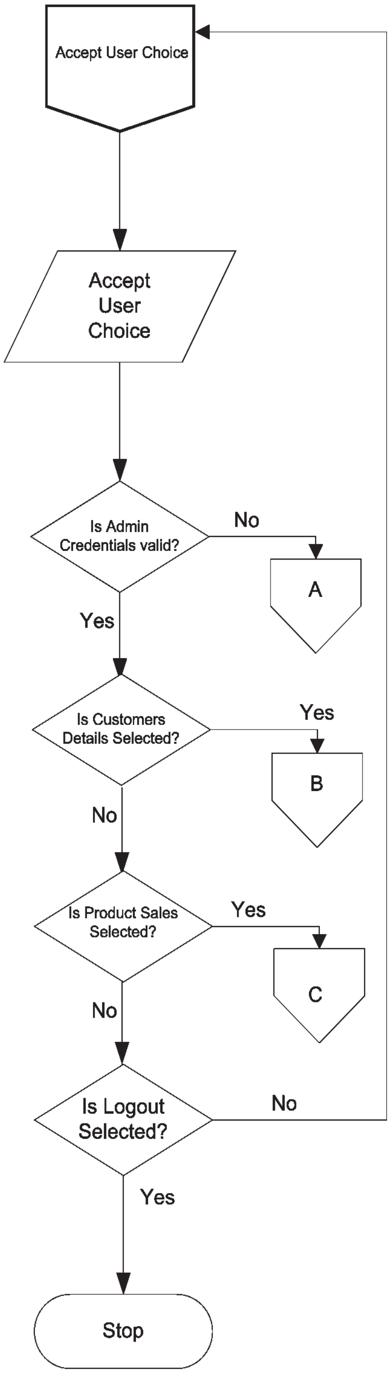

Accept User Choice
Accept User Choice
Is Admin Credentials valid?
No
A
Yes
Is Customers Details Selected?
Yes
B
No
Is Product Sales Selected?
Yes
C
No
Is Logout Selected?
No
Yes
Stop

Figure 1.7: Application Modules

Admin Process

Figure 1.8 shows the different admin activities associated with the application.

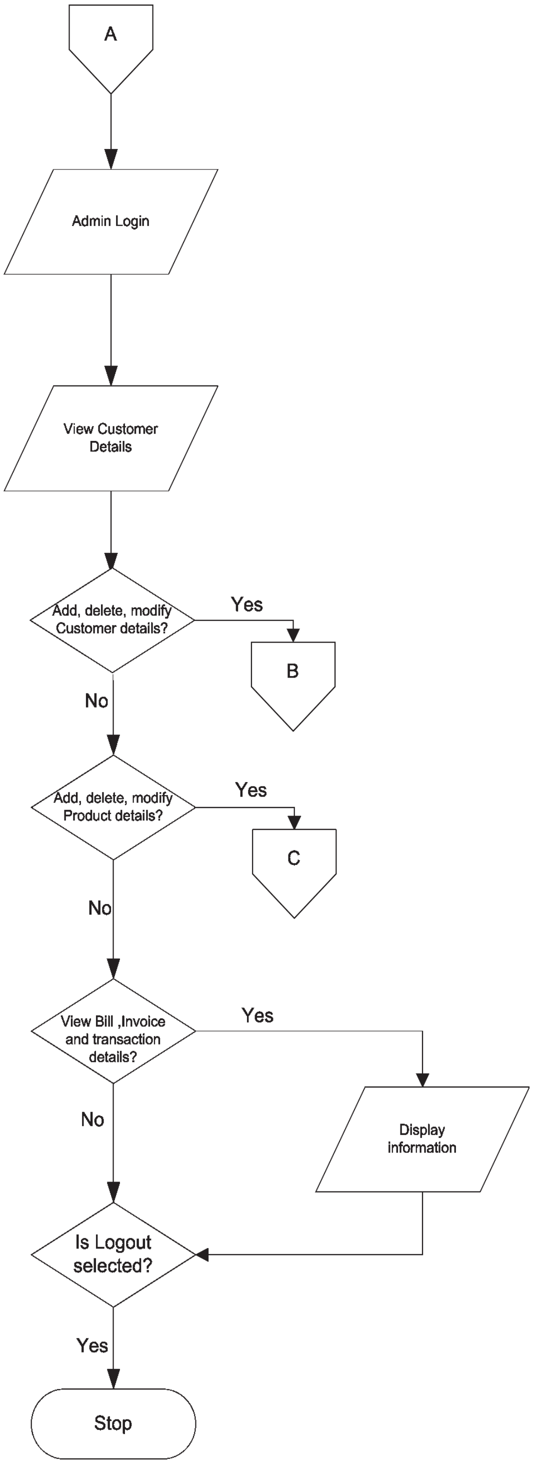

A
Admin Login
View Customer Details
Add, delete, modify Customer details?
Yes
B
No
Add, delete, modify Product details?
Yes
C
No
View Bill ,Invoice and transaction details?
Yes
Display information
No
Is Logout selected?
Yes
Stop

Figure 1.8: Admin Process

Selling Products

Figure 1.9 shows the application flow for selling products.

