Chapter 5

5. System Analysis and Design

5.1 Introduction

Using of correct and clear System Analysis and Design methodology helps to close the communication gaps between business people and IS developers, as well as between developers and developers. This not only helps us ensure the system is built right but also that the right system is built for the business.

In this chapter, all the information that needs to understand the current system has been discussed. This chapter gives a clear picture of what actually the physical system is?

Identification the scope of the system, identification of user's requirement, recommendations to overcome the limitations and problems of the present system and the feasibility of the proposed Business Solution Options (BSOs) are the main focused areas of this chapter.

In addition to that the architectural design of the proposed system and in detail specifications of the proposed system also have been discussed including context diagrams, DFDs, LDS, Relation ship diagram etc of the proposed system.

In the analysis phase below questions were answered

- Who will use the system?
- What the system will do?
- Where and when will it be used?

In the design phase it was decided how the system will operate, in terms of the hardware, software, and network infrastructure; the user interface, forms, and reports that will be used; and the specific programs, databases, and files that will be needed.

5.2 System Investigation.

Structured System Analysis and Designing Methodologies (SSADM) is used as the system analysis and designing methodology.

5.2.1 Requirement Gathering.

Interviewing is a common technique used to collect information from key stakeholders in a software project. Since this method is having following advantages it same was used for requirement gathering

- Generally easy, because it can be done with minimal preparation.
- Interviews of individuals and small groups require less planning and scheduling effort than large workshops.
- Interviews of individuals and small groups require less stakeholder commitment than large workshops.
- Interviews provide an opportunity to explore or clarify topics in more detail.

Initially, few meetings had with production manager and collected all the necessary information about the requirement. All the formulas and methodology for calculation of productivity was given by him. After identifying his requirement, then separate set of meetings had with the managers of other relevant departments. Their requirements also were identified. All the information with the copies of existing report and documents also were collected. Based on the collected data and information, functional and non-functional requirement were identified.

5.2.2 Functional and Non Functional Requirements

5.2.2.1 Functional Requirements

Functional requirements explain what the system should do, and identify the necessary task, action or activity that must be accomplished.

Identified functional requirements are listed below and the system must provide theses facilities:

- System shall provide facility to calculate the productivity ratio line wise weekly basis.

- System shall provide facility to calculate the average productivity ratio line wise for a given week range (Monthly).
- System shall provide facility to input adjustment parameter line wise to adjust the productivity value if necessary.
- System shall provide facility to input the “setup factor” parameter line wise to adjust the productivity value if necessary.
- Standard timing data for transformers should be able to add/modify/print.
- Best operation should be able to mark and should be able to identify separately in the timing sheet.
- Standard timing sheet of a transformer should be able to copy for another transformer
- Standard timing sheet of a transformer should be able to copy for as for a job timing sheet.
- Facility should be available to copy standard timing sheet for a given job range at once as a bulk. Option should be available to filter based on job start date too
- System shall provide facility to add/modify job operations
    - New operations should be able to add for the job timing sheet
    - Timing of job operations should be able to change
    - Job operations should be able to remove if there is no any transactions for that operation
- The system shall provide facility to assign different machines to different production lines.
- System shall provide facility to add/modify weekly labour hours week wise
- System shall provide facility to query daily transaction data on Job wise.
- System shall provide facility to generate validation reports for completed quantities operations based on operation order.
- System should validate the production quantity at the time of entering the data with the job quantity based on the line setup parameters.
- System should automatically picks up the correct operation timing for the operation based on the available timing and the predefined priority criteria.
- System should have the facility to graphically analyse trends of the productivity variations of lines.

5.2.2.2 Non Functional Requirements

A non-functional requirement is a statement of how a system must behave; it is a constraint upon the systems behaviour. The quality of an application depends on more than how well it satisfies user-functional requirements. Even an application that successfully makes it through development and deployment can encounter grumblings from users., if it is hard to use, keeps failing, is difficult to diagnose, or consumes excessive resources. In addition to user-functional requirements, you must also consider how well the application satisfies the non-functional requirements and fits into the organization's operational environment

Identified Non-functional requirement are listed below.

- Daily production output data sheets should be sent to data entry operators on time.
- Database server should be kept on active always.
- Database server should be capable enough to handle huge data volume
- Uninterrupted power system should be available to supply power for the servers.
- A reliable networking facility should be available as many users work in the system simultaneously online.
- 6 to 10 data entry operators should be available for huge data entering.
- Proper procedure has to be implemented in order to collect the data sheets from production lines
- HR department should provide line wise working hours weekly on time.

5.2.3 Software Requirement Specification for the Proposed System

After identifying all the functional and non-functional requirements, a feasibility study was done and documented. Then Software Requirement Specification (SRS) was prepared.

SRS document was distributed to all the stakeholders for the acceptance. Then some additional requirements were proposed by some stakeholders. Another meeting had with all the stakeholders and acceptance was received form all the users and finalized the SRS.

5.3 Business Activity Model of the Existing System

The objective of Business Activity Modelling is to find the relevant business activities in order to isolate the ones that depend on an interchange of information in the form of data items. Having isolated the subset of data dependent activities, it is easy to see the shape of the potential computer information system.

Business Activity Modelling plays an important part in our communication with users because it uses their own language to help them see which business activities will remain unaffected by the future computerised information system. Figure 5.1 shows the Business Activity model of the system which was used to identify activities and to define the system boundary.

