Project Brief

Project Name:

Student Information System

Objective:
The main aim of the project is to develop a complete web application of student information system of City Public School which will facilitate the user to manage all the information regarding Student information system.

Undertaken BY:

Muhammad Tanver Iqbal
Shaukat Ali

Supervised By:

Mr. Ali Samad Tauni
Lecturer of Department Of Computer Science & IT.

Operating System:

Windows 7(32bit) Ultimate

Project Started:

June-01-2011

Project Finished:

Nov-04-2011

Source Language:
“GOD HELPS THOSE WHO HELP THEMSELVES”

MAY ALLAH HELP ME FOR SUCCESS (AMEEN)
DEDICATION

My Loving Parents
    Whose Support
    Give me Strength
    And determination
    To Accomplish my Goal
Acknowledgement

First of all, my deepest gratitude to Almighty ALLAH: the most beneficent, compassionate, most merciful and most gracious whose favor and kindness made it possible for me to complete this project work.

A very special thanks and appreciation goes to my parents and other family members for always encouraging me. They really deserve me for enduring my problems with great patience and love and whose endless prayers are a source of determination for me.

I wish to thank to Mr. Dr. Shahid Naveed Head of Department for providing me the facility and best environment to complete my task.

I am also greatly indebted to my respectable teacher Mr. Ali Samad Tauni for his supervision, kind support, unforgettable devotion and encouraging behavior.

I am also thankful to all staff members of the Department of Computer Science & IT for their coordination.

Student Name

__________________________

Student Information System
ABSTRACT

Student Information System is a model system to store information about Student information system of the City public school like teacher profile, student records, classes, subjects, fees, timetables, datasheets and reports. The system is designed to meet the purpose of dealing with student information system.

As project mainly concentrate on Student information system so keeping the friendly user interface the system should provide all necessary Student information facilities. A Login Form which asks the User to browse the whole system and perform different operations step by step such as Saving, Updating, Deleting and loading records as well as providing the facility to Administrator to generate the sequence by hiding un-necessary tasks from the user. Also the system is capable of managing records for all the students. The system is also capable to generate reports about student results, classes, timetable, datesheets and certificates.

Currently the System stores information in databases and retrieve information on the interfaces with the help of data access.
# Table Of Contents

<table>
<thead>
<tr>
<th>Serial No.</th>
<th>Chapter Details</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Chapter No 1</strong></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Organization Profile</td>
<td>11</td>
</tr>
<tr>
<td>1.2</td>
<td>Problem Statement</td>
<td>11</td>
</tr>
<tr>
<td>1.3</td>
<td>Project Feasibility</td>
<td>11</td>
</tr>
<tr>
<td>1.4</td>
<td>Project Scope</td>
<td>13</td>
</tr>
<tr>
<td>1.5</td>
<td>Task Dependency Table</td>
<td>14</td>
</tr>
<tr>
<td>1.6</td>
<td>Network Activity Diagram</td>
<td>15</td>
</tr>
<tr>
<td>1.7</td>
<td>Tools and Technology</td>
<td>16</td>
</tr>
<tr>
<td>1.8</td>
<td>Vision Document</td>
<td>17</td>
</tr>
<tr>
<td>1.9</td>
<td>Risk List</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter No 2</strong></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>System Specification</td>
<td>21</td>
</tr>
<tr>
<td>2.2</td>
<td>Identifying External Entities</td>
<td>23</td>
</tr>
<tr>
<td>2.3</td>
<td>Context Level Data Flow Diagram</td>
<td>25</td>
</tr>
<tr>
<td>2.4</td>
<td>User Characteristics</td>
<td>26</td>
</tr>
<tr>
<td>2.5</td>
<td>Allocate Requirement</td>
<td>26</td>
</tr>
<tr>
<td>2.6</td>
<td>Prioritize Requirement</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>Requirement Traceability Matrix</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter No 3</strong></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>High Level Use-Case Diagram</td>
<td>35</td>
</tr>
<tr>
<td>3.2</td>
<td>Analysis Level Use-Case Diagram</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>Use-Case Description</td>
<td>38</td>
</tr>
<tr>
<td>3.4</td>
<td>Domain Model</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>System Sequence Diagram</td>
<td>45</td>
</tr>
<tr>
<td>3.6</td>
<td>Sequence Diagram</td>
<td>59</td>
</tr>
<tr>
<td>3.7</td>
<td>Collaboration Diagram</td>
<td>70</td>
</tr>
<tr>
<td>3.8</td>
<td>Design Class Diagram</td>
<td>77</td>
</tr>
<tr>
<td>3.9</td>
<td>State Chart Diagram</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter No 4</strong></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction to Database</td>
<td>81</td>
</tr>
<tr>
<td>4.2</td>
<td>Database system</td>
<td>81</td>
</tr>
<tr>
<td>4.3</td>
<td>Advantages of Database System</td>
<td>84</td>
</tr>
<tr>
<td>4.4</td>
<td>Benefits of Database Approach</td>
<td>84</td>
</tr>
</tbody>
</table>

*Student Information System*
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>Data Independence</td>
<td>85</td>
</tr>
<tr>
<td>4.6</td>
<td>Data Integrity</td>
<td>85</td>
</tr>
<tr>
<td>4.7</td>
<td>Identification of Entities</td>
<td>86</td>
</tr>
<tr>
<td>4.8</td>
<td>Attributes</td>
<td>86</td>
</tr>
<tr>
<td>4.9</td>
<td>Normalization</td>
<td>89</td>
</tr>
<tr>
<td>4.10</td>
<td>Database Design</td>
<td>92</td>
</tr>
<tr>
<td>4.11</td>
<td>Data Modeling</td>
<td>93</td>
</tr>
<tr>
<td>4.12</td>
<td>Snap-Shots of Tables</td>
<td>100</td>
</tr>
<tr>
<td>4.13</td>
<td>Data Model Showing Relationships</td>
<td>103</td>
</tr>
<tr>
<td><strong>Chapter No 5</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Testing</td>
<td>105</td>
</tr>
<tr>
<td>5.2</td>
<td>Testing Strategy</td>
<td>105</td>
</tr>
<tr>
<td>5.3</td>
<td>Test Cases</td>
<td>105</td>
</tr>
<tr>
<td><strong>Chapter No 6</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Site Map</td>
<td>110</td>
</tr>
<tr>
<td>6.2</td>
<td>Gate Clerk</td>
<td>110</td>
</tr>
<tr>
<td>6.3</td>
<td>Computer Weight Clerk</td>
<td>114</td>
</tr>
<tr>
<td>6.4</td>
<td>Managing Partner</td>
<td>115</td>
</tr>
<tr>
<td>6.5</td>
<td>Bale Weight Clerk</td>
<td>115</td>
</tr>
<tr>
<td>6.6</td>
<td>Accountant</td>
<td>116</td>
</tr>
<tr>
<td>6.7</td>
<td>Reports</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>Reverences</td>
<td>123</td>
</tr>
</tbody>
</table>

Chapter 1

INTRODUCTION

1.1. Organization Profile
The system which I am going to develop as my final project is the Student Information System of City public school. This document is developed to serve as a starting point of the software development process.

1.1. Organization Profile:

City Public School established on 23\textsuperscript{th} of July 2004. It is registered by the Bahawalpur Board. There are two partners who are running this System. The School is located at Maroot.
The head office of the school is located in **ALLAMA IQBAL TOWN(A)**.

1.2. Problem Statement:

The existing system uses the concepts and basis of conventional system with clerical office and clerical staff to maintain information. The existing system works manually. The School itself has to maintain, manage and set policies from the format and maintain data. This project work is related to the development of Student Information System of **City Public School**. This project is build because the existing system is manual and is not efficient to provide all information in due time.

1.3. Project/Product Feasibility Report:

1.3.1. Technical Feasibility:

The system which is to be developed is Web based, and PHP technology along with MYSQL Server will be used to develop it. The project team has got the status to use these technologies. The software required for doing this project is easily available.

1.3.2. Operational Feasibility:

The staff that will be designed software so that it is easy to use. They will additionally provided with help and guidance (if needed) to operate the software.

1.3.3. Economic Feasibility:

There is no need for purchasing the tools and license used during the development of the project. All tool and technologies that are required during development are already with the development team. This makes the development economically feasible. Only costing factor is the effort of the project members and time that is utilized in project development. The maintenance cost and operation cost is there.

1.3.4. Schedule Feasibility:

Time is an important factor. I have got the required resources to complete the project on time. I am in the final semester of my program and there is sufficient time available to me for completing this project on the required date and time.
1.3.5. Specification Feasibility:

The project team has a clear picture of what we have to develop and what the system must have in it to be successful. The project team will have a complete and clearer picture when we are through with the requirements specification and gathering phase. The requirements are becoming clearer and definite with the passage of time.

1.3.6. Information Feasibility:

The information regarding its completion, reliability, and meaningfulness is ensured by the use of the Internet, books, and software development requirements. The project will itself be informative and helpful to the concerned authorities after completion.