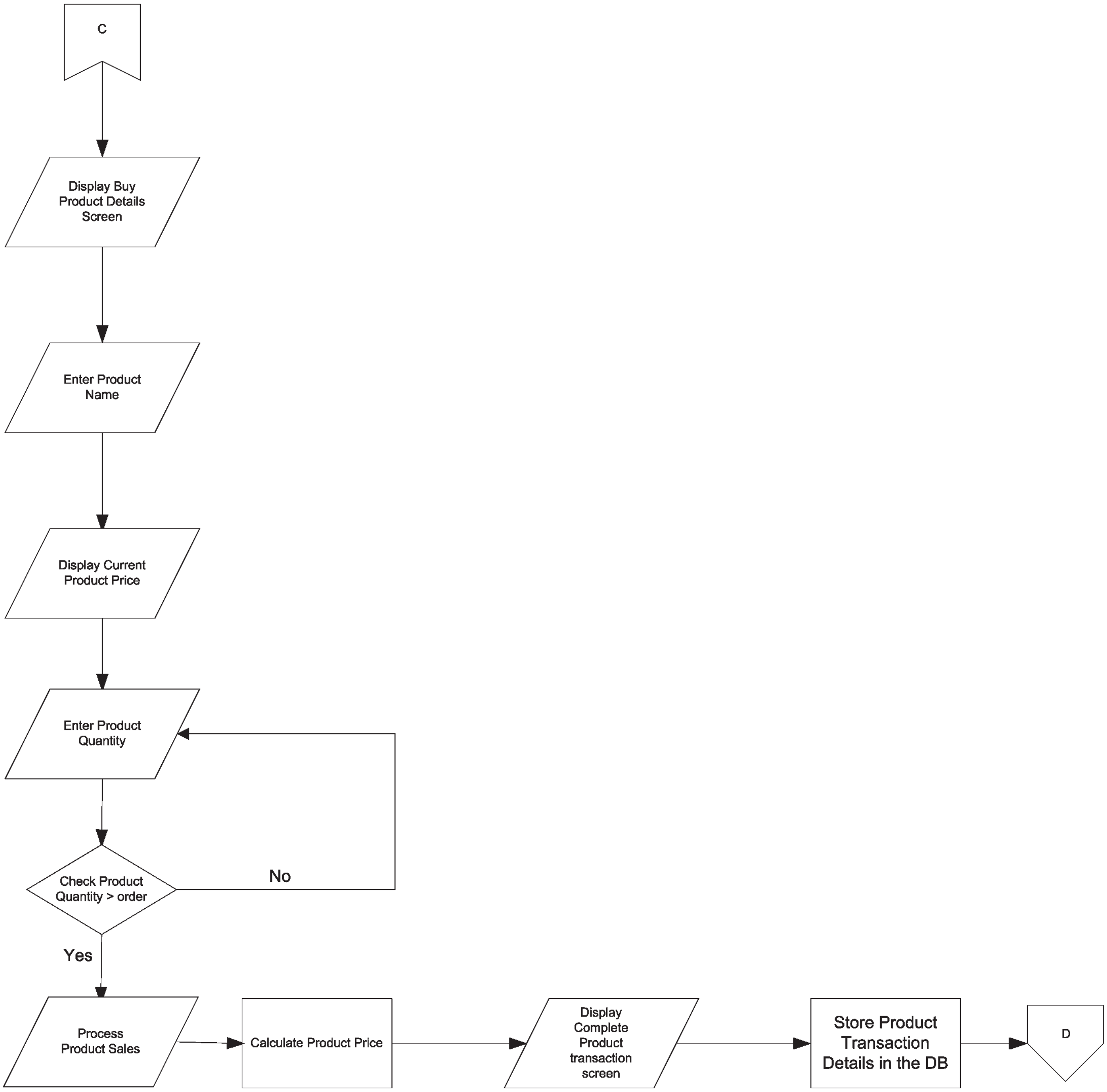

C
Display Buy Product Details Screen
Enter Product Name
Display Current Product Price
Enter Product Quantity
Check Product Quantity > order
No
Yes
Process Product Sales
Calculate Product Price
Display Complete Product transaction screen
Store Product Transaction Details in the DB
D

Figure 1.9: Selling Products

Figure 1.10 shows the application flow for addition of customers.

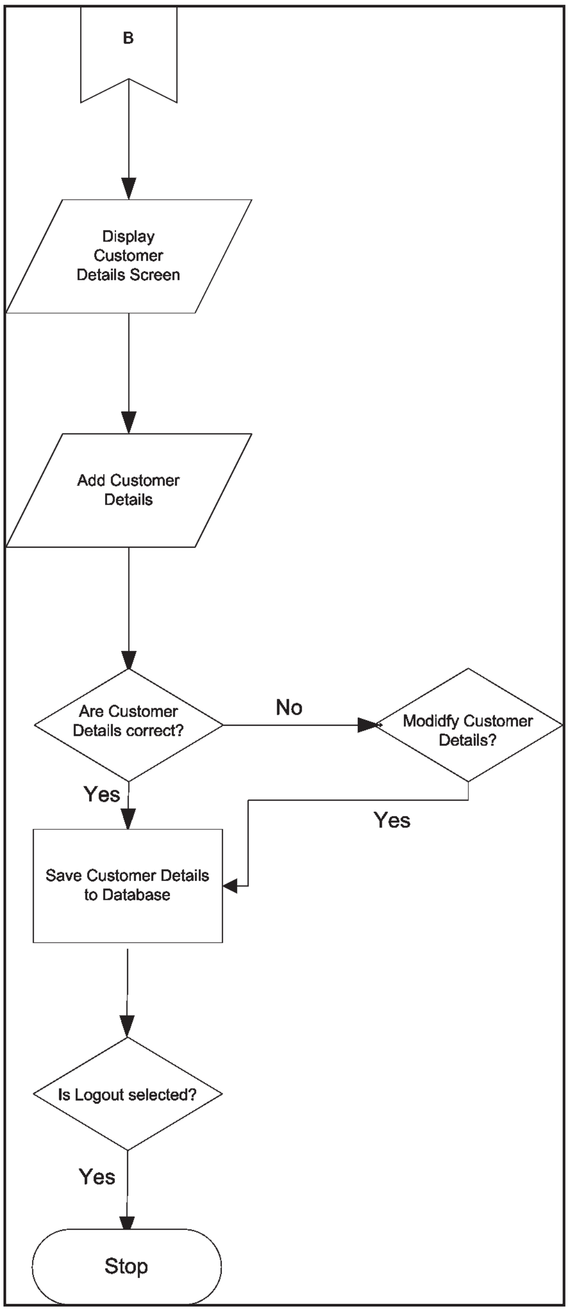

B
Display Customer Details Screen
Add Customer Details
Are Customer Details correct?
No
Modidfy Customer Details?
Yes
Yes
Save Customer Details to Database
Is Logout selected?
Yes
Stop

Figure 1.10: Add Customer

Database structure of the different Tables is given in the following sections:

- **Customer Details**

The Customer Details Table stores the customer details.

Field Name	Data Type	PK/FK	Description
Customer_ID	Varchar(50)	PK	Generates Customer ID
Customer_Name	Varchar(50)		Generates Customer Name for given ID
Customer_ Address	Varchar(100)		Generates Customer Address
Customer_mb_no	int		Generates Customer Mobile Number
Email_id	Varchar(50)		Generates email id for mentioned Customer ID
Credit_limit	int		Generates Credit limit for mentioned Customer ID
Total_balance	Int		Generates total balance of the Customer
Outstanding_Balance	Int		Generates Outstanding balance of the Customer

- **Admin**

The Admin Table stores the details of the admin of the organization belonging to a particular region.