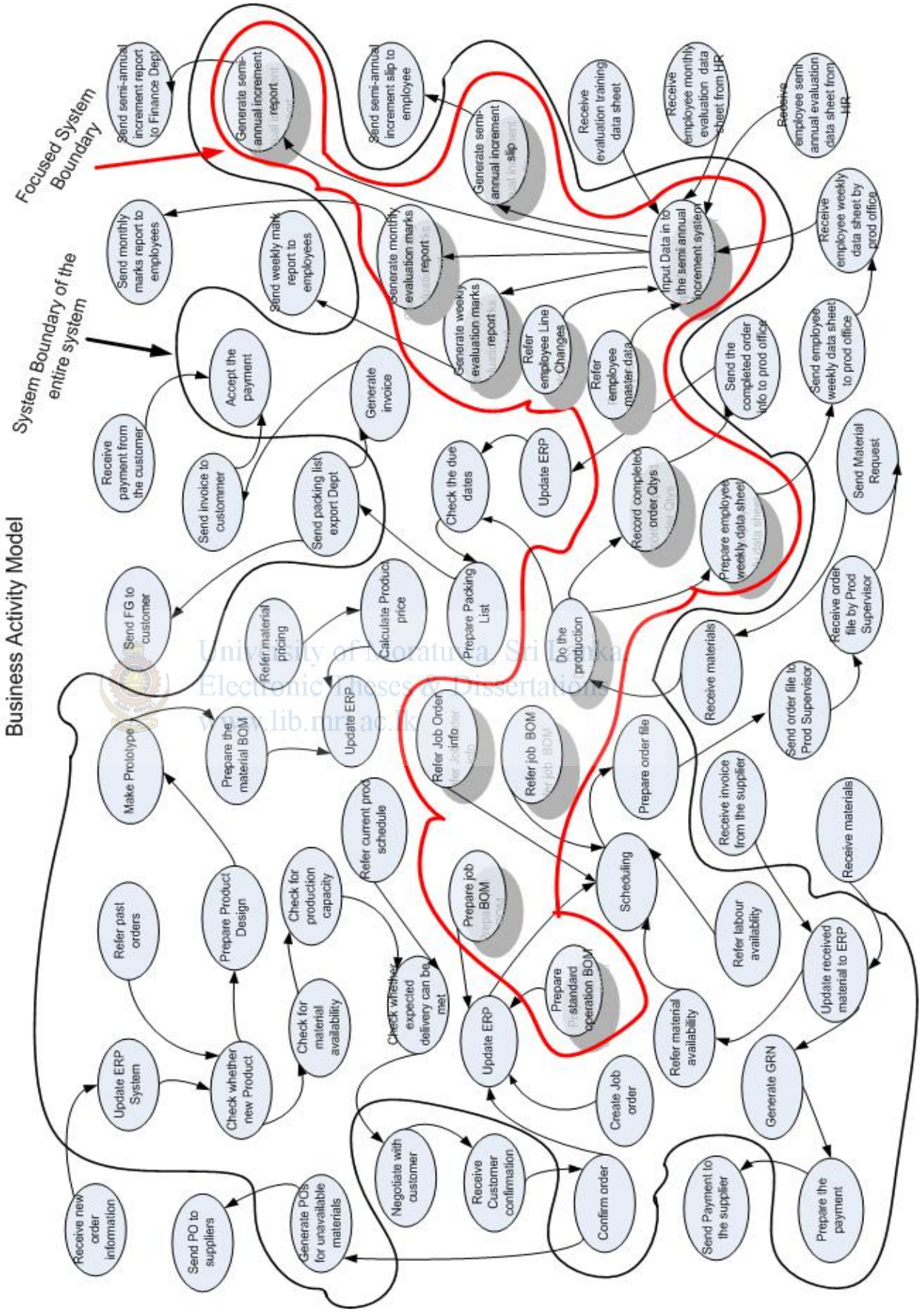

Figure 5.1 : Business Activity Model of the Existing System

5.4 Data Flow Diagrams

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

Dataflow diagrams were used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system. These DFDs helped to explain to users how the system is developed.

The old system's dataflow diagrams was drawn up and compared with the new system's dataflow diagrams to draw comparisons to implement a more efficient system.

5.4.1 Context Diagram of the Existing System

System Context Diagrams are diagrams which are used in systems design to represent all external entities that may interact with a system. This diagram is the highest level view of a system and shows the system at the center, with no details of its interior structure, surrounding by all its interacting systems, environment and activities with its inputs and outputs from/to external factors/entities. The objective of a system context diagram is to focus attention on external factors and events that should be considered in developing a complete set of system requirements and constraints.

Figure 5.2 shows existing system as a whole with all the external entities and their interactions and its inputs and outputs. This was used early in a project to get agreement on the scope under investigation.