1.3.7. Motivational Feasibility:

The client’s staffs that will actually using the system are motivated to use this system as one of the goals of the system is helping them with their work.

1.3.8. Legal & Ethical Feasibility:

The system is free of any infringements or liabilities. It is not violating any legal or ethical values.

1.4. Project Scope:

This project work covers the whole data about the Student Information System of the City Public school. This Project includes teacher profile, student record, fees, classes, subject’s information, datesheet, timetable and reports.

1. A MYSQL Server that provides interaction with the system to the users that are physically distributed on same location.
2. A centralized database.
3. The system will have adequate redundancy to ensure fail-safe operation.
4. It will handle the student records.
5. The system will be accessible to all who have a valid login; users with a valid login can access the system virtually anywhere and do the required task where computer and Internet facilities are available.
6. It is not costly, only single computer is sufficient to manage the whole record.
7. It is not a time consuming system.
8. It needs only a single person with little computer literacy.
9. It is fast and generates reports within no time.
10. Insertion, Deletion and updating of the records are very easy.
11. It is easy to add or remove new records according to the requirements.
12. It has backup facility and data can be stored on any removable media.
### 1.5. Task Dependency Table:

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Task ID</th>
<th>Duration</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Analysis</strong></td>
<td>T1</td>
<td>12</td>
<td>None</td>
</tr>
<tr>
<td>Business Analysis</td>
<td>T2</td>
<td>3</td>
<td>None</td>
</tr>
<tr>
<td>Formulation</td>
<td>T3</td>
<td>3</td>
<td>None</td>
</tr>
<tr>
<td>Develop Vision</td>
<td>T4</td>
<td>2</td>
<td>T2</td>
</tr>
<tr>
<td>Developing User Hierarchy</td>
<td>T5</td>
<td>1</td>
<td>T2,T3</td>
</tr>
<tr>
<td>Developing Use Cases</td>
<td>T6</td>
<td>7</td>
<td>T2,T3</td>
</tr>
<tr>
<td>Refining Use Case Model</td>
<td>T7</td>
<td>2</td>
<td>T6</td>
</tr>
<tr>
<td><strong>Analysis Modeling</strong></td>
<td>T8</td>
<td>14</td>
<td>T1</td>
</tr>
<tr>
<td>Use Case Analysis</td>
<td>T9</td>
<td>6</td>
<td>T7</td>
</tr>
<tr>
<td>Content/Database Modeling</td>
<td>T10</td>
<td>3</td>
<td>T9</td>
</tr>
<tr>
<td>Interaction Modeling</td>
<td>T11</td>
<td>3</td>
<td>T9,T10</td>
</tr>
<tr>
<td>Functional Modeling</td>
<td>T12</td>
<td>2</td>
<td>T9,T10,T11</td>
</tr>
<tr>
<td>Configuration Modeling</td>
<td>T13</td>
<td>2</td>
<td>T9</td>
</tr>
<tr>
<td><strong>Design</strong></td>
<td>T14</td>
<td>10</td>
<td>T8</td>
</tr>
<tr>
<td>Identify Design Elements</td>
<td>T15</td>
<td>1</td>
<td>T12</td>
</tr>
<tr>
<td>Interface Design</td>
<td>T16</td>
<td>3</td>
<td>T15</td>
</tr>
<tr>
<td>Content Design(Database,Classes)</td>
<td>T17</td>
<td>5</td>
<td>T15</td>
</tr>
<tr>
<td>Navigational Design</td>
<td>T18</td>
<td>1</td>
<td>T15</td>
</tr>
<tr>
<td>Review Design</td>
<td>T19</td>
<td>4</td>
<td>T16,T17,T18</td>
</tr>
<tr>
<td><strong>Implementation</strong></td>
<td>T20</td>
<td>21</td>
<td>T14</td>
</tr>
<tr>
<td>Structure Implementation Model</td>
<td>T21</td>
<td>3</td>
<td>T19</td>
</tr>
<tr>
<td>Implement Design Elements</td>
<td>T22</td>
<td>15</td>
<td>T21</td>
</tr>
<tr>
<td>Review Code</td>
<td>T23</td>
<td>3</td>
<td>T22</td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td>T24</td>
<td>5</td>
<td>T20</td>
</tr>
<tr>
<td>Test Content</td>
<td>T25</td>
<td>3</td>
<td>T23</td>
</tr>
<tr>
<td>Test Interface</td>
<td>T26</td>
<td>2</td>
<td>T23</td>
</tr>
<tr>
<td>Test Navigation</td>
<td>T27</td>
<td>2</td>
<td>T23</td>
</tr>
<tr>
<td>Test Configuration</td>
<td>T28</td>
<td>3</td>
<td>T23</td>
</tr>
<tr>
<td>Test Security</td>
<td>T29</td>
<td>2</td>
<td>T23</td>
</tr>
<tr>
<td>Test Performance</td>
<td>T30</td>
<td>3</td>
<td>T23</td>
</tr>
<tr>
<td>Review Test Results</td>
<td>T31</td>
<td>2</td>
<td>T25,T26,T27,T28,T29,T30</td>
</tr>
<tr>
<td><strong>Deployment</strong></td>
<td>T32</td>
<td>14</td>
<td>T24</td>
</tr>
<tr>
<td>Review for modification</td>
<td>T33</td>
<td>3</td>
<td>T31</td>
</tr>
<tr>
<td>Assess and Advocate Quality</td>
<td>T34</td>
<td>2</td>
<td>T33</td>
</tr>
<tr>
<td>Make Deployment Plan</td>
<td>T35</td>
<td>1</td>
<td>T34</td>
</tr>
<tr>
<td>Deploy the Product</td>
<td>T36</td>
<td>2</td>
<td>T35</td>
</tr>
<tr>
<td>Train Users</td>
<td>T37</td>
<td>6</td>
<td>T36</td>
</tr>
</tbody>
</table>
1.6. Network Activity Diagram:

[Diagram of network activity diagram]

Start → T2 → T3 → T6 → T7 → T9 → T10 → T12 → T13 → T15 → T16 → T17 → T18 → T19 → T21 → T22 → T23 → T25 → T26 → T27 → T28 → T29 → T30 → Finish

- Req. Analysis
- Analysis Modeling
- Design
- Implementation
- Testing
- Deployment
1.7. Tools and Technology:

Student information system of City Public School is web application, which makes selection of the technologies required for the implementation of the project.

1.7.1. Front-End Technology (PHP):

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on web sites or elsewhere. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP parser compiles input to produce byte code for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Asp.net, Sun Microsystems' JavaServer Pages, and mod_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD).

1.7.2. Back-End Technology (SQL Server 2008):

For saving the data and user information along with efficient retrieval of data, the MYSQL will be used. The reason of using this technology is that MYSQL is a widely available and secure source of relational database management system (RDBMS) that uses Structured Query Language (SQL), the most popular language for adding, accessing, and processing data in a database. MYSQL is noted mainly for its speed, reliability, and flexibility. Regarding my project there will be online
access to database updating and deletion that require speed and efficiency. Therefore MYSQL is suitable for our project as a back end technology.

1.7.3. Microsoft Project:

MS Project is project management software that is ideal for the designing and documentation of large projects. MS Project is designed to assist project managers in developing plans, assigning resources to tasks, tracking progress, managing budgets and analyzing workloads.

1.7.4. Microsoft Visio:

Microsoft Visio is used to make diagrams for design phase and documentation.

1.8. Vision Document:

The Student information System of City Public School is an effort of automating the education system of the City Public School. This system will enable an efficient and reliable management of the Student information System of City Public School. This system is moving towards automation in order to keep up with the demands of the modern times. My objective is to successfully automate its admission management system. The scope of this project is limited to education management system only, due to a number of reasons. Covering all the departments in the limited time span of four months by a small team of one member is not possible! Secondly, Education system has its own software department that has allocated some skilled professionals for working on the other domains. This student information system shall be a very comprehensive system that covers all aspects of student information system.

1.9. Risk list:

The possible risks that can occur during the course of the project are listed below:

<table>
<thead>
<tr>
<th>#</th>
<th>Risks</th>
<th>Risk Type</th>
<th>Probability</th>
<th>Mitigation Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Student Information System</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Requirement Analysis</td>
<td>Risk Type</td>
<td>Risk Level</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td>-----------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>The schedule pressure can force some function points to be changed or dropped from being implemented as planned in the planning phase.</td>
<td>Schedule Risk</td>
<td>50%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• We have divided the whole process in modules.</td>
<td></td>
<td>• All activities are listed on the network diagram with proper planning, and sufficient time allocated to each activity.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>The requirements can change over time.</td>
<td>Scope Risk</td>
<td>40%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Sufficient time is provided for requirement elicitation.</td>
<td></td>
<td>• The applicable changes will be handled if possible.</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>The product scope can keep expanding.</td>
<td>Scope Risk</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The product will be built by using relatively independent modules so that any new functionality can be added.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>The transaction time can be a bit higher depending upon the internet speed</td>
<td>Technological Risk</td>
<td>25%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A relatively simple and efficient solution will be found and tried.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>This is the largest project the team has ever attempted, so it can result in some pressures and problems because of the lack of experience.</td>
<td>Organization Risk</td>
<td>15%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Experienced people in the related fields will be consulted.</td>
<td></td>
<td>• Lack of experience will be reduced by the usage of knowledge and technology.</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Although the team members have appropriate skills, but they have not used their skills on such a broader scope.</td>
<td>People Risk</td>
<td>15%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Experienced people in the related fields will be consulted.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>The users of the system might need some time to get familiarize with the system</td>
<td>Technological Risk</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The user will be provided with sufficient on the hands help to learn the usage of the system early and easily.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Chapter 2 REQUIREMENT ANALYSIS**
In this chapter I have discussed requirements engineering process, which provides the appropriate mechanism for understanding what the customer wants, analyzing needs, assessing feasibility, negotiating a reasonable solution, specifying the solution
unambiguously, validating the specification and managing the requirements as they are transformed into an operational system.