Field Name	Data Type	PK/FK	Description
User ID	Varchar(50)	Pk	Generates User ID, (unique)
Username	Varchar(50)		Generates Username associated with User ID
Password	Varchar(50)		Generates Password associated with User ID

- **Product Master**

The Products master stores product details.

Field Name	Data_type	PK/FK	Description
Product_ID	Varchar(50)	PK	Generates Product ID
Product_Category	Varchar(50)		Generates Product Category
Product_Name	Varchar(50)		Generates Product Name
Product_Quantity	Int		Generates Product Quantity
Product_Price	Int		Generates Product price
Discount	Float		Generates discount for given total amount

- **Product Sales**

The Product Sales Table stores the transaction details of the product sales of all the customers.

Field Name	Data_type	PK/FK	Description
Customer_ID	Varchar(50)	FK	Generates Customer ID
Product_ID	Varchar(50)	PK	Generates Product ID
Product_Quantity	Int		Generates Product Quantity
Product_Price	Int		Generates Product price
Total Price	Int		Generates Total price for the given customer
Date_Of_Transaction	Date		Generates Date and time for mentioned order
Payment_Mode	Varchar(50)		Generates type of payment made by the customer
Payment_made	int		Generates Payment value of the given Customer ID
Outstanding_balance	Int		Generates outstanding balance of the provided customer

1.6 Phases in a Project's Life Cycle

Every software development activity can be distinguished into clear phases. Each phase has precise starting and ending points, with clearly identifiable deliverables to the next phase. Each phase may have certain documents, which help to keep track of the various activities, processes, procedures, inputs, and outputs associated with that phase of the Project.

The project development life cycle consist of the following phases:

- **Phase 1: Definition of the Problem**
- **Phase 2: Requirement analysis**
- **Phase 3: Design**
- **Phase 4: Development**
- **Phase 5: Evaluation/Testing**
- **Phase 6: Implementation**
- **Phase 7: Maintenance**

Each of these phases is discussed.

1.6.1 Definition of the Problem

Correct and accurate deduction of the client requirements and expectations from the system is the key to the development of the system. This phase assumes more significance as it has cost implications attached and these will affect the whole project. When the need is wrongly identified, the whole system will be developed on a wrong premise.

Therefore, this phase involves defining the problem and fixing up its boundaries. The needs and problems faced by the client are recorded in this phase. At the end of this phase, the team is clear with the project objectives and their work purview. Inputs to this phase are always unstructured. These inputs are gathered from interactions with the client.

The activities involved in this phase are:

- Meeting the Client
- Understanding the Customer's needs

- Identifying the probable solution

- Defining the scope of the project

The ***Problem definition*** in the Regal Soft application would be: With the rise in retail business, every Retailer has to maintain a single record of every product whether it's buying or selling. Many organizations are deploying industry standard quality management methodologies based on ISO 9001 and /or any other quality standards relevant to their type of industry. There is a need to offer software solutions to offset cost by substituting paper based expenses with Increased control, efficiency, productivity, and live documentation.

This Software will enable people with all the guidance required to select from variety of products i.e. Sales, Purchase, Invoice, Ledger, Customer Details and Month-wise reports, Stock reports, and Notifications.

Also, the application should be capable of storage, retrieval, and manipulation of data in appropriate database files.

This Problem Definition has been identified and recorded in the ***Problem Definition Document*** in "**SWD/Form No.1**".

Once a client's problem has been identified, the next step would be to generate the **Customer Requirement Specification** as a part of the Requirement Analysis phase.

Now, carefully study and understand the Case Study specifications in the ***Regal Soft Application***.

1.6.2 Requirement Analysis Phase

During this phase, the Analyst identifies the processes of the current system, and the inputs and outputs for those processes. The processes, inputs and outputs are recorded in a document called the Customer Requirement Specification. Actually, the **Customer Requirement Specification** is not a single document; it is a docket of many documents. It consists of:

- **List of inputs to the system**

  These refer to the inputs required for the system to work on and thus produce the desired output.

  For example, in the ***Regal Soft Application*** case, some inputs would be Admin details, Customer details, Product details, and so on.

- **List of outputs expected from the system**

  These refer to the various reports that the system would produce.

For example, in the ***Regal Soft Application*** case, some of the outputs would include a search results for customer details, Product details, and so on.

- **Overview of processes involved in the system**

Once the inputs and outputs are listed down, the list of processes that convert the inputs into desired outputs is prepared.

For example, in the ***Regal Soft Application*** case, this list would include the following:

- Admin Login
- Customer registration
- Buy/Sell Products

- **Hardware and software required for implementing the project**

Here, the team gives a list of software and hardware required to implement the system. The Customer uses this information to get an approximation of the required setup and training requirements.

For example, in the ***Regal Soft Application*** case, the software required could be Java SE 7.0, SQL Server 2012, and NetBeans 7.1.

For the ***Regal Soft Application***, you must record the same in the document **"SWD/Form No.2A".**

- **Customer's acceptance criteria for the project**

The software product developed must meet the customer's needs and expectations as specified in the problem definition. Apart from this, it should also fulfill the performance, speed, and reliability parameters specified by the Customer.

For example, in the ***Regal Soft Application*** case, the acceptance criteria could be the following:

- Create an 'Admin' user
- Login in to the system with the created Admin login ID and Password
- Create a customer
- Buy Products

- Sell Products

For the ***Regal Soft Trade Application***, you must record the **Customer Acceptance Criteria** in the document **"SWD/Form No.2B"**.

This Customer Requirement Specification Document needs customer approval, which is termed as ***sign-off***.

Once the customer has given a sign off on the ***Customer Requirement Specification*** Document, the ***Project Leader*** creates a "**Project Plan**". A Project Plan is a brief of all the phases, deliverables, milestones, and implementation issues related to a project.

For the ***Regal Soft Application***, you must create a "**Project Plan**" and submit to the teacher for review. You can fill your **Project Plan** details in the document "**SWD/Form No. 3**" given in the ***Documentation Section*** of this Project Guide. Your Teacher will prepare the "**Project Plan**" for the ***Regal Soft Trade Application*** and demonstrate it to you.

1.6.3 Design Phase

Design Phase involves the preparation of the blueprint of the proposed system, which involves the following:

1. Designing the GUI Standards
2. Designing the Interface for the application
3. Designing the Database
4. Designing the Process Modules
5. Designing the Coding Standards
6. Building the Prototype
7. Assigning and Monitoring Tasks

➢ **Designing GUI Standards**

When an application is designed, it must follow standards with respect to flow, appearance, and look of an application. Standards are used to bring about consistency throughout the application.