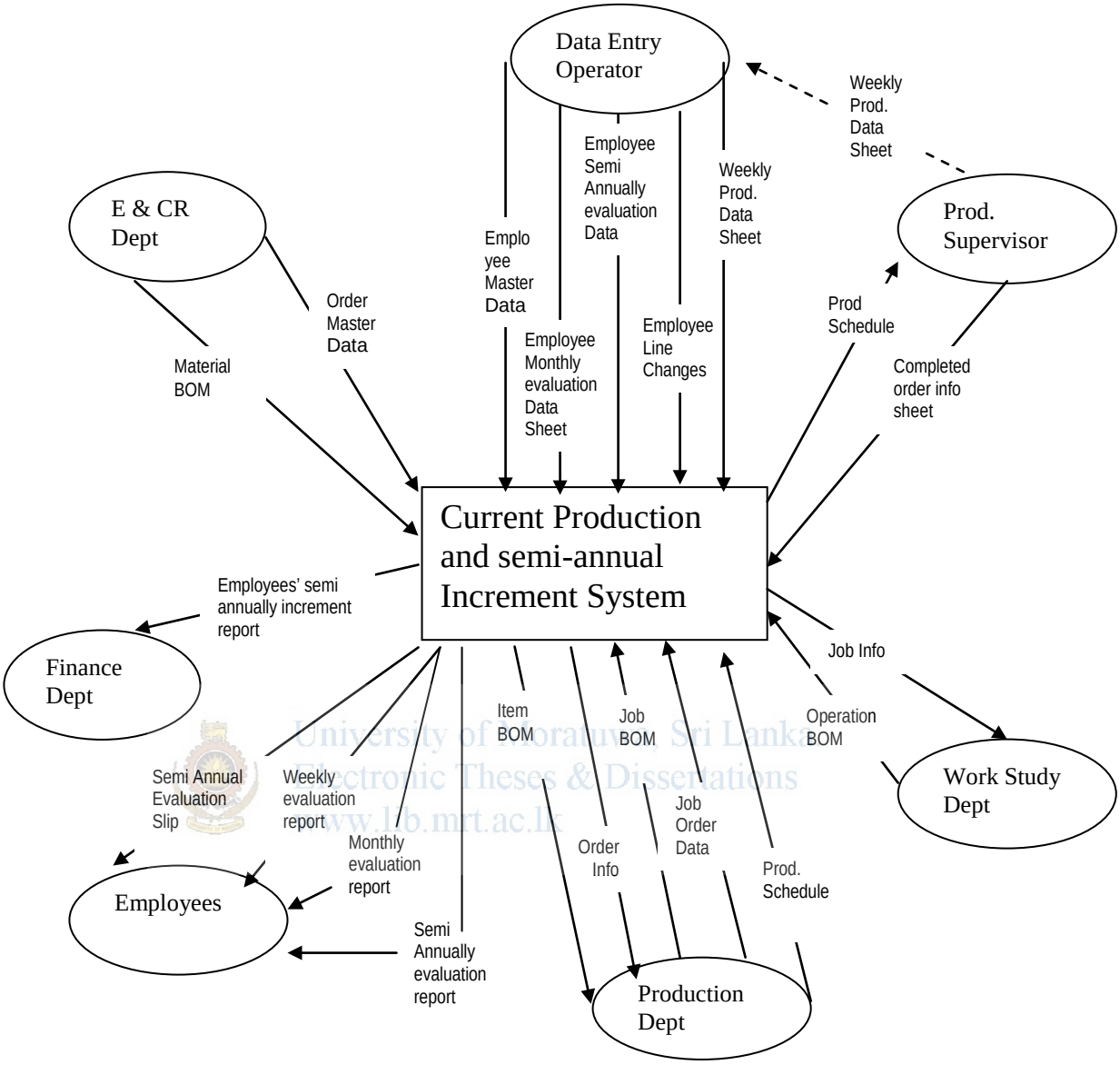

Figure 5.2 : Context Diagram of the Existing System

5.4.2 Level 1 DFD of the Existing System.

The level 1 DFD identifies the major business processes at a high level and any of these processes can then be analyzed further - giving rise to a corresponding level 2 business process diagram. This process of more detailed analysis can then continue – through level 3, 4 and so on.

Figure 5.3 shows the level 1 DFD of the existing system and all major business process could be communicated to others clearly and easily.

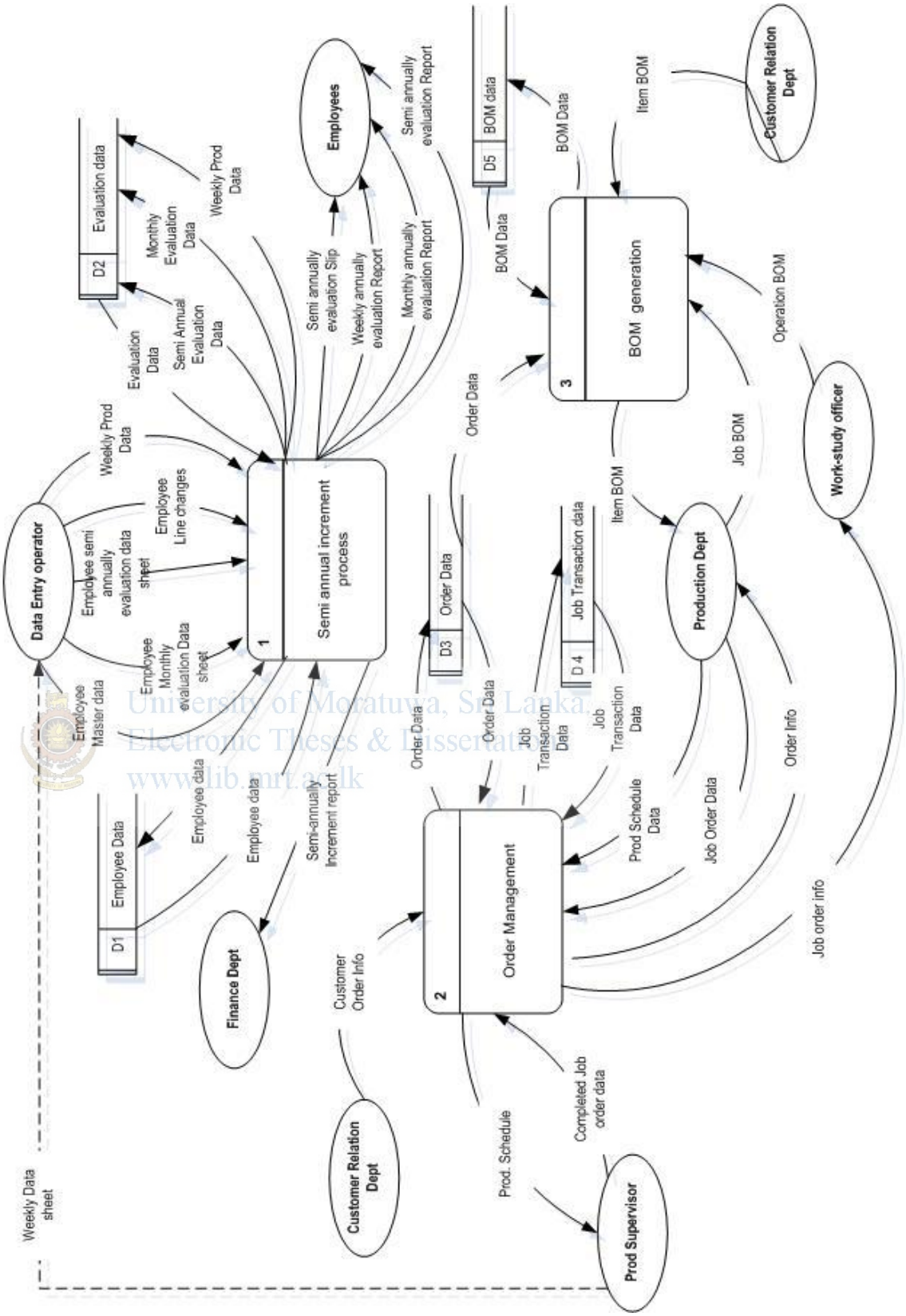

Figure 5.3 : Level 1 DFD of the Existing System

Further decomposed DFDs of Level I DFD of the existing system are available in *Appendix D.*

5.5 Business System Options (BSOs)

Once the current system was documented and the requirements of the new system were understood, the different ways (BSOs) that the requirements can be met were investigated. Two main BSOs were identified and one was based on manual data entering system and second one was for using wireless devises to collect the daily job transaction.