1. Requirements elicitation
2. Requirements analysis and negotiation
3. Requirements specification
4. System modeling
5. Requirements validation
6. Requirements management

Here, requirements specification is to be discussed. Requirements specification would lead to the following steps:

1. Identify external interfaces
2. Development of context diagram
3. User characteristics
4. Allocate requirements
5. Prioritize requirements
6. Development of requirements traceability matrix

2.1. System Specification:
2.1.1. City Public School Organization Chart:

![Organizational Chart Diagram]

- **Top Management**
  - Managing Partner
  - Partners
  - Chief Selector

- **Middle Management**
  - Manager

- **Lower Management**
  - Labor In charge
  - Labors
  - Mechanical Staff
  - Selection Officer
  - Accountant
  - Cashier
  - Clerk

*City Public School*
2.1.2. Summary of Requirements (Initial Requirements):

Our purposed system must fulfill the requirements as follow:

1. Database administrator shall register all the users. He will be able to update, recover and backup the database. He will set the user privileges. He will also handle day to day matters regarding the database. He will also define the basic entities like the teacher, student, fees, timetable etc.

2. A user must login to the system to be able to use the system. A user shall be able to view different types of reports regarding the system according to his privilege level. He must logout of the system after using the system.

3. When vehicle enters in the factory the Gate Clerk enters information about the Party and product and then issue Gate Pass.

4. Then it comes to the Computer Balance Machine for weight where the computer weight clerk enters vehicle code, party code and product code and sends it into the unloading area.

5. After unloading the vehicle returns and again the computer weight clerk weight the empty vehicle and calculate the net weight.

6. The Selection Officer selects some sample of the product and checks the quality of the product by checking that how much wastage in the product.

7. Then the Chief Selector deducts the amount of wastage in the product and he obtained the original weight of product that is useable for production.

8. The Managing Partner of the Factory decides the rate of the product (Raw Cotton) with the party and pay the amount to the party
9. The Cotton Lint which is in the form of Bales for sale checked by the Bale Weighing Clerk, he checks the moisture in each Bale and form a Lot which is of 100 Bales and gives a Lot No.

10. The factory sales its products to different textile mills, oil mills and different traders.

11. The Sale Order which consist of product code, sale voucher no and Lot no on which total payment payable to factory is decided.

12. The Sale Voucher decides that which Party is purchasing Factory’s product.

13. The factory has Bank Accounts which are used for debit and credit.

14. The Bank Account has Account type, Account Holder and Bank.

15. The transaction slip has Account id and debit or credit amount.

2.2. Identifying External Entities:

The identification of the external entities is based on the information contained in the above abstract. The Identification of External entities is done in two phases.

2.2.1. Over Specify Entities from Abstract:

1. Admin
2. Teacher
3. Student
4. Class
5. Subject
6. Fees
7. Result
8. Timetable
9. Datesheet
10. Module
11. Contents
12. Setting
13. Events

2.2.2. Refined Entities:

After refining the entities we found the following entities to fit our System Logic.

1. Admin
2. Teacher
3. Student
4. Class
5. Subject
6. Fees
7. Result
8. Timetable
9. Datesheet

2.3. Context Level Data Flow Diagram:
2.4. User Characteristics:
### Initial Requirements

<table>
<thead>
<tr>
<th>Para #</th>
<th>Initial Requirements</th>
</tr>
</thead>
</table>

---

Student Information System
## Project Report

1.0 DB Admin “shall” register all users.
1.0 DB Admin “shall” set privileges for all users.
1.0 DB Admin “shall” update database.
2.0 A user “shall” login to the system.
2.0 A user “shall” logout of the system.
2.0 A user “shall” view his action list.
3.0 Gate Clerk “shall” enter Party Information.
3.0 Gate Clerk “shall” enter Product information
3.0 Gate Clerk “shall” issue Gate Pass
4.0 Computer Weight Clerk “shall” weight the Vehicle.
4.0 Computer Weight Clerk “shall” issue Computer Weight Note Slip.
5.0 Selection Officer “shall” select sample of Raw Cotton.
5.0 Selection Officer “shall” calculate the amount of wastage.
6.0 Chief Selector “shall” check the Sample.
6.0 Chief Selector “shall” deduct the amount of wastage.
6.0 Chief Selector “shall” issue Slip.
7.0 Managing Partner “shall” receive the Slip.
7.0 Managing Partner “shall” decide the rate
8.0 Bale Weight Clerk “shall” check moisture in Bales
8.0 Bale Weight Clerk “shall” form Lots
9.0 Sale Manager “shall” issue sale order and sale voucher
10.0 Accountant “shall” receive D.D from Party

### 2.5. Allocate Requirements:

<table>
<thead>
<tr>
<th>Para #</th>
<th>Initial Requirements</th>
<th>Use Case Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>DB Admin “shall” register all users.</td>
<td>UC_Register</td>
</tr>
<tr>
<td>1.0</td>
<td>DB Admin “shall” update database.</td>
<td>UC_Update</td>
</tr>
<tr>
<td>1.0</td>
<td>DB Admin “shall” set privileges for all users.</td>
<td>UC_Privileges</td>
</tr>
<tr>
<td>2.0</td>
<td>A user “shall” login to the system.</td>
<td>UC_Login</td>
</tr>
<tr>
<td>2.0</td>
<td>A user “shall” logout of the system.</td>
<td>UC Logout</td>
</tr>
<tr>
<td>2.0</td>
<td>User “shall” view reports.</td>
<td>UC_Reports_View</td>
</tr>
<tr>
<td>3.0</td>
<td>Gate Clerk “shall” enter Party Information.</td>
<td>UC_Enter_Party</td>
</tr>
<tr>
<td>3.0</td>
<td>Gate Clerk “shall” enter Product information.</td>
<td>UC_Enter_Product</td>
</tr>
</tbody>
</table>
### 3.0 Gate Clerk “shall” issue Gate Pass

UC_GatePass_Issue

### 4.0 Computer Weight Clerk “shall” weight the Vehicle.

UC_Weight_Vehicle

### 4.0 Computer Weight Clerk “shall” issue Computer Weight Note Slip.

UC_WeightSlip_Issue

### 5.0 Selection Officer “shall” select sample of Raw Cotton

UC_Select

### 5.0 Selection Officer “shall” calculate the amount of wastage.

UC_Calculate

### 6.0 Chief Selector “shall” check the Sample.

UC_Check

### 6.0 Chief Selector “shall” deduct the amount of wastage.

UC_Deduct_Wastage

### 6.0 Chief Selector “shall” issue Slip.

UC_Issue_Slip

### 7.0 Managing Partner “shall” receive the Slip.

UC_Slip_Receive

### 7.0 Managing Partner “shall” decide the rate

UC_Decide_Rate

### 8.0 Bale Weight Clerk “shall” check moisture in Bales

UC_Check_Moisture

### 8.0 Bale Weight Clerk “shall” form Lots

UC_Lots

### 9.0 Sale Manager “shall” issue sale order and sale voucher

UC_Sale_Voucher

### 10.0 Accountant “shall” receive D.D from Party

UC_Receive_DD

### 2.6. Prioritize Requirements:

<table>
<thead>
<tr>
<th>Para #</th>
<th>Rank</th>
<th>Initial Requirements</th>
<th>Use Case ID</th>
<th>Use Case Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Highest</td>
<td>DB Admin “shall” register all users.</td>
<td>UC_01</td>
<td>UC_Register</td>
</tr>
<tr>
<td>1.0</td>
<td>Highest</td>
<td>DB Admin “shall” set privileges for all users.</td>
<td>UC_02</td>
<td>UC_Privileges</td>
</tr>
<tr>
<td>2.0</td>
<td>Highest</td>
<td>A user “shall” login to the system.</td>
<td>UC_03</td>
<td>UC_Login</td>
</tr>
<tr>
<td>2.0</td>
<td>Highest</td>
<td>A user “shall” logout of the system.</td>
<td>UC_04</td>
<td>UC_Logout</td>
</tr>
<tr>
<td>3.0</td>
<td>Medium</td>
<td>Gate Clerk “shall” enter Party Information.</td>
<td>UC_05</td>
<td>UC_Enter_Party</td>
</tr>
<tr>
<td>3.0</td>
<td>Medium</td>
<td>Gate Clerk “shall” enter Product Information.</td>
<td>UC_06</td>
<td>UC_Enter_Product</td>
</tr>
<tr>
<td>3.0</td>
<td>Medium</td>
<td>Gate Clerk “shall” issue Gate Pass</td>
<td>UC_07</td>
<td>UC_GatePass_Issue</td>
</tr>
<tr>
<td>4.0</td>
<td>Medium</td>
<td>Computer Weight Clerk “shall” weight the Vehicle.</td>
<td>UC_08</td>
<td>UC_Weight_Vehicle</td>
</tr>
<tr>
<td>4.0</td>
<td>Medium</td>
<td>Computer Weight Clerk “shall” issue Computer Weight Note Slip.</td>
<td>UC_09</td>
<td>UC_WeightSlip_Issue</td>
</tr>
<tr>
<td>5.0</td>
<td>Medium</td>
<td>Selection Officer “shall” select sample of Raw Cotton</td>
<td>UC_10</td>
<td>UC_Select</td>
</tr>
</tbody>
</table>
### 2.7. Requirements Traceability Matrix:

<table>
<thead>
<tr>
<th>#</th>
<th>Para #</th>
<th>Initial Requirements</th>
<th>Build</th>
<th>Use Case Name</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>DB Admin “shall” register all users.</td>
<td>B1</td>
<td>UC_Register</td>
<td>Business</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>DB Admin “shall” set privileges for all users.</td>
<td>B1</td>
<td>UC_Privileges</td>
<td>Business</td>
</tr>
<tr>
<td>3</td>
<td>2.0</td>
<td>A user “shall” login to the system.</td>
<td>B1</td>
<td>UC_Login</td>
<td>Business</td>
</tr>
<tr>
<td>4</td>
<td>2.0</td>
<td>A user “shall” logout of the system.</td>
<td>B1</td>
<td>UC_Logout</td>
<td>Business</td>
</tr>
<tr>
<td>5</td>
<td>2.0</td>
<td>User “shall” view reports.</td>
<td>B1</td>
<td>UC_Reports_View</td>
<td>Business</td>
</tr>
<tr>
<td>6</td>
<td>3.0</td>
<td>Gate Clerk “shall” enter Party Information.</td>
<td>B1</td>
<td>UC_Enter_Party</td>
<td>Business</td>
</tr>
<tr>
<td>7</td>
<td>3.0</td>
<td>Gate Clerk “shall” enter Product Information.</td>
<td>B1</td>
<td>UC_Enter_Product</td>
<td>Business</td>
</tr>
<tr>
<td>7</td>
<td>3.0</td>
<td>Gate Clerk “shall” issue Gate Pass</td>
<td>B1</td>
<td>UC_GatePass_Issue</td>
<td>Business</td>
</tr>
<tr>
<td>8</td>
<td>4.0</td>
<td>Computer Weight Clerk “shall” weigh the Vehicle.</td>
<td>B1</td>
<td>UC_Weight_Vehicle</td>
<td>Business</td>
</tr>
<tr>
<td>9</td>
<td>4.0</td>
<td>Computer Weight Clerk “shall” issue Computer Weight Note Slip.</td>
<td>B1</td>
<td>UC_WeightSlip_Issue</td>
<td>Business</td>
</tr>
<tr>
<td>10</td>
<td>5.0</td>
<td>Selection Officer “shall” select</td>
<td>B1</td>
<td>UC_Select</td>
<td>Business</td>
</tr>
</tbody>
</table>
Project Report

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>5.0</td>
<td>Selection Officer “shall” calculate the amount of wastage.</td>
<td>UC_Calculate</td>
<td>Business</td>
</tr>
<tr>
<td>12</td>
<td>6.0</td>
<td>Chief Selector “shall” check the Sample.</td>
<td>UC_Check</td>
<td>Business</td>
</tr>
<tr>
<td>13</td>
<td>6.0</td>
<td>Chief Selector “shall” deduct the amount of wastage.</td>
<td>UC_Deduct_Wastage</td>
<td>Business</td>
</tr>
<tr>
<td>14</td>
<td>6.0</td>
<td>Chief Selector “shall” issue Slip.</td>
<td>UC_Issue_Slip</td>
<td>Business</td>
</tr>
<tr>
<td>15</td>
<td>7.0</td>
<td>Managing Partner “shall” decide the rate</td>
<td>UC_Decide_Rate</td>
<td>Business</td>
</tr>
<tr>
<td>16</td>
<td>8.0</td>
<td>Bale Weight Clerk “shall” check moisture in Bales</td>
<td>UC_Check_Moisture</td>
<td>Business</td>
</tr>
<tr>
<td>17</td>
<td>8.0</td>
<td>Bale Weight Clerk “shall” form Lots</td>
<td>UC_Lots</td>
<td>Business</td>
</tr>
<tr>
<td>18</td>
<td>9.0</td>
<td>Sale Manager “shall” issue sale order and sale voucher</td>
<td>UC_Sale_Voucher</td>
<td>Business</td>
</tr>
<tr>
<td>19</td>
<td>10.0</td>
<td>Accountant “shall” receive D.D from Party</td>
<td>UC_Receive_DD</td>
<td>Business</td>
</tr>
</tbody>
</table>

Chapter3  OBJECT ORIENTED ANALYSIS AND DESIGN

3.1. High Level Use case Diagram

3.2. Analysis Level Use Case Diagram

Student Information System 30
The objective of Object Oriented Analysis and Design is to develop a model that describes computer software as it works to satisfy a set of requirements. After understanding the current situation of the problem domain the team is ready to strive for the solution by using OOAD approach.

**Actors:**

- Student Information System
Following are the actors that interact with Sales and Purchase Management System of AL-Noor Cotton Ginners and Oil Mills.

- Administrator
- User
- Gate Clerk
- Computer Weight Clerk
- Selection Officer
- Chief Selector
- Managing Partner
- Bale Weight Clerk
- Sale Manager
- Accountant

Figure:

**Administrator:**

![Administrator Diagram](attachment:image.png)

**User:**
Gate Clerk:

Computer Weight Clerk:

Selection Officer:

Chief Selector:
Project Report

Managing Partner:

Bale Weight Clerk:

Sale Manager:

Accountant:
ACCOUNTANT
3.1. High Level Use case Diagram:
3.2. Analysis Level Use Case Diagram:

**AL-NOOR INDUSTRIES**

**USER**
- Login «extends » Authorization
- UC_Login «include» Invalid Input

**View Reports**
- UC_Reports_View «extends » Select Report

**Register**
- UC_Register «extends » Cancel
- New User «extends » Main Form
- Save «include»

**Register**
- UC_Privileges «extends » Cancel
- Select User «extends » Set Privileges

**Update**
- UC_Update «extends » Update User
- Select User «extends »
### 3.3. Use case Description:

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Register</td>
</tr>
</tbody>
</table>

**Pre-Condition:** The DB Admin must login in the system with “administrator” login.

**Description:** This use case describes the creation of user accounts.

**Basic Flow:**

1. DB Admin clicks on create a new account link.
3. DB Admin fills in the required data (i.e. username, password of the user).
4. DB Admin clicks on the save button and exits.

**Alternative Flow:** The user can cancel the process at any time.

**Post-Condition:** DB Admin informs the desired person of his username and password.

**Extensions:** None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-02</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Update</td>
</tr>
</tbody>
</table>

**Pre-Condition:**  
The DB Admin must login the system as administrator.

**Description:**  
This use case describes how the DB Admin updates any form of the system.

**Basic Flow:**

1. DB Admin double clicks on the grid view in the form button.
2. DB Admin updates the required fields.
3. DB Admin clicks on the Update Button of the form.
4. DB Admin clicks on Exit Button.

**Alternative Flow:** The user can cancel the process at any time.

**Extensions:** None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-03</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Login</td>
</tr>
</tbody>
</table>

**Pre-Condition:**  
The user must have a valid username and password.

**Description:**  
This use case describes how a user can login the system.

**Basic Flow:**

1. The user opens the project and run the project.
2. The user enters his username and password and clicks on the Login button.
3. If the user is validated by the system he is shown main form depending on his privilege level.
4. The user is ready for doing his work.

**Alternative Flow:**

3. If the username and/or password is wrong the user remains on the same screen.

The user can cancel the process at any time.

**Post-Condition:** A log is created.

**Extensions:** None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-05</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Enter_Party</td>
</tr>
</tbody>
</table>

**Pre-Condition:**

Gate Clerk must log in.

Gate Clerk must open the Party Form.

**Description:** This use case describes how Gate Clerk enters Party information.

**Basic Flow:**

1. The Gate Clerk opens the Party Form from Main Menu.
2. Gate Clerk fills the required fields.
3. Gate Clerk clicks the Save Button.
4. Message Shown (Data Inserted).
5. Gate Clerk clicks the Exit Button.