GUI Standards are related to the appearance of an application. It is mandatory for the Project Leader and team members to visualize the entire look of the application before it is actually developed. This visualization has to be defined in terms of GUI Standards so that each screen being developed maintains consistency in look and flow. The color, font style, size of titles and labels, appearance of header and footer, theme, position and size of controls on various screens are defined here.

For example, in the case of ***Regal Soft Application***, multiple screens are created that are linked to each other. Each of these screens should have a consistent look with respect to appearance, and theme used. The font styles, color of the labels, design and appearance of the command buttons, the appearance of header and footer, and the design and size of controls like check boxes or text boxes should be consistent through all the forms.

For the ***Regal Soft Application***, set your GUI Standards using the document "**SWD/Form No. 4**" given in the ***Documentation Section*** of this Project Guide.

- **Designing the Interface**

Here, the layouts of the screens are designed in line with the **GUI Standards** set. Either these screens can be input forms, which accept user inputs, or Reports that display information to the user. In this phase, the content and appearance of the input forms and Reports are decided. The number of forms and the purposes of each form are decided here.

The navigation details of the entire application are also specified during this phase if the application has a browser interface.

The Interface Design is recorded in the ***Interface Design Document***.

For the ***Regal Soft Trade Application***, you must record your ***Interface Design*** details in the document "**SWD/Form No.5**" given in the ***Documentation Section*** of this Project Guide.

- **Designing the Database**

In this phase, the files have to be created according to the specification given in the project guide.

For the ***Regal Soft Application***, you must record your ***Table Design*** details in the document "**SWD/Form No.6**" given in the ***Documentation Section*** of this Project Guide.

Your teacher will demonstrate the file design process to you using the ***Regal Soft Application*** case study.

- **Designing the Process Modules**

Process design involves translating the process definitions arrived at in the analysis phase into code modules. This module design is then expanded into program specifications. The types of validations needed to verify the functionality of each process in the project, are also specified in this phase.

For example, in the case of ***Regal Soft Application*** the modules could be for:

- Sales
- Customer
- Products

For the ***Regal Soft Application***, you must record your **Process Design** details in the document "**SWD/Form No.7**" given in the **Documentation Section** of this Project Guide.

➢ Designing Coding Standards

The process modules designed will be considered good only if they are standardized. Standardization involves setting up of naming conventions of program entities to name a few. Standardization helps to bring about better readability and easy maintenance of the code. Standard naming conventions make it easy to refer to program entities like forms or modules. When standard naming conventions are used, the names of variables, forms and modules will denote their purpose even to a person other than the developer.

The Teacher will show you how to set the Coding Standards using the ***Regal Soft Application*** case.

For the ***Regal Soft Application***, you must define and state your ***Coding Standards*** in the document "**SWD/Form No.8**" given in the ***Documentation Section*** of this Project Guide.

➢ Building the Prototype

Next, a Prototype of the application is created and shown to the client for approval. The Designer delegates the development of the prototype to the developer. The Prototype is a model of what the application would look like. The Prototype's screens allow the Client to view the User Interface of the application. In case a browser interface is used, the Client gets an idea of the navigation sequence of the application. The Prototype also helps the Client to understand the functionalities that will be achieved in the completed application. Once a Prototype is created, it needs Client approval. The Client has to give approval for:

- The navigation sequence of the application
- The look and appearance of the application
- The functions that will be performed by the application when fully developed.

There is no document associated with the Prototype since the client is shown the prototype on the computer for approval.

Once all the ***Design Specification Documents*** have been prepared, they need Client approval, which is called ***sign-off***.

- *After the **sign-off,** the system is taken up for development. Now, if the client requests massive new additions or changes to be made, then a document known as **IMPACT ANALYSIS** is generated.*

- ***Impact Analysis** is a document which informs the client about the additional cost to be incurred as a result of the changes suggested by him and also the time delays involved due to the requested changes.*

After a Client gives a sign off on Impact Analysis, the requested changes are made.

During the Design Phase, the Project Leader also carries out the task of ***Assigning and Monitoring Tasks.***

- **Assigning and Monitoring Tasks**

  When the project development starts, the Project Leader needs to budget the number of person-hours required to complete the project. In order to do this, the Project Leader creates a “**Task Sheet**“.

  A task sheet is used for recording the following:

  - The number of man-hours required to complete the project
  - The planned start date of the project
  - The planned end date of the project
  - The number of members required in the project team
  - The Module name and specification allocated to each project member
  - The time required to complete each module
  - The progress status.

With a Task sheet, the Project leader will be able to monitor and track the various phases of project development. It helps the Project leader to anticipate delays, and therefore become proactive and take preventive action.

For the ***Regal Soft Application***, you are expected to fill the ***Task sheet*** as shown in the document "**SWD/Form No.9**" given in the ***Documentation Section*** of this Project Guide. The teacher will demonstrate how to fill the **Task sheet** using the ***Regal Soft Application*** case.

1.6.4 Development Phase

This phase involves the following:

- Giving the finishing touches to the User Interface built during the prototype
- Building the Database with the RDBMS tool
- Making the screens functional by adding code to them.

For the ***Regal Soft Application***, you must now implement the User Interface using Java. You are required then to design the file structure.

1.6.5 Evaluation/Testing

This is the most crucial phase where each unit is tested for its functionality. Test data is used to check if the module is able to process it without causing any errors. Test data may be live data extracted from existing records in the system, or dummy data. Then, the individual tested modules are integrated and tested as a whole through its various paths.

During this phase, the Project leader reviews the developed system against each of the Customer Requirement Specifications and thus ensures that the developed system is able to resolve the Problem Definition completely.

After finishing the project, give it to your peers for testing the entire project. The project should be verified for the data entered by the customer in the various forms. The findings can be records in the document "**SWD/Form No. 10**".

1.6.6 Implementation Phase

In this phase, the developed system is ported to the client's computers. The implementation engineer ensures that the installation procedure has been performed accurately. At the end of this phase, a final ***sign off*** is taken from the client.

This phase is beyond the scope of the ***Regal Soft Application*** project.

1.6.7 Maintainance Phase

In this phase, troubleshooting support is given to the Client. Depending upon external changes in the system, any software upgrades required by the application are performed.

This phase, again, is beyond the scope of the ***Regal Soft Application*** project.