BSO 1

Entering of huge data volume was a one of the key areas that was considered. The data volume that was needed to be managed was very much high. Under this BSO option, manual data entering has been considered with eight data entry operators and eight additional client personal computes. Further additional network printer was needed to connect to the existing local area network other than the available network computer facilities. This BSO has been illustrated in Figure 5.4.

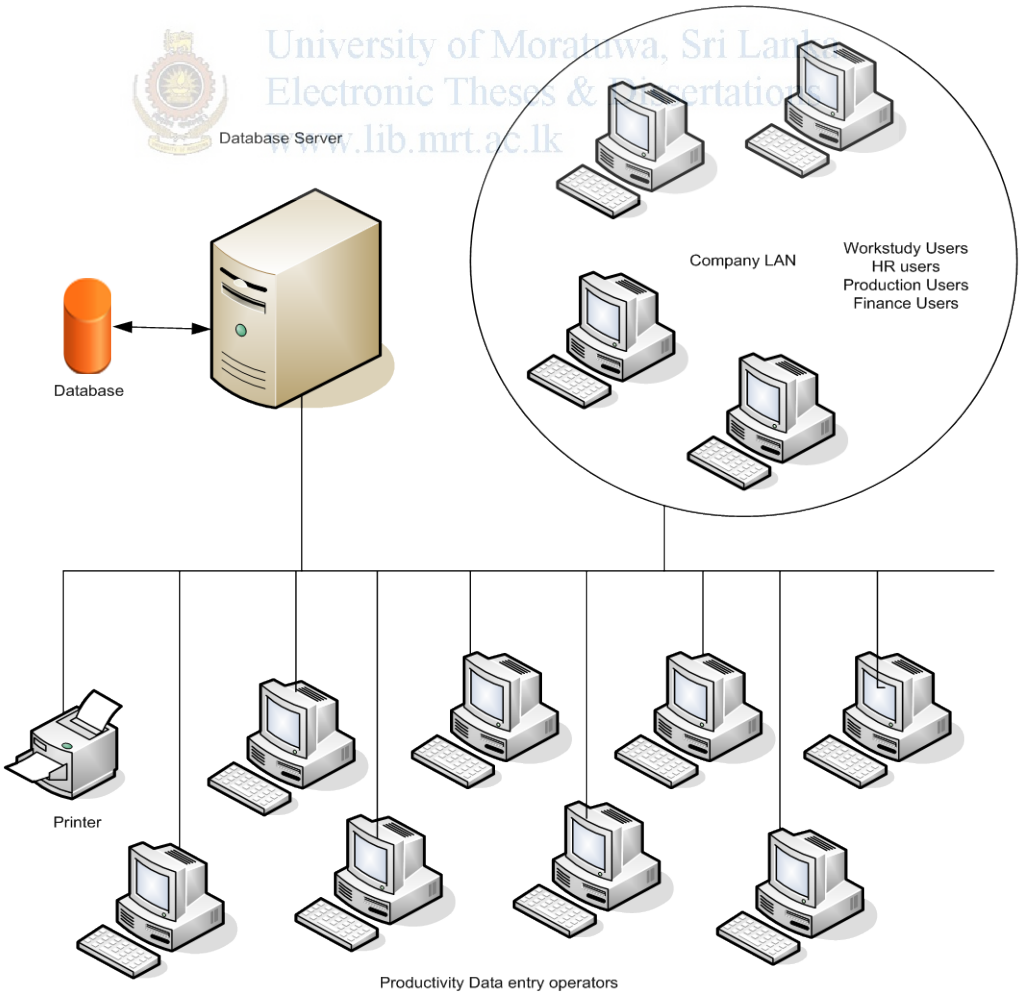

Figure 5.4 : Visual Representation of BSO I

Software and hardware requirement for BSO I

Minimum hardware and software requirements that are needed to install and run the BSO I is listed below.

Server requirement

Hardware

1 GHz processor

RAM 1 GB or more

Free hard disk space 10 GB or above

VGA colour monitor

Software

Microsoft Windows Server 2000 or above with the latest patch or

Windows NT with latest patch

Database requirement

In addition to the operating system of the server, we need to have a database installed on the server. The following databases are supported:

- Microsoft SQL Server 2000 Standard Edition or SQL Server 2005 Standard Edition, with the latest service pack
- Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2005 Enterprise Edition, with the latest service pack

Client machine requirement

Eight PCs needed with following configurations

Hardware

1 GHz or above processor

RAM 256 MB or more

Free hard disk space 500 MB or above

VGA colour monitor

Software

Windows Vista or

Microsoft Windows XP Professional SP2 or

Microsoft Windows 2000

Printer

Laser printer

Network connection

Local Area Network

Speed 100 Mbps or above

Development tools

Coding language

Visual Studio 6 (VB 6) or above

Report development tool

Crystal Report 8 or above

BSO 11

Under BSO II the possibility of automation of capturing data using wireless hand held devices were considered. In this scenario, only one additional user was considered with one additional client computer. Other than that one printer was needed to be added to the existing network. Further, wireless access facility to the local area network was needed to be activated and six wireless hand held computers (scanning bar-coding units) were considered. Three barcode printers also were needed for barcode printing. Figure 5.5 shows the architecture of the BSO II.