**Alternative Flow:** The user can cancel the process at any time.

**Post-Condition:** The Data is entered in the Data Base.

**Extensions:** None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-06</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Enter_Product</td>
</tr>
</tbody>
</table>

**Pre-Condition:**

Gate Clerk must log in.

Gate Clerk must open the Product Form.

**Description:** This use case describes how Gate Clerk enters Product information.

**Basic Flow:**

1. The Gate Clerk opens the Product Form from Main Menu.
2. Gate Clerk fills the required fields.
3. Gate Clerk clicks the Save Button.
4. Message Shown (Data Inserted).
5. Gate Clerk clicks the Exit Button.

**Alternative Flow:** The user can cancel the process at any time.

**Post-Condition:** The Data is entered in the Data Base.

**Extensions:** None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-08</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UC-Name</strong></td>
<td>UC_Weight_Vehicle</td>
</tr>
</tbody>
</table>

**Pre-Condition:**
Computer Weight Clerk must log in.
Computer Weight Clerk must open the Weight Form.

**Description:** This use case describes how Computer Weight Clerk weights the Vehicle.

**Basic Flow:**
1. The Computer Weight Clerk opens the Weight Form from Main Menu.
2. Computer Weight Clerk fills the required fields.
3. Computer Weight Clerk clicks the Save Button.
4. Message Shown (Data Inserted).
5. Computer Weight Clerk clicks the Exit Button.

**Alternative Flow:**
The user can cancel the process at any time.

**Post-Condition:**
The Data is entered in the Data Base.

**Extensions:** None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-16</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UC-Name</strong></td>
<td>UC_Decide_Rate</td>
</tr>
</tbody>
</table>

**Pre-Condition:**
Managing Partner must log in.
Managing Partner must open the Purchase Rate Form.

**Description:** This use case describes how the Managing Partner inserts Rate.

**Basic Flow:**
1. The Managing Partner opens the Purchase Rate Form from Main Menu.
2. The Managing Partner fills the required fields.
3. The Managing Partner clicks the Save Button.
4. Message Shown (Data Inserted).
5. The Managing Partner clicks the Exit Button.
### Alternative Flow:
The user can cancel the process at any time.

### Post-Condition:
The Data is entered in the Data Base.

### Extensions:
None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-19</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Lots</td>
</tr>
</tbody>
</table>

### Pre-Condition:
Bale Weight Clerk must log in.

Bale Weight Clerk must open the Lots Form.

### Description:
This use case describes the insertion of Lots Information.

### Basic Flow:
1. The Bale Weight Clerk opens the Lots Form from Main Menu.
2. The Bale Weight Clerk fills the required fields.
3. The Bale Weight Clerk clicks the Save Button.
4. Message Shown (Data Inserted).
5. The Bale Weight Clerk clicks the Exit Button.

### Alternative Flow:
The user can cancel the process at any time.

### Post-Condition:
The Data is entered in the Data Base.

### Extensions:
None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-20</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Sale_Voucher</td>
</tr>
</tbody>
</table>

### Pre-Condition:
Sale Manager must log in.

Sale Manager must open the Sale Voucher Form.

### Description:
This use case describes how the sale information inserted.

### Basic Flow:
1. The Sale Manager opens the Sale Voucher Form from Main Menu.
2. The Sale Manager Clerk fills the required fields.
3. The Sale Manager Clerk clicks the Save Button.
4. Message Shown (Data Inserted).
5. The Sale Manager clicks the Exit Button.

### Alternative Flow:
The user can cancel the process at any time.
Post-Condition: The Data is entered in the Data Base.
Extensions: None.

<table>
<thead>
<tr>
<th>UC-ID</th>
<th>UC-21</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-Name</td>
<td>UC_Receive_DD</td>
</tr>
</tbody>
</table>

Pre-Condition:
Sale Manager must log in.
Sale Manager must open the Transaction Form.

Description: This use case describes how amount is inserted as credit or debit.

Basic Flow:
1. The Accountant opens the Transaction Form from Main Menu.
2. The Accountant Clerk fills the required fields.
3. The Accountant Clerk clicks the Save Button.
4. Message Shown (Data Inserted).
5. The Accountant clicks the Exit Button.

Alternative Flow: The user can cancel the process at any time.
Post-Condition: The Data is entered in the Data Base.
Extensions: None.
3.4. Domain Model:
3.5. System Sequence Diagram:

The UML system sequence diagram (SSD) illustrates events sequentially inputs from external source to the system. The SSD will define the system events and operations. System sequence diagrams are a timeline drawing of an expanded use case. Events are related by time with the top events occurring first. System events are the important items. These are events that cause a system response.

The System Sequence Diagrams of the Online Inventory System of WAPDA are shown below.
3.5.1 **UC_Login:**

1: Invoke Login Form

2: Show Login Form

3: Login (User Name, Password)

4: Validate(U,P)

5: Invalid Login

6: Show Main Form
3.5.2 UC_Logout:

1: Initiate Main Form

2: Show Main Form

3: Select Log Out from Main Form

4: Out from Log Table Adapter

5: Log Out From System
3.5.3 UC_Reports_View:

1. Invoke Main Form
2. Show Main Form
3. Select Required Report
4. Prepare Report
5. Show Report
6. Select Action (action)
7. Action()
8. Action() Result
3.5.4 UC_Enter_Party:

1: Invoke Main Form
2: Show Main Form
3: Select Party Form From Main Menu
4: Show Party Form
5: Fill Required Fields
6: Action(action)
7: Action()
8: Required Action Performed
3.5.5 UC_Enter_Product:

1: Invoke Main Form
2: Show Main Form
3: Select Product Form From Main Menu
4: Show Product Form
5: Fill Required Fields
6: Action(action)
7: Action()
8: Required Action Performed
3.5.6 UC_Enter_Vehicle:

1: Invoke Main Form
2: Show Main Form
3: Select Vehicle Form From Main Menu
4: Show Vehicle Form
5: Fill Required Fields
6: Action(action)
7: Action()
8: Required Action Performed
### 3.5.7 UC_Weight_Vehicle

1: Invoke Main Form

2: Show Main Form

3: Select Weight Form From Main Menu

4: Show Weight Form

5: Fill Required Fields

6: Action(action)

7: Action()

8: Required Action Performed
3.5.8 UC_Decide_Rate:

1: Invoke Main Form
2: Show Main Form
3: Select Purchase Rate Form From Main Menu
4: Show Purchase Rate Form
5: Fill Required Fields
6: Action(action)
7: Action()
8: Required Action Performed
3.5.9 UC_Lots:

1: Invoke Main Form
2: Show Main Form
3: Select Lots Form From Main Menu
4: Show Lots Form
5: Fill Required Fields
6: Action(action)
7: Action()
8: Required Action Performed
3.5.10 UC_Sale_Voucher:

1: Invoke Main Form

2: Show Main Form

3: Select Bank Account Form From Main Menu

4: Show Bank Account Form

5: Fill Required Fields

6: Action(action)

7: Action()

8: Required Action Performed
3.5.11 UC_Receive_DD:

1: Invoke Main Form
2: Show Main Form
3: Select Transaction Form From Main Menu
4: Show Transaction Form
5: Fill Required Fields
6: Action(action)
7: Action()
8: Required Action Performed
3.5.12 UC_RegisterUser:

1: Request Register Form
2: Show Register Form
3: Fill Required Details
4: Valid Input()
5: Register User()
6: Registration Confirmed
3.5.13  **UC_UpdateUser:**

1: Request Update Form

2: Show Update Form

3: Update Required Fields

4: Valid Updation()

5: Updation Confirmed
3.6. Sequence Diagram:

A Sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behavior of a system.

A Sequence diagram is two-dimensional in nature. On the horizontal axis, it shows the life of the object that it represents, while on the vertical axis, it shows the sequence of the creation or invocation of these objects.

The sequence diagrams of the Sales and Purchase Management System of AL-Noor Cotton Ginters and Oil Mills are shown below:

3.6.1. Use_login:
3.6.2. UC_View_Reports:
3.6.3. UC_Enter_Party:

Student Information System
1: Request Party Form
4: Fill Required Fields (Party)
5: Save Information
9: Show Message
2: Prepare Party Info
3: Show Party Form
6: Valid Input
7: Insert Party Info
8: Party Info Saved
3.6.4. UC_Enter_Product:

1: Request Product Form
2: Prepare Product Info
3: Show Product Form
4: Fill Required Fields (Product)
5: Save Information
6: Valid Input
7: Insert Product Info
8: Product Info Saved
9: Show Message
3.6.5. UC_Enter_Vehicle:

1: Request Vehicle Form

2: Prepare Vehicle Info

3: Show Vehicle Form

4: Fill Required Fields (Vehicle)

5: Save Information

6: Validate Input

7: Insert Vehicle Info

8: Vehicle Info Saved

9: Show Message
3.6.6. UC_Weight_Vehicle:

1: Request Weight Vehicle Form
2: Prepare Weight Vehicle Info
3: Show Weight Vehicle Form
4: Fill Required Fields (Weight)
5: Save Information
6: Valid Input
7: Insert Weight Vehicle Info
8: Weight Vehicle Info Saved
9: Show Message
3.6.7. UC_Decide_Rate:

![Diagram of UC_Decide_Rate](image)

3.6.8. UC_Lots:
3.6.9. UC_Sale_Voucher:

Student Information System
3.6.10. UC_Receive_DD
3.6.11. UC_Register_User:
3.6.12. UC_Update:

Student Information System
3.7. Collaboration Diagram:

A collaboration diagram describes a pattern of interaction among objects; it shows the objects participating in the interaction by their links to each other and the messages that they send to each other.