1.6.8 Project Tracking and Monitoring Activities

During the Project Development Life Cycle, the Project leader and other team members follow certain monitoring procedures and practices which help to streamline projects and achieve the project objectives. These activities happen in parallel to the process of project development. The Project's successful completion depends on these activities. These are known as project reviews.

A Review is a procedure used to check the progress of a project. It helps to understand and resolve constraints and to ensure that the project is progressing as per the Project Plan and will meet all Project Specifications, Guidelines and deadlines.

Reviews can be:

- Internal Reviews- reviews conducted within the team members
- External Reviews – reviews conducted with the Client.
- On a weekly or a monthly basis and can be conducted either through emails or in person.

Review Minutes are recorded in the ***Project Review and Monitoring Reports*** document.

For the ***Regal Soft Application project***, your teacher will conduct a review to understand the project status and record the findings of the ***Project Review*** in the document "**SWD/Form No.11**", given in the ***Documentation Section*** of this Project Guide.

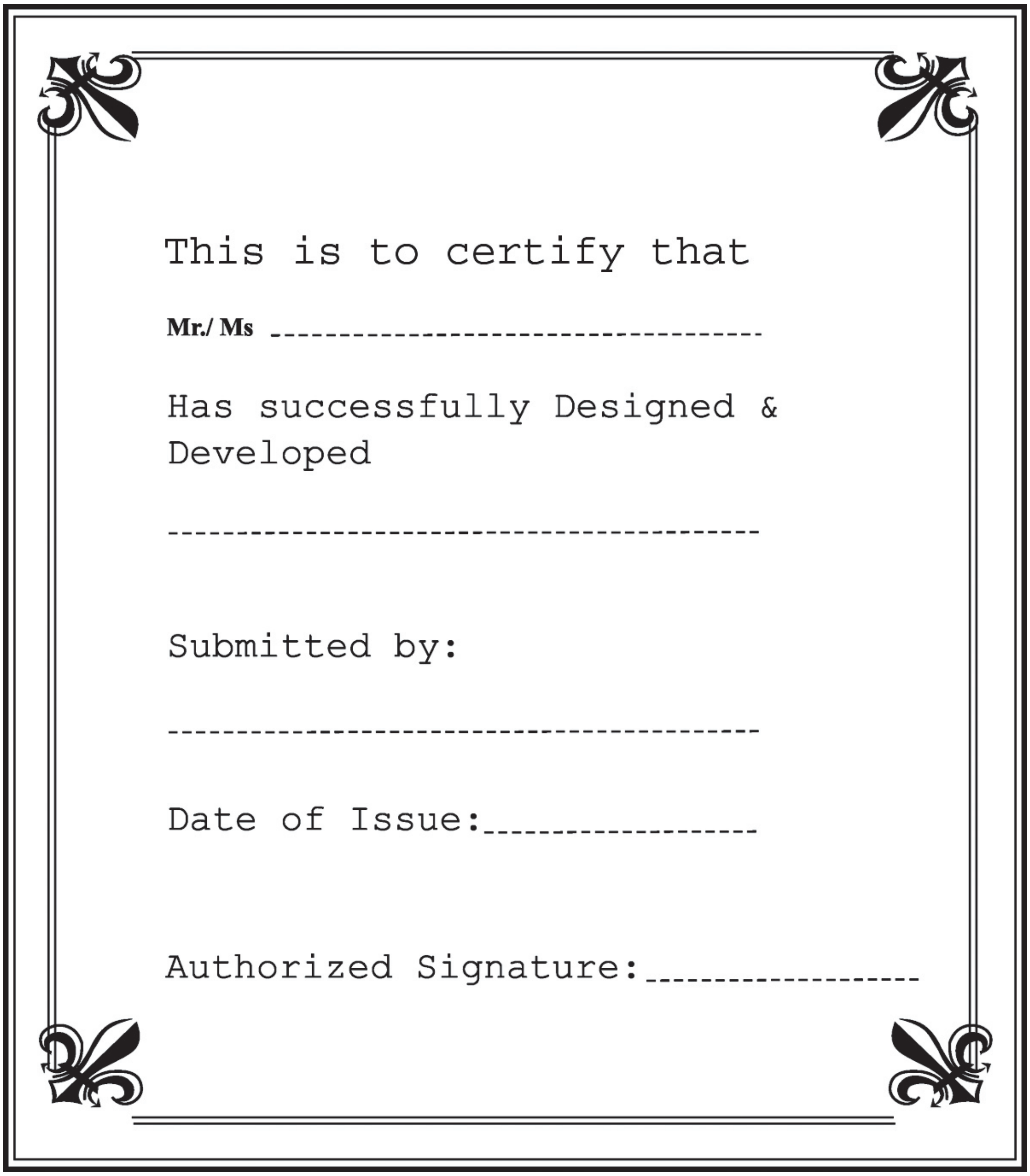
Finally, your project would be considered complete when you give a final demonstration of the project to the teacher. During the demonstration, the teacher will fill and sign the "**Final Check List**" (document –"**SWD/Form No.12**") given in the Documentation Section. After this, furnish a spiral bound book consisting of the following documents:

- **Certificate of Completion**
- **Table of Contents**
- **Definition of the Problem**

- **Customer Requirement Specification Document**
- **Project Plan**
- **GUI Standards Document**
- **Interface Design Document**
- **Coding Standards Document**
- **Process Description Document**
- **Task Sheet**
- **Integration Testing Report**
- **Review Report**
- **Final checklist**

1.7 Documentation Section

- Certificate of Completion



This is to certify that

Mr./ Ms ------------------------------------

Has successfully Designed & Developed

Submitted by:

Date of Issue:-------------------

Authorized Signature:-------------------

- **Various forms required in the project**

Design Plan:	Document Name: Problem Definition Document	SWD/Form No. 1
Effective Date:	Version: 1	Page No.

Problem definition of Regal Soft Application –

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: CRS	SWD/Form No. 2A
Effective Date:	Version: 1	Page No.

Client/Project Undertaken: ________________________________

1. List of inputs to the system

____

____

____

____

2. List of outputs expected from the system

____

____

____

____

3. Overview of processes involved in the system

4. Hardware and software required for implementing the project

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: CRS/ Customer Acceptance Criteria	SWD/Form No. 2B
Effective Date:	Version: 1	Page No.

S.No	Customer's Acceptance Criteria
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: Project Plan	SWD/Form No. 3
Effective Date:	Version: 1	Page No.