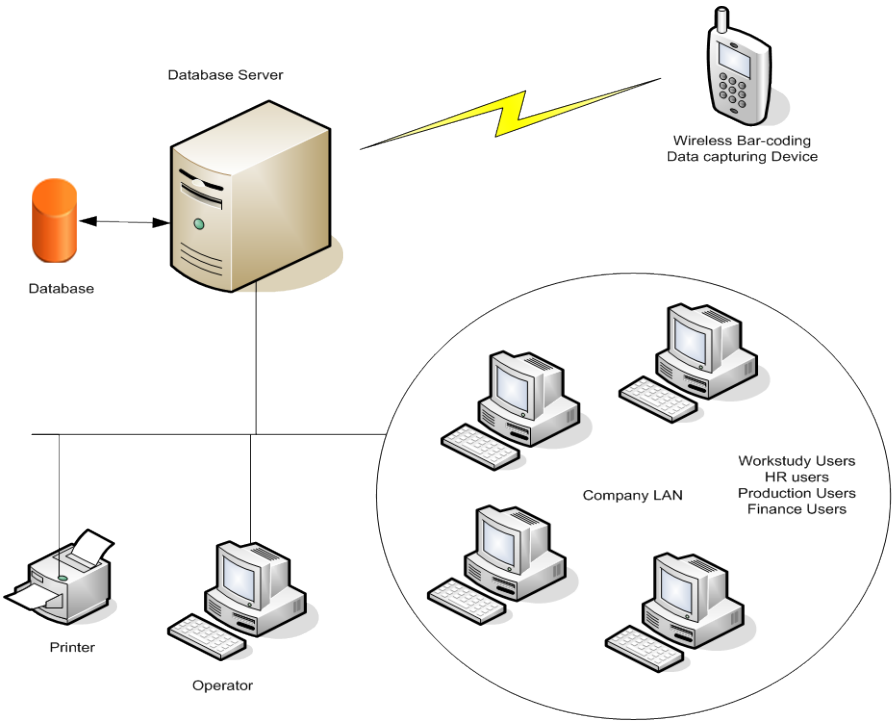

Figure 5.5 : Visual Representation of BSO II

Software and hardware requirement for BSO II

Minimum hard ware and software requirements that need to install and run the BSO II are listed below.

Server requirement

Hardware

1 GHz processor

RAM 1 GB or more

Free hard disk space 10 GB or above

VGA colour monitor

Software

Microsoft Windows Server 2000 or above with the latest patch or

Windows NT with latest patch

Database requirement

In addition to the operating system of the server, we need to have a database installed on the server. The following databases are supported:

- Microsoft SQL Server 2000 Standard Edition or SQL Server 2005 Standard Edition, with the latest service pack
- Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2005 Enterprise Edition, with the latest service pack

Client machine requirement

One client computer is needed with following configurations

Hardware

1 GHz or above processor

RAM 256 MB or more

Free hard disk space 500 MB or above

VGA colour monitor

Software

Windows Vista or

Microsoft Windows XP Professional SP2 or

Microsoft Windows 2000

Printer

One Laser printer

Three printers for barcode printing

Wireless handheld computers (barcode scanning units)

Six units are needed

Network connection

Local Area Network with wireless access facility

Speed 100 Mbps or above

Development tools

Coding language

Visual Studio 6 (VB 6) or above

Report development tool

Crystal Report 8 or above

5.6 Feasibility Study

Feasibility study was done for the identified two BSOs in order to decide whether or not it is worth carrying on with system development process.

Since the company is already having LAN including database servers, application servers and around 200 PCs including necessary software that needs to develop the application and to store the data, environment was feasible enough for the implementation of the target MIS. Feasibility was evaluated under three key areas.

1. Technical Feasibility.
2. Operational Feasibility.
3. Economic Feasibility.

5.6.1 Technical Feasibility.

Company is already having the necessary technology & Skills that needs to implement the proposed system. Local Area Network (LAN) and MS Sql Database server are already available. Since the speed of LAN is 1000 Mbps and all the workstations in the net works are having Windows XP operating system with memory more than 512 Mb, speed more than 2 GHz, above two BSOs can be installed in the existing network without any issue.

5.6.2 Operational Feasibility.

Since all the users are currently using computers, E-mail facilities, Internet facilities, there is no need to train the users specially to operate this system. Further the company is having an Enterprise Recourse Planning (ERP) system and all the activities are integrated through this system, the LAN has been established in a way that the ERP can be operated smoothly and efficiently. It implies that the down time of the network is very less and well established with a high speed network.

Since, BSO I and BSO II is also being gone to install on this network, the delivery of information at the right place on right time can be assured. In order to implement the bar-coding system in wireless network environment, new bar-coding method has to be introduced, enabling to capture production output of each operation through the handheld wireless devices. Barcode has to be introduced including the data like Job no, operation, machine etc. But in the investigation stage, it was found that introducing a bar-coding in the existing production setup is difficult because of the complexity of the operations of the production.

So, when existing complex production environment is considered, BSO II is not operationally feasible because of inability of using of barcodes.

5.6.3 Economic Feasibility

For the BSO I, only additional data entry operators have to be recruited and PCs for them have to be purchased with an additional network printer. For the BSO II, six wire less bar-coding units have to be purchased. Further barcode printers have to be purchased and barcode tags have to be printed. Cost wise both BSOs are within the approved budget limit and hence both BSOs are economically feasible.

5.7 Selected BSO and Justification

Both BSOs satisfied all the functional and Non-functional requirements. Comparison report is available in *Appendix E*. When huge data volume that has to be captured to the system is considered, it is very easy if BSO II can be implemented because all the data can be collected through handheld bar-coding scanners. Time consumption for data inputting is very less and accuracy is also very much high. But although, BSO I is Technically, Operationally, Economically feasible, BSO II is not Operationally feasible because it is very mush difficult use barcodes when considered the way of

production at each stages of operations. Further BSO I is more cost effective than the BSO II. Hence BSO I was selected as the best and feasible solution.