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaborations are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

The collaboration diagrams of the Sales and Purchase Management System of AL-Noor Cotton Ginners and Oil Mills are shown below:
3.7.1. UC_Login:

3.7.2. UC_Reports_View:

3.7.3. UC_Enter_Party:
Project Report

3.7.4. UC_Enter_Product:

3.7.5. UC_Enter_Vehicle:
3.7.6. UC_Weight_Vehicle:

3.7.7. UC_Decide_Rate:
3.7.8. UC_Lots:

3.7.9. UC_Sale_Voucher:
3.7.10. **UC_Receive_DD:**

3.7.11. **UC_Register_User:**
3.7.12. UC_Update:

3.8. Design Class Diagram:
Project Report

3.9. State Chart Diagram:

3.9.1. UC_Login:

```
Start → System Screen → Login → Main Form → End
```

3.9.2. UC_View_Reports:

```
Main Form → View Report → Report Form → End

End
```

3.9.3. UC_Enter_Party:

```
Main Form → Request Party Form → Party Form → End

End
```

3.9.4. UC_Enter_Product:

```
Main Form → Request Product Form → Product Form → End

End
```

3.9.5. UC_Enter_Vehicle:

```
Main Form → Request Vehicle Form → Vehicle Form → End

End
```

3.9.6. UC_Weight_Vehicle:
3.9.7. UC_Decide_Rate:

3.9.8. UC_Lots:

3.9.9. UC_Sale_Voucher:

3.9.9. UC_Receive_DD:

Chapter 4  DATABASE INTERNAL & TECHNICAL DESIGN
4.1. Introduction to Database
4.2. Database system
4.3. Advantages of Database
4.4. Benefits of Database approach
4.5. Data Independence
4.6. Data Integrity
4.7. Identification of Entities
4.8. Attribute
4.9. Normalization
4.10. Database design
4.11. Data Modeling
4.12. Snap-Shots of Table
4.13. Data Model Showing Relationships
4.1. Introduction of database:
Database is a collection of information in a structured way. We can say that it is a collection of a group of facts. Your personal address book is a database of names you like to keep track of, such as personal friends and members of your family.
A database consists of an organized collection of data for one or more uses, typically in digital form. One way of classifying databases involves the type of their contents, for example: bibliographic, document-text, statistical etc.

4.1.1. What is database?
“A database is a collection of information organized into interrelated tables of data and specifications of data objects.”
The specific features of database may include:
- Representing/capturing the information about a real-world enterprise or part of an enterprise.
- Collected and maintained to serve specific data management needs of the enterprise.
- Activities of the enterprise are supported by the database and continually update the database.

4.2. Database system:
It is a kind of electronic filing cabinet. It is also known as computerized data files. Database system is essentially a computerized record keeping system. The database itself can be regard there are basically four components of database system

4.2.1. Database Management System (DBMS):
A general purpose software system enabling:
- Creation of large disk-resident databases.
- Posing of data retrieval queries in a standard manner.
- Retrieval of query results efficiently.
- Concurrent use of the system by a large number of users in a consistent manner.
4.2.2. DBMS Approach:

DBMS:
- Separation of data and metadata
- Flexibility of changing metadata
- Program-data independence

Data access language:
- Standardized – SQL
- Ad-hoc query formulation – easy

System development:
- Less effort required
- Concentration on logical level design is enough
- Components to organize data storage
- Process queries, manage concurrent access, recovery from failures, manage access control are all available.

4.2.3 Three-Level Architecture
4.2.4. Data:

Data is information that has been organized and categorized for a predetermined purpose. The term data is often used to distinguish binary machine-readable information from textual human-readable information. For example, some applications make a distinction between data files (files that contain binary data) and text files (files that contain ASCII data).

4.2.5. Hardware:

Hardware is a machine which we use to store access manipulates and manages the data. It consists of the following two things:

1. The secondary storage volumes typically moving head magnetic tapes.
2. The processor and associated main memory that are used to support the execution of the database system software.

4.2.6. Software:

All the request from users for access to the database are handled by DBMS.

4.2.7. Users:

a) Application programmer:

An Application programmer is someone who works in many different programming languages to create the source code, which is responsible for creating small or large parts of a piece of software in concert with others. Applications programming is the meat and potatoes of programming, and requires a very creative mind, as well as one that can retain lots of information about the requirements of the software, the requirements of their teammates and the code itself.

b) End user:

The second class of user is end user. End-users use the software to assist with some task. This may be flying an aircraft managing insurance policies, writing a books etc. They want to know how the software can help them. They are not interested in computer or administration details. These were final or ultimate user of a computer system. The end user is the individual who uses the product after it has been fully developed and marketed.
c) Database administrator:
A database administrator (DBA) is a person responsible for the design, implementation, maintenance and repair of an organization's database. They are also known by the titles Database Coordinator or Database Programmer, and are closely related to the Database Analyst.

4.3. Advantages of database:
The advantages for database system over traditional paper based record keeping will perhaps be more readily apparent in these examples

4.3.1. Data Security:
Data is the most important asset. Therefore, there is a need for data security. Database management systems help to keep the data secured.

- **Compactness**
  No need for possible voluminous paper files.
- **Speed**
  Machines can retrieve and update data for faster than human can.
- **Accuracy**
  Accurate up to date information is available on demand at any time.

4.4. Benefits of database approach:
The benefits of the database approach are as follows

- Data Independence
- Consistency of Data
- Control Over Redundancy
- Integrity of Data
- Greater Security of Data
- Centralized Control of Data
- Increased Productivity
- Minimal Data Redundancy
- Data Sharing
Project Report

- Ease of application development
- Enforcement of standards
- Data can be shared
- Physical data independence
- Logical data independence

4.5. Data independence:

Data independence is the type of data transparency that matters for a centralized DBMS. It refers to the immunity of user applications to make changes in the definition and organization of data. Physical data independence deals with hiding the details of the storage structure from user applications. The application should not be involved with these issues, since there is no difference in the operation carried out against the data. The data independence and operation independence together gives the feature of data abstraction.

4.6. Data integrity:

Data integrity is data that has a complete or whole structure. All characteristics of the data including business rules, rules for how pieces of data relate dates, definitions and lineage must be correct for data to be complete. Data that has integrity is identically maintained during any operation (such as transfer, storage or retrieval). Put simply in business terms, data integrity is the assurance that data is consistent, certified and can be reconciled.

4.6.1. Entity integrity:

Entity integrity concerns the concept of a primary key. Entity integrity is an integrity rule which states that every table must have a primary key and that the column or columns chosen to be the primary key should be unique and not null.

4.6.2. Referential integrity:
Referential integrity concerns the concept of a foreign key. The referential integrity rule states that any foreign key value can only be in one of two states. The usual state of affairs is that the foreign key value refers to a primary key value of some table in the database. Occasionally, and this will depend on the rules of the business, a foreign key value can be null. In this case we are explicitly saying that either there is no relationship between the objects represented in the database or that this relationship is unknown.

4.6.3. Domain integrity:

Domain integrity specifies that all columns in relational database must be declared upon a defined domain. The primary unit of data in the relational data model is the data item. Such data items are said to be non-decomposable or atomic. A domain is a set of values of the same type. Domains are therefore pools of values from which actual values appearing in the columns of a table are drawn.

4.7. Identification of Entities:

Entity is a basic data object in database modeling. Entity can be person, a place, an event or a thing about which we have to save data in the database. If we assume that our database is a language then we can say that entities are nouns. Database is a collection of entities. The first step in database modeling is to identify entities of database. This is of the major parts in conceptual database modeling. Following are some characteristics of an entity and it is very important to consider these while identifying entities.

- Each entity should be significant.
- Each entity should be generic.
- Each entity should be fundamental.
- Each entity should be unitary.

a) Significant:
List only entities that are important to your database users and that are worth the trouble and expense of computer tabulation.

**b) Generic:**

List only types of things, not individual instances. For instance, *symphony* might be an entity, but *Beethoven's Fifth* would be an entity instance or entity occurrence.

**C) Fundamental:**

List only entities that were exist independently and do not need something else to explain them. Anything you might call a trait, a feature, or a description is not an entity. For example, a *part number* is a feature of the fundamental entity called *part*.

**d) Unitary:**

Be sure that each entity you name represents a single class. It cannot be separated into subcategories, each with its own features. In the telephone directory example, the telephone number, an apparently simple entity, actually consists of three categories, each with different features.