Project Plan

1. Project Details

- **Name of the Client:**

  ____

- **Date of Project Plan:** ______________________________________

- **Project Vision/Objectives:**

(Define the project vision/objectives as stated by the client)

____

____

____

- **Scope:**

(Mention the scope of the project giving the locations that will be covered, processes, range of services, and so on.)

____

____

____

____

- **Our understanding of the client organization:**

(Give the range of services, functions, overview of processes, and so on.)

____

____

____

- **Project Organization with Responsibilities and Authorities:**

(Give the name of Project team members their roles and responsibilities.)

____

____

____

____

2. Project Initiation/Requirement Documents

(Information required from the Client as inputs regarding his system; could be the information about his services, processes.)

____

____

____

____

3. Deliverables

(The documents to be handed over to the client – like CRS, Design Document, Installation Manual, User Manual, Maintenance Manual, and Code documents.)

____

____

4. Project Dependencies

Any event or task outside the scope of the project, which will affect the success of the project.

5. Major Project Milestones

(Generating CRS, Building a Prototype.)

6. Quality Plan

- **Review Activities** (Review meeting participants, frequency)

- **Testing Activities** (Final Test)

____

____

____

- **Backup and recovery strategies** (in case of disk crash, network failures)

____

____

____

____

____

____

	Prepared By (Student)	**Approved By (Teacher)**
Date		

Design Plan:	**Document Name: GUI Standards Document**	**SWD/Form No. 4**
Effective Date:	**Version: 1**	**Page No.**

Document Design

Property	Value
Document theme and color scheme	
Form-Background color	
Title-Font Size	
Title-Font Color	
Title-Font Style	
Title-Alignment	
Background color of Controls on the form	
Foreground color of Controls on the form	
Control Caption- Font size	
Control Caption- Font Color	
Control Caption- Font Style	
Control caption and controls-Alignment	
Command button-Alignment	

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: Interface Design Document	SWD/Form No. 5
Effective Date:	Version: 1	Page No.

1. List of forms to be created

Document Name	Description	Controls on the Document

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: Table Design Document	SWD/Form No. 6
Effective Date:	Version: 1	Page No.

Name of the Table	Table Description	Number of Fields	Primary Key	Related Tables	Foreign Key
				1. 2. 3.	

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: Process Design Document	SWD/Form No. 7
Effective Date:	Version: 1	Page No.

1. **Process Name**

____

2. **Process Description and Details**

____

____

____

____

____

____

____

____

____

____

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: Coding Standards Document	SWD/Form No. 8
Effective Date:	Version: 1	Page No.

1. **Programming Standards**

____

____

____

2. Standards for code writing style

____

____

____

3. Standards for declaring Variables

____

____

____

4. Standards for Function Declarations

____

____

____

5. Other Standards

	Prepared By (Student)	Approved By (Teacher)
Date		

Design Plan:	Document Name: Task Sheet	SWD/Form No. 9
Effective Date:	Version: 1	Page No.

Project Ref. No.:	Project Title:	Activity Plan Prepared By:		Date of Preparation of Activity Plan:		
Task Sub division	Description	Planned Start Date	Actual Start Date	Actual Days	Team Member Names	Status
1. 2.						

	Prepared By (Student)	Approved By (Teacher)
Date		

Test Document:	Document Name: Testing Document	SWD/Form No. 10
Effective Date:	Version: 1	Page No.

S.No.	Features Tested	Remarks

	Prepared By (Student)	Approved By (Teacher)
Date		

Review Report:	Document Name: Project Review	SWD/Form No. 11
Effective Date:	Version: 1	Page No.

Date:	Project Plan Activity/ Milestone	Work Specification	Status of the Activity	Remarks	Responsibility

	Prepared By (Student)	Approved By (Teacher)
Date		

Final Check List

(This document has to be filled by the teacher only. The teacher will do a complete functionality testing of the application and add his/her findings and suggestions to this form.)

Test Document:	Document Name: Final Check List	SWD/Form No. 12
Effective Date:	Version: 1	Page No.

S.No.	Aspect tested	Suggestion/ Remarks
1.	Have all the modules been properly integrated and are they completely functional?	
2.	Does each unit meet its objective and purpose? Are all the validations happening as specified in Process Design?	
3.	Have all Design and Coding standards been followed and implemented?	
4.	Is the GUI design consistent all over?	
5.	Are the codes working as per the specification?	
6.	Does the application's functionality resolve the client problem, and satisfy his needs completely?	
7.	Have the hardware and software been correctly chosen?	
8.	Additional features and utilities that give value addition to the entire project.	

	Prepared By (Student)	Approved By (Teacher)
Date		

1.8 Naming Convention for Classes, Objects, and Fields

Identifier	Naming Convention
Package	Package name should use only lower case letters and digits and no underscores. **Example:** `package chat;`
Class	Names should start with uppercase letter, and capitalize the first letter of subsequent words as well any letters that are part of acronym. All other characters in the name have to be lower case. Avoid using underscores for separating words. **Example:** `UserName, Server, ChatServer`
Interface	Same as Class

Method	Names should start with a lowercase letter. Capitalize the first letter of subsequent words as well as any letters that are part of an acronym. Method names should be verbs or verb phrases.
	Example:
	`public void sendMessage()`
	`{...........}`

Variables	Same as Method. However variable names should be nouns or noun phrases.
	Example:
	`private int count;`
	`char recordDelimiter;`

1.9 Coding Standards for GUI Controls

Table 1 lists the prefixes to be used for various GUI controls.

Prefix	Control Description	Prefix	Control Description
frm	JFrame	txf	JFormattedTextField
apl	JApplet	txp	JTextPane
pnl	JPanel	jop	JOptionPane
lbl	JLabel	dlg	JDialog
btn	JButton	mbr	JMenuBar
chk	JCheckbox	mnu	JMenu
rad	JRadioButton	mni	JMenuItem
txt	JTextField	mnc	JCheckboxMenuItem
txa	JTextArea	mnp	JPopupMenu
pwd	JPasswordField	fcr	JFileChooser
scp	JScrollPane	tbr	JToolbar
sdr	JSlider	pgb	JProgressBar
tmr	Timer		

Table 1: Prefixes for GUI controls

Table 2 lists the various prefixes to be used for data access objects.