5.8 DFDs of the Proposed System

5.8.1 Context Diagrams of the Proposed System

Since, System Context Diagrams are diagrams which are used in systems design to represent all external entities that may interact with a system, the same was drawn for the proposed system too and shown in Figure 5.6. This diagram shows the highest level view of a system proposed system and shows the system at the center, surrounding by all its interacting systems, environment and activities with its inputs and outputs from/to external factors/entities. The objective of a system context diagram was to focus attention on external factors and events that should be considered in developing a complete set of system requirements and constraints.

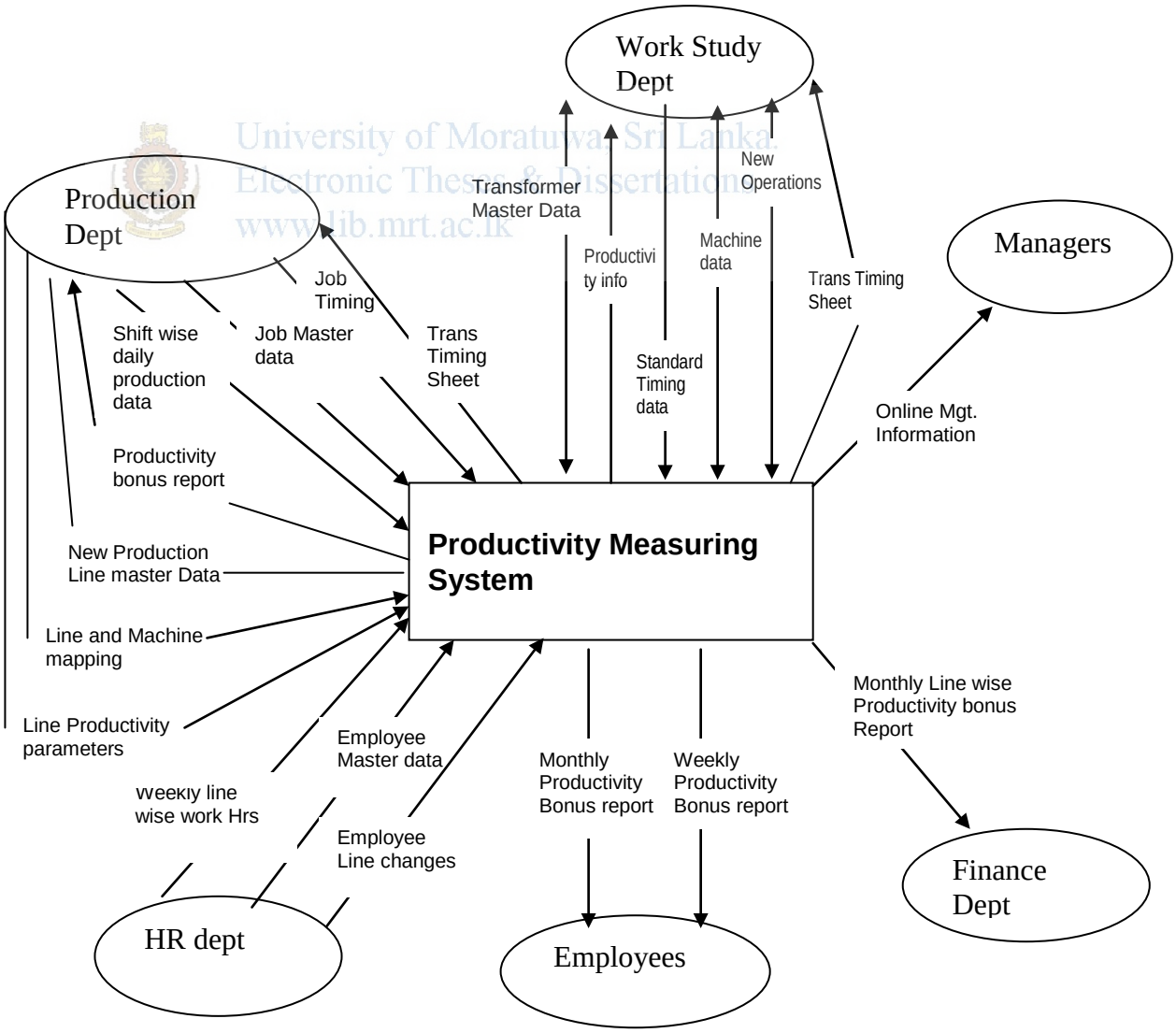

Figure 5.6 : Context Diagram of the Proposed System

5.8.2 Level 1 Data Flow Diagrams of the Proposed System

Level 1 DFD was drawn in order to identify the major business processes at a high level and in order to further analyze these processes - giving rise to a corresponding level 2 business process diagram. Level 1 DFD of the proposed system is shown in the Figure 5.7. Then level 2 DFDs were drawn by further decomposing the higher level business processes which were shown in level 1 DFD. All the level 2 DFDs are available in the *Appendix F*.