**e) Weak entity:**

A weak entity is an entity that exists only if is related to a set of uniquely determined entities, which are called the owners of the weak entity. For instance, we could extend our library with a weak entity type *edition*; each book has several editions, and certainly it is nonsense to speak about an edition if this does not happen in the context of a specific book. From a user interface viewpoint, weak entities are usually edited in the context of (one of) their owners. When an entity is deleted from a schema instance, all owned weak entities are deleted, too. We shall call the type of a weak entity a weak entity type.

**4.8. Attribute:**
Entity contains a set of attributes. We can call attributes as properties, features or quality of the entity. An attributes is smallest information that can’t be divided further. If we say that entity is a table then columns would be attributes.

While defining attributes, we should consider following points.

- Attributes should be significant.
- Attributes should not be derived.
- Attributes should not be decomposable.
- Data of attributes should be of same type.

4.8.1. **Primary key attributes:**

The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique (such as Social Security Number in a table with no more than one record per person) or it can be generated by the DBMS (such as a globally unique identifier, or GUID, in Microsoft SQL Server). Primary keys may consist of a single attribute or multiple attributes in combination.

Primary key attributes are given below:

- Primary key should be unique
- Primary key should not be null.
- Primary key should be not updateable.

4.9. **Normalization:**
Normalization is a process to organize the data in an efficient manner. There are two basic results which we expect from normalization. First is to remove redundant data and second is avoid duplicate date to be recorded in database.

<table>
<thead>
<tr>
<th>Std_id</th>
<th>Name</th>
<th>Address</th>
<th>Subject</th>
<th>Credit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ali</td>
<td>2-C</td>
<td>D.B</td>
<td>3.0</td>
</tr>
<tr>
<td>2</td>
<td>Rizwan</td>
<td>3-C</td>
<td>O.S</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Table 2: Maintenance detail Table
Through this example we explain different problems that might occur if the table is not normalized.

**a) Redundant Data:**
Just consider that if we want to add a new entry of Maintenance then we has to enter all the information regarding item again, in the above table, there are only a few records. Just imagine what will happen where we have to store thousands of records.

**b) Modification Anomaly:**
Now consider another situation where we have to update the record of maintenance item then we have to update it at many places, now what would happen when we will have millions of records.

**c) Deletion Anomaly:**
What if we want to delete any record of any item from above mentioned table, then we will also lose the information about slip no so what if we want to keep the record of thesis maintenance information but still want to delete some information?

**d) Insertion Anomaly:**
Suppose another situation where we have to insert a new record of thesis maintenance information but we do not want insert data about vehicle.
To avoid such situations which are described above, we have to normalize the database.
Project Report

We can divide the whole normalization process into four steps; until and unless, we are done with first step we cannot move to next step.

These four steps are given below:

- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce-Codd Normal Form

4.9.1. First Normal Form:

First normal form enforce that the value of each column in table should be atomic which means there should not be a group of data for one column. To understand the concept of first normal form just considers the following example where we want to store the record of suppliers.

<table>
<thead>
<tr>
<th>Std_id</th>
<th>Name</th>
<th>Subject</th>
<th>Credits</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ali Samad</td>
<td>D.B</td>
<td>3.0</td>
<td>2-C</td>
</tr>
<tr>
<td>2</td>
<td>Rizwan Qureshi</td>
<td>O.S</td>
<td>3.0</td>
<td>3-C</td>
</tr>
</tbody>
</table>

Table2: Maintenance detail Table

In This Maintenance Detail table Receipt _no will be a primary key.

Hence we can say, in first normalization form, we have to do following things:

- Eliminate Redundant Data.
- Declare Primary Key.
4.9.2. **Second Normal Form:**

First of all, to implement second normal form, we have to implement first normal form. First normal form requires maintaining the atomicity of data and second normal form requires relationship between the key and non key attributes.

According to second normal form, all non key attributes must be dependent on key attribute. If primary key is composite then non key attribute must depend on all the key columns.

Understand the concept of second normal form, consider the following example.

<table>
<thead>
<tr>
<th>Std_id</th>
<th>Name</th>
<th>Dept</th>
<th>Session</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ali Samad</td>
<td>C.S</td>
<td>90</td>
<td>Fresh</td>
</tr>
<tr>
<td>2</td>
<td>Rizwan Qureshi</td>
<td>Math</td>
<td>95</td>
<td>Fresh</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Std_id</th>
<th>Course</th>
<th>Date Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C++</td>
<td>X/Y/Z</td>
</tr>
<tr>
<td>2</td>
<td>Java</td>
<td>X/Y/Z</td>
</tr>
</tbody>
</table>
4.9.3. **Third Normal Form:**

Third normal form requires that all the non key attributes should complete depend on primary key attribute which mean that there should be any transitive dependency in table attributes.

Transitive dependency means that any non key attribute is depending on any other non key attribute which is depending on key attribute.

There is a general rule to find whether your table is in third normal form or not. You have to identity column which need upgrading when you upgrade any other column in that table.

<table>
<thead>
<tr>
<th>Std_id</th>
<th>Name</th>
<th>Major</th>
<th>Advisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rizwan</td>
<td>Math</td>
<td>ALI</td>
</tr>
<tr>
<td>2</td>
<td>Faheem</td>
<td>English</td>
<td>JAWAAD</td>
</tr>
</tbody>
</table>

Some time, it can make things too complex while implementing these normal form so best way is to find a balance as when it gets too complex then many DBMS requires more resource or it can decrease the performance.

4.10. **Database design:**

A carefully thought-out database design forms the foundation for future success. These links will help you plan your database designs to maintain performance and integrity through future growth. Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

4.11. **Data Modeling:**
Data modeling in software engineering is the process of creating a data model by applying formal data model descriptions using data modeling techniques. The data requirements are recorded as a conceptual data model with associated data definitions. Actual implementation of the conceptual model is called a logical data model.

After creating conceptual database design, you have to represent that by using any modeling techniques. Currently, there are many modeling tools and techniques are available which are given below:

- Unified Modeling Language (UML)
- Entity Relationship Diagram (ERD)
- Relation Model
- Relational Algebra

Here in this document, we are going to discuss entity relationship diagram and UML as this is the most widely used technique in the world.

a) Entity Relationship Diagram (ERD)

An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. Entity Relationship diagram can also help developers in initial phases to create better understanding of user’s requirements. Now we are going to explain the basic principles to develop entity relationship diagram.

- Identify Entities
- Define relationships
  - Define Cardinality of relationships
  - Identify Attributes and Primary Keys
  - Map all Attribute

a) Basic Objects:
c) Developing ERD:

- Use rectangle to represent an entity
- Use Diamond to draw relationship between two entities
- Use Ellipse to represent attributes
- Use to show the linkage
- Use to represent for weak entity
- Use to show multi-valued attributes
- Use to represent foreign key attributes
- Use to mention primary key attribute
To understand the core concept of ERD, consider the Book Wholesale System. There are following steps for Developing ERD.

**Step 1- Identify Entities:**

The first step of developing ERD is identifying entities. We can identify following entities.

- Admin (Login)
- Teacher
- Student
- Class
- Subject
- Fees
- Time Table
- Datesheet
- Accounttbl
- Bank Account

**Step-2 Identify Relationship:**

Second step of Developing ERD is to identify relationship between listed entities. First of All, we are going to examine which entities have relation between them. So from problem statement, we can include that physician can given Prescription to patient and Prescription has medicines for illness. Physician, Patient, medicine has relationship with prescription and illness has relationship with the medicine as well as with patients.

**Step -3 Identify Cardinality:**

Now we have to find out the Cardinality for each relationship.
These relationships are:

- One to
- Many and cardinality will be 1: M.

**Step-4 Identify Attributes:**

The following attributes were identified

**Attributes of User:**

- **Userld**
- **Login**
- **Password**

<table>
<thead>
<tr>
<th>User(Login)1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
</tr>
<tr>
<td>Login</td>
</tr>
</tbody>
</table>

**Attributes of Vehicle:**

- **VehicleId**
- **VehicleName**
- **RegistrationNo**
- **ProductId**
- **Date**
- **Time**
- **VehicleStatus**

<table>
<thead>
<tr>
<th>Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
</tr>
<tr>
<td>VehicleName</td>
</tr>
<tr>
<td>PartId</td>
</tr>
<tr>
<td>Date</td>
</tr>
<tr>
<td>VehicleStatus</td>
</tr>
</tbody>
</table>

**Attributes of Party:**
Attributes of Product:

Attributes of Weight:

Attributes of Lots:
Attributes of Sale:

Attributes of SaleVoucher:

Attributes of AccountHead:
4.12. Snap-Shots of Tables:
4.12.1. Party Table

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PartyId</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>PartyName</td>
<td>varchar(50)</td>
<td></td>
</tr>
<tr>
<td>CNICNo</td>
<td>nchar(15)</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>varchar(50)</td>
<td></td>
</tr>
<tr>
<td>ContactNo</td>
<td>varchar(50)</td>
<td></td>
</tr>
<tr>
<td>NTNNo</td>
<td>varchar(50)</td>
<td></td>
</tr>
</tbody>
</table>