Prefix	Data Access Objects
cn	Connection
st	Statement
pst	PreparedStatement
rs	ResultSet
ros	RowSet
crs	CachedRowSet
jrs	JDBCRowSet
dmd	DatabaseMetaData
rmd	ResultSetMetaData
pmd	ParameterizedMetaData

Table 2: Prefixes for Data Access objects

1.10 Tips for Normalization

Dr. Codd originally defined three levels of normalization. These three levels were called f*irst normal form, second normal form* and *third normal form* respectively. Normalization is usually discussed in terms of forms. Normal forms are table structures with minimum redundancy. The normal forms that have been identified are:

- First Normal Form (1st NF)
- Second Normal Form (2nd NF)
- Third Normal Form (3rd NF)
- Boyce-Codd Normal Form

The first three forms were defined by Dr.Codd. Later Dr. Codd and Boyce introduced one more normal form called the Boyce-Codd normal form. The theory of normalization is based on the concept of *functional dependency*. Figure 1.11 illustrates the levels of normalization.

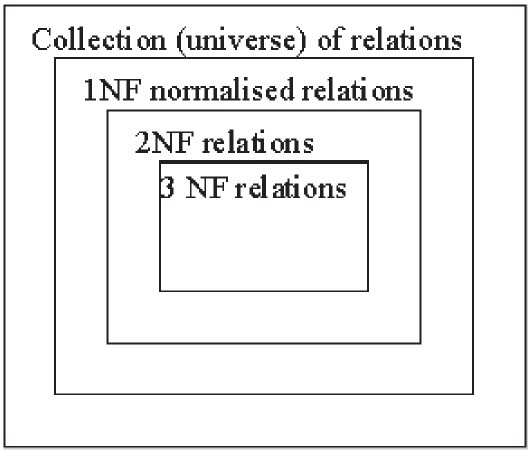


Figure 1.11: Levels of Normalization

In order to understand more about normal forms, we must first understand what is meant by functional dependency. This will be covered in the following section.

Functional Dependency

This concept is best explained with the help of an example. In the library example, it was seen that the telephone number and address were obtained by looking up the name of the member. In this case, the telephone number is functionally dependent on the member name.

Based on this we can now say that functional dependency is a relationship between attributes (fields).

Because of functional dependency, given the value of one attribute, it is possible to look up the value of another attribute. In general, attribute Y is functionally dependent on attribute X, if the value of X determines the value of Y. To put it across simply, if we know the value of X, we can determine the value of Y.

First Normal Form

This form is represented as 1NF. To put data into 1NF all the repeating information has to be removed. Consider a library issue table. Look at how this data is recorded in the librarian's register.

Mem Name	Contact	Book Code	Cat Code	Issue Date	Due Date
JOE	414 2319	B0020	SCIENCE	18-3-98	18-4-98
SAM	890 0909	B0189	ASTRO	18-3-98	18-4-98
		B0090		18-3-98	18-4-98
		B0091		18-3-98	18-4-98
PENNY	421 9498	B0656	ROMANCE	19-3-98	19-4-98
JENNY	567 0967	B0198	ADVENTURE	20-3-98	20-4-98
Mem Name	**Contact**	**Book Code**	**Cat Code**	**Issue Date**	**Due Date**
		B0212		20-3-98	20-4-98
		B0400		20-3-98	20-4-98
		B0555		20-3-98	20-4-98
PETER	328 9565	B0001	CLASSIC	21-3-98	21-4-98

Table 3: Un-normalized data

The data shown in Table 3 is un-normalized data. Since a relational model does not permit this, this table has to be normalized first. This table stores details about the books issued to its members. Each member is assigned a code and has a contact telephone number. Each book too has been assigned a code, which also has a category code to indicate which category a book belongs to: fiction, non-fiction, political, science, etc. Most category codes are self-explanatory.

Mem-Cd	Mem-Name	Contact	Book Code	Cat Code	Issue Date	Due Date
M001	JOE	414 2319	B0020	SCIENCE	18-3-98	18-4-98
M008	SAM	890 0909	B0189	ASTRO	18-3-98	18-4-98
M008	SAM	890 0909	B0090	ASTRO	18-3-98	18-4-98
M008	SAM	890 0909	B0091	ASTRO	18-3-98	18-4-98
M067	PENNY	421 9498	B0656	ROMANCE	19-3-98	19-4-98
M0123	JENNY	567 0967	B0189	ADVENTURE	20-3-98	20-4-98
M0123	JENNY	567 0967	B0212	ADVENTURE	20-3-98	20-4-98
M0123	JENNY	567 0967	B0400	ADVENTURE	20-3-98	20-4-98
M0123	JENNY	567 0967	B0555	ADVENTURE	20-3-98	20-4-98

Mem-Cd	Mem-Name	Contact	Book Code	Cat Code	Issue Date	Due Date
M0880	PETER	328 9565	B0001	CLASSIC	21-3-98	21-4-98

Table 4: First Normal Form

Let us understand what is stored in these tables and why, before we proceed to understand normal forms. The code, name and contact telephone number of the member is stored. Next, book details are stored. In the end, issue dates and Due dates are stored. It is obvious here that a lot of information is being repeated. The table 4 is said to be in its First normal form. For converting a table to its First normal form, follow the steps given as follows:

1. The un-normalized data in the first table has been listed together as shown in Table 3.

2. A key that will uniquely identify each record should be assigned to the table. This key has to be unique because it should be capable of extracting information from any row. In our case, a unique way of identifying a row would be a combination of member code + book code. This will be the primary key.

Several problems can be identified with this table with respect to inserting, updating and deleting data. Let us understand what these problems are.

Inserting data: A new book in the library cannot be entered until it is issued to some one.

Updating data: Any change in one of the records will lead to changing more than one record.

Deleting records: When a member returns a book, that particular record will be deleted. This will also result in loss of information about the book and its category.

According to C. J. Date, the DBM authority, a table is said to be in its first normal form if and only if all the underlying domains contain values that are not decomposable any further.

Since there is redundant data in this table, let us convert this table to its Second normal form.

Second Normal Form

To bring a table to its Second normal form, data that is dependent on part of the key should be separated.

The table is further decomposed without loss of information as follows:

Member Code	Member Name	Contact
M001	JOE	414 2319
M008	SAM	890 0909
M067	PENNY	421 9498
M0123	JENNY	567 0967
M0880	PETER	328 9565

Table 5: Member table

The primary key in Table 5 is the member code.