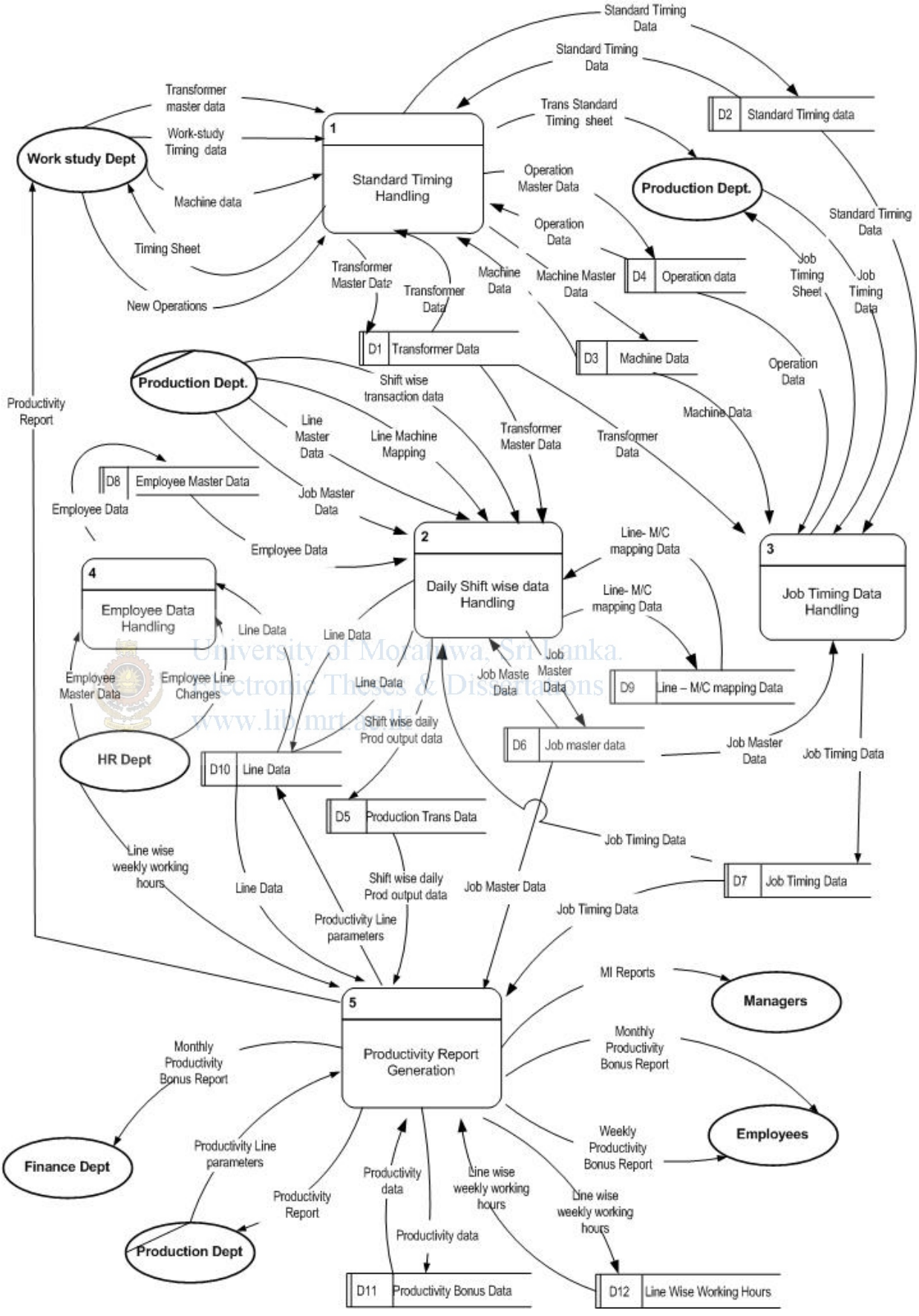

Figure 5.7 : Level 1 DFD of the Proposed System

5.9 Architectural Design of the System

System decomposition into subsystems or architectural components is very much important because a good decomposition satisfies the principle of loose coupling between the pieces, facilitated by clean interfaces, simplifying the problem by dividing it into reasonably independent pieces that can be tackled separately.

In order to reduce the complexity and to support for early delivery of the system, complete system was decomposed in to three major modules as in following Figure 5.8. Decomposing was done such a way that the decomposed structure supports the functionality or services required of the system and maintains the system integrity.

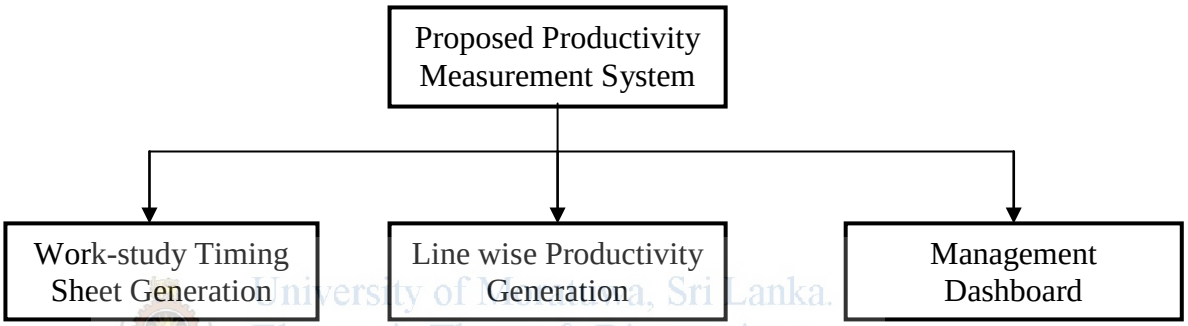

Figure 5.8 : High level Architectural Design of the Proposed System

Based on the architectural design of the system, system development was done under three phases. As the first phase, work-study timing sheet generation was completed and implemented. Then productivity calculation module was developed and implemented. Final implementation was the management dash board for analyzing and monitoring the productivity.

RAD (Rapid Application Development) methodology which gives faster development and higher quality was used as the development methodology. Since there were three main modules, application development was done phase wise by using incremental and Iterative approach that emphasizes continuous user involvement.

Its goal was to deliver the software systems on time and on budget while adjusting for changing requirements along the development process. Software Development Life Cycle (SDLC) can be illustrated as in below Figure 5.9.