4.12.2. Product Table:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductId</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>ProductName</td>
<td>varchar(50)</td>
<td></td>
</tr>
</tbody>
</table>

4.12.3. Vehicle Table:

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>VehicleId</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>VehicleName</td>
<td>varchar(50)</td>
<td></td>
</tr>
<tr>
<td>RegistrationNo</td>
<td>varchar(50)</td>
<td></td>
</tr>
<tr>
<td>PartyId</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>ProductId</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td>VehicleStatus</td>
<td>varchar(50)</td>
<td></td>
</tr>
</tbody>
</table>

4.12.4. Weight Table:
4.12.5. Lots Table:

### Table - dbo.Lots

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>LotNo</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>LotWeight</td>
<td>numeric(18, 0)</td>
<td></td>
</tr>
<tr>
<td>NoOfBales</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td>Moisture</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>Remarks</td>
<td>varchar(50)</td>
<td></td>
</tr>
</tbody>
</table>

4.12.6. Sale Table:

### Table - dbo.Sale

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaleId</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>VoucherNo</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>productId</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>LotNo</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>NoOfBales</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>TotalWeight</td>
<td>numeric(18, 0)</td>
<td></td>
</tr>
</tbody>
</table>

4.12.7. BankAccount Table:
4.12.8. Transactiontbl Table:

4.12.9. SaleVoucher Table:

4.12.10. AccountHead Table:

4.13. Data Model Showing Relationships:
Chapter 5  TESTING
5.1. Testing

5.2. Testing Strategy

5.3. Test Cases

5.1. Testing:
Project Report

It is the process used to help identify the correctness, completeness, security, and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate.

5.2. Testing Strategy:

Software testing methods are traditionally divided into:

1. White box testing.
2. Black box testing.

This Strategy used for testing is Black Box Testing. Every module is tested and after the integration of all modules again individually testing is done.

5.3. Test Cases

5.3.1. Test Case: 1

System:

Sales and Purchase Management System of AL-Noor Industries

Test:

Login form is opened and login and password is entered.

Instructions:

1. Open the main form
2. Enter user name and password.

Expected Result:

Student Information System
Message will appear that please enter correct user name and password.

**Actual Result:**

Error occur with the following message “Please enter correct username and password

5.2.2. Test Case: 2

**System:**

Sales and Purchase Management System of AL-Noor Industries

**Test:**

Main form is opened and username and password is entering in a Main Menu.

**Instructions:**

1. Open the Main Form
2. Select Main Menu to enter in a Party Form.

**Expected Result:**

Main Information form will be opened.

**Actual Result:**

Result was as per expected.

5.2.3. Test Case: 3

**System:**

Student Information System
Test:

Choose Party Form from Main Menu and Enter Party information and select on save button then Exit button.

**Instructions:**

1. Open Party form.
2. Enter Owner record.
3. Click the button Save and Exit.

**Expected Result:**

Database will be updated with new Party record.

**Actual Result:**

Result was as per expected.

5.2.4. Test Case: 4

**System:**

Sales and Purchase Management System of AL-Noor Industries

Test:

Delete Product Entry form the Product Form in Main Form.

**Instructions:**

1. Select required record form the list.
2. Click Delete button.
Expected Result:

Database will be updated and record will be deleted.

Actual Result:

Result was as per expected.

5.2.5. Test Case: 5

System:

Sales and Purchase Management System of AL-Noor Industries

Test:

Update Vehicle Form from the Main Form.

Instructions:

1. Open Vehicle From.
2. Clicks Update Button.

Expected Result:

Database will be updated and record will be deleted.

Actual Result:

Result was as per expected

Chapter 6 USER INTERFACE AND DESIGN
6.1. Site Map

6.2. Gate Clerk

6.3. Computer Weight Clerk

6.4. Managing Partner

6.5. Bale Weight Clerk:

6.6. Accountant

6.7. Reports

6.1. Site Map

1. Login Form

2. Main Form
3. Relevant Form according to the Project

6.2. Gate Clerk:

1. Login Form
2. Main Form
3. Enter Party, Product, Vehicle information by opening these forms
4. Issue Gate Pass
PARTY INFORMATION

Party Name

CNIC No.

Address

Contact No

N-T-N No.

<table>
<thead>
<tr>
<th>Party Id.</th>
<th>Party Name</th>
<th>Address</th>
<th>Contact No</th>
<th>CNIC</th>
<th>NTN NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Ali Khan</td>
<td>31201-1234567-1</td>
<td>Ahmed Pur East</td>
<td>(0333)-7665687</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Muhammad Iqbal</td>
<td>31201-2345688-8</td>
<td>One Unit Colony</td>
<td>(0333)-5391512</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Malik Allah Baig</td>
<td>31202-4568989-2</td>
<td>Uch Sharif</td>
<td>(0623)-5645621</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mohd. Ali</td>
<td>32024-2555564-2</td>
<td>Moza Aarbi Uch</td>
<td>(0321)-4561223</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Ch. Mukhtar Ali</td>
<td>32540-0215456-2</td>
<td>Rao Colony Ahmm...</td>
<td>(0253)-2552233</td>
<td></td>
</tr>
</tbody>
</table>
VEHICLE ENTRY

Vehicle Name: [ ]
Registration No: [ ]
Date: Saturday, November 13, 2010
Time: Saturday, November 13, 2010

Vehicle Status: [ ]

<table>
<thead>
<tr>
<th>Vehicle Id.</th>
<th>Vehicle Name</th>
<th>Registration No</th>
<th>Date</th>
<th>Time</th>
<th>Vehicle Status</th>
<th>Product Id.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Truck</td>
<td>b123</td>
<td>10/31/2010 6:27...</td>
<td>1/1/1900 12:30...</td>
<td>empty</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Tractor</td>
<td>MNH-145</td>
<td>11/1/2010 12:00...</td>
<td>11/1/2010 11:44...</td>
<td>Full</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Tractor</td>
<td>BR-2726</td>
<td>11/3/2010 12:00...</td>
<td>11/3/2010 9:11...</td>
<td>Full</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Tractor</td>
<td>BRN-2456</td>
<td>11/3/2010 12:00...</td>
<td>11/3/2010 9:19...</td>
<td>Full</td>
<td>1</td>
</tr>
</tbody>
</table>
### Gate Pass

<table>
<thead>
<tr>
<th>PartyName</th>
<th>CNICNo</th>
<th>Address</th>
<th>ContactNo</th>
<th>Vehc</th>
<th>VehicleName</th>
<th>Registration</th>
<th>Product/Name</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
</table>
6.3. Computer Weight Clerk:

1. Login Form
2. Main Form
3. Enter Weight Information by opening Main Form
4. Issue Weight Slip
6.4. Managing Partner:
   1. Login Form
   2. Main Form
   3. Enter Decided Rate by opening Purchase Rate Form

6.5. Bale Weight Clerk:
   1. Login Form
   2. Main Form
   3. Enter Lots Information by opening Lots Form
6.6. Accountant:

1. Login Form
2. Main Form
3. Enter Account heads and Banking Information by opening Account Head and Bank Account Form
4. Enter Amount as Debit or Credit by opening Transaction Form
Account No. 
Account Holder 
Account Type 
Bank 

<table>
<thead>
<tr>
<th>Account Id</th>
<th>Account No.</th>
<th>Account Holder</th>
<th>Account Type</th>
<th>Bank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12365-AF</td>
<td>Ali</td>
<td>saving</td>
<td>bank of punjab</td>
</tr>
<tr>
<td>2</td>
<td>3322665-223</td>
<td>Umer Shafi</td>
<td>Current</td>
<td>United Bank Ltd.</td>
</tr>
</tbody>
</table>
Project Report

![Image of a student information system interface]

The interface includes fields for Amount, Account Head Id, Bank Account No, Description, and Date. There is also a table showing transactions with columns for Transaction Id, Debit Amount, Credit Amount, Description, and Date.
6.7. Reports:

6.7.1 Party Information:
6.7.2 Products:

<table>
<thead>
<tr>
<th>ProductId</th>
<th>ProductName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Raw Cotton</td>
</tr>
<tr>
<td>2.00</td>
<td>Cotton Seeds</td>
</tr>
<tr>
<td>3.00</td>
<td>Cotton Lint</td>
</tr>
<tr>
<td>4.00</td>
<td>Oil</td>
</tr>
<tr>
<td>5.00</td>
<td>Oil Cake</td>
</tr>
<tr>
<td>6.00</td>
<td>Wastage</td>
</tr>
</tbody>
</table>
6.7.3. Party and Vehicle According Name:

![Image of Party and Vehicle Form]

- **PartyName**: Ali Khan
- **Vehicle Name**: Truck
- **Vehicle Status**: Empty
- **Date**: 10/31/2010
- **Time**: 12:00 AM
- **Registration**: 123
REFERENCES

Website References:

www.google.com

www.mamma.com

www.yahoo.com

www.bing.com

Book References:

• Visual Basic Black Book

• UML in 24 Hours

• Visual Basic.Net