Member Code	Book Code	Category-Code	Issue Date	Due Date
M001	B0020	SCIENCE	18-3-98	18-4-98
M008	B0189	ASTRO	18-3-98	18-4-98
M008	B0090	ASTRO	18-3-98	18-4-98
M008	B0091	ASTRO	18-3-98	18-4-98
M067	B0656	ROMANCE	19-3-98	19-4-98
Member Code	**Book Code**	**Category-Code**	**Issue Date**	**Due Date**
M0123	B0198	ADVENTURE	20-3-98	20-4-98
M0123	B0212	ADVENTURE	20-3-98	20-4-98
M0123	B0400	ADVENTURE	20-3-98	20-4-98
M0123	B0555	ADVENTURE	20-3-98	20-4-98
M0880	B0001	CLASSIC	21-3-98	21-4-98

Table 6: Issue table

The key in Table 6 is the book code.

Let us understand what has been done here. The first normal form table that we saw earlier has been split into two tables - the member table and the book table. Both are independent tables now, and are linked through the member code.

Let us look at the definition of Second normal form. According to C. J. Date, a table is in its Second normal form when it is already in its First normal form and every field, which is not a key, is fully dependent on the primary key.

To convert a table to 2NF, follow these steps:

- Find and remove fields that are related to only a part of the key

- Group the removed items in another table

- Assign the new table with a key that is a part of the old composite key

There is still some amount of redundancy in the tables that we have created. We will now convert this table to its Third normal form.

Third Normal Form

A table is said to be in its Third normal form when it is in its 2NF and every field, which is not a key, is functionally dependent on just the primary key. The tables have been further decomposed as shown. The Member table, Table 7 remains as it is.

Member Code	Member Name	Contact
M001	JOE	414 2319
M008	SAM	890 0909
M067	PENNY	421 9498
M0123	JENNY	567 0967
M0880	PETER	328 9565

Table 7: Member Table

The Issue table is further decomposed. Look at Table 8:

Member Code	Book Code	Issue Date	Due Date
M001	B0020	18-3-98	18-4-98
M008	B0189	18-3-98	18-4-98
M008	B0090	18-3-98	18-4-98
M008	B0091	18-3-98	18-4-98
M067	B0656	19-3-98	19-4-98
M0123	B0189	20-3-98	20-4-98
M0123	B0212	20-3-98	20-4-98
M0123	B0400	20-3-98	20-4-98
M0123	B0555	20-3-98	20-4-98
M0880	B0001	21-3-98	21-4-98

Table 8: Issue Table

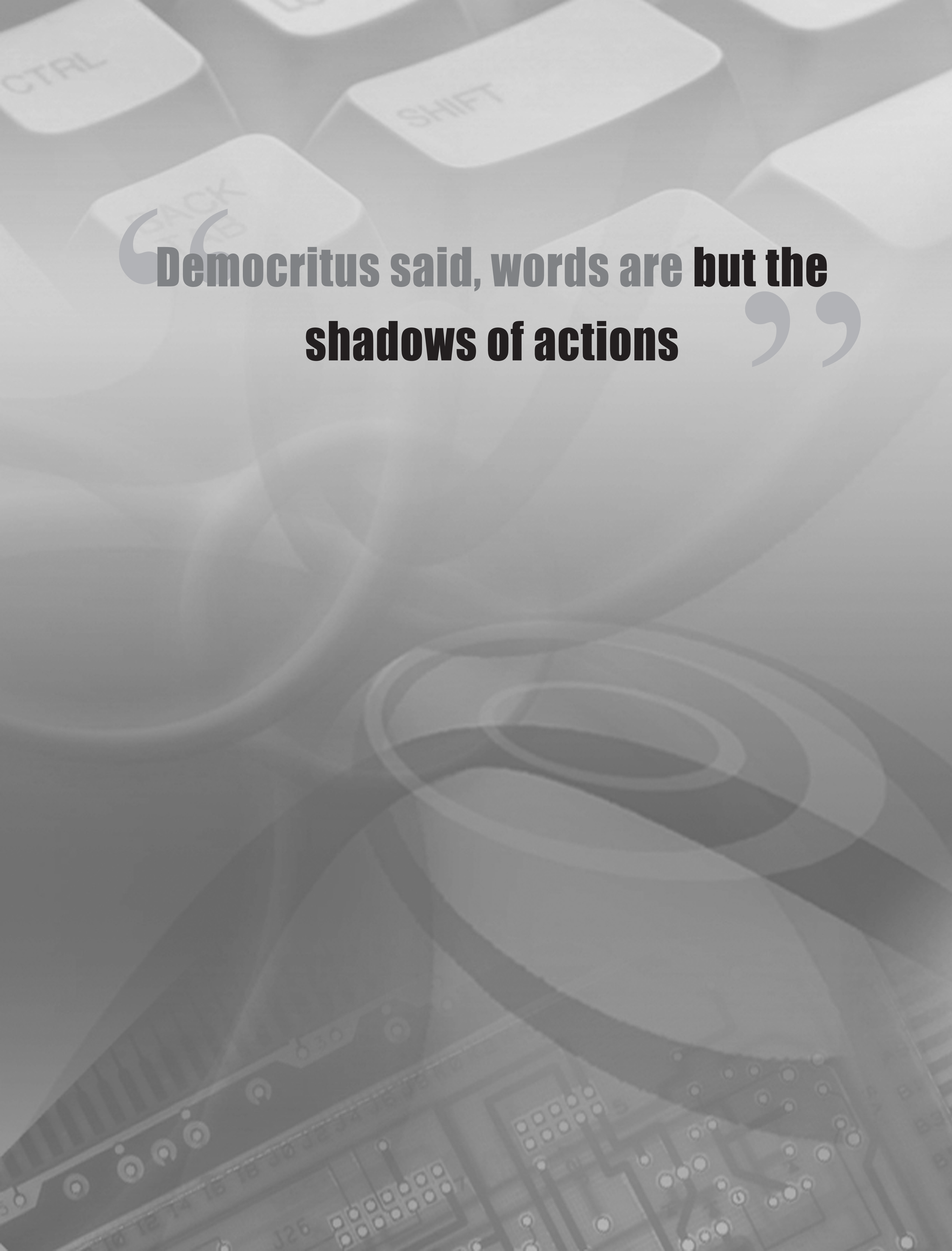
The Book table, Table 9 is as follows:

Book Code	Category-Code
B0020	SCIENCE
B0189	ASTRO
B0090	ASTRO

B0091	ASTRO
B0656	ROMANCE
B0189	ADVENTURE
B0212	ADVENTURE
B0400	ADVENTURE
B0555	ADVENTURE
B0001	CLASSIC

Table 9: Book Table

We have arrived at the Third normal form. In situations where the third normal form is not adequate, the Boyce-Codd normal form is used.



“Democritus said, words are but the shadows of actions”