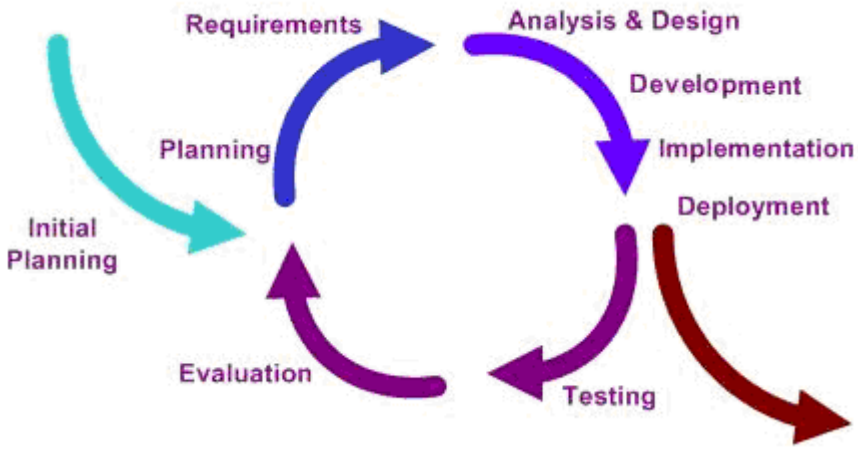

Figure 5.9 : System Development Life Cycle

5.10 Database Design

For the proposed system, MS SQL Server database which is a Relational Database Management System (RDBMS) was used. The advantages of using an RDBMS are as follows;

- No data redundancy
- Data Integrity
- Reduced data duplication
- Data security
- Improve storage space efficiency

The database was normalized up to third normal form and aappropriate user permissions were set based on the authority level of the each user in order to achieve the security requirements of the data.

5.10.1 Logical Data Structure

Logical Data Structure (LDS) is a diagram to show relationships between entities. LDS for the system was built by following below three steps and shown in Figure 5.10.

- Identify the entities
- Find the RELATIONSHIPS between them
- Decide the CARDINALITY

The purposes of developing the LDS are listed below

- To provide a common understanding of business data elements and requirements
- To provides foundation for designing a database
- To facilitates avoidance of data redundancy and thus prevent data & business transaction inconsistency
- To facilitates data re-use and sharing

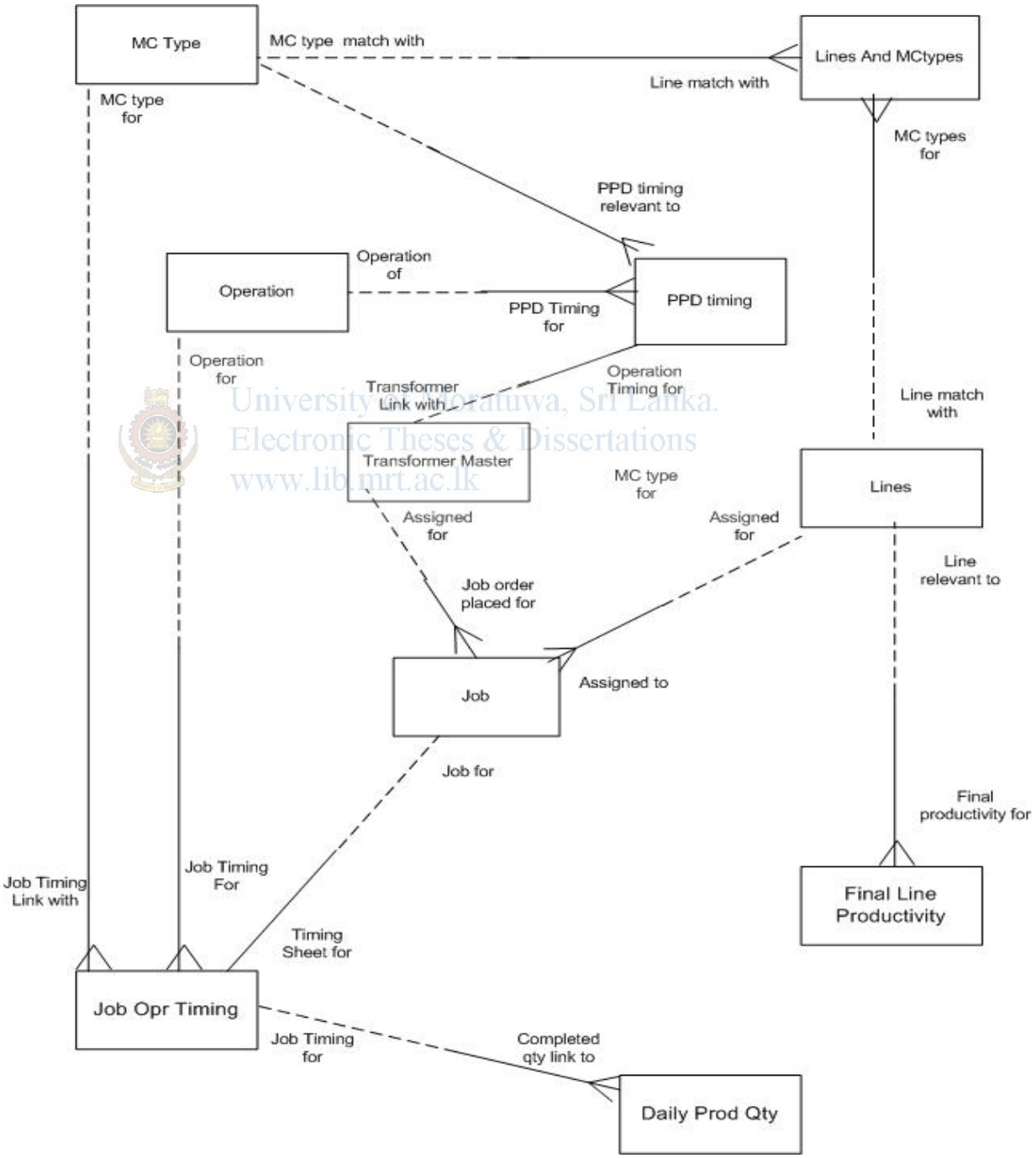

Figure 5.10 : Logical Data Structure

5.10.2. Relation Ship Diagram

Entity Relationship(ER) Diagrams are a major data modelling tool and helps to organize the data in the project and define the relationships between the entities. Entity, relation, and attributes were considered to form a conceptual view of data. In order to make easy of creating and maintaining the database structures, entity relational data model was created based on the logical Data Structure. It is shown in Figure 5.11. Primary key fields which provide a unique identifier for each record were identified and relationships between tables were created using matching fields of tables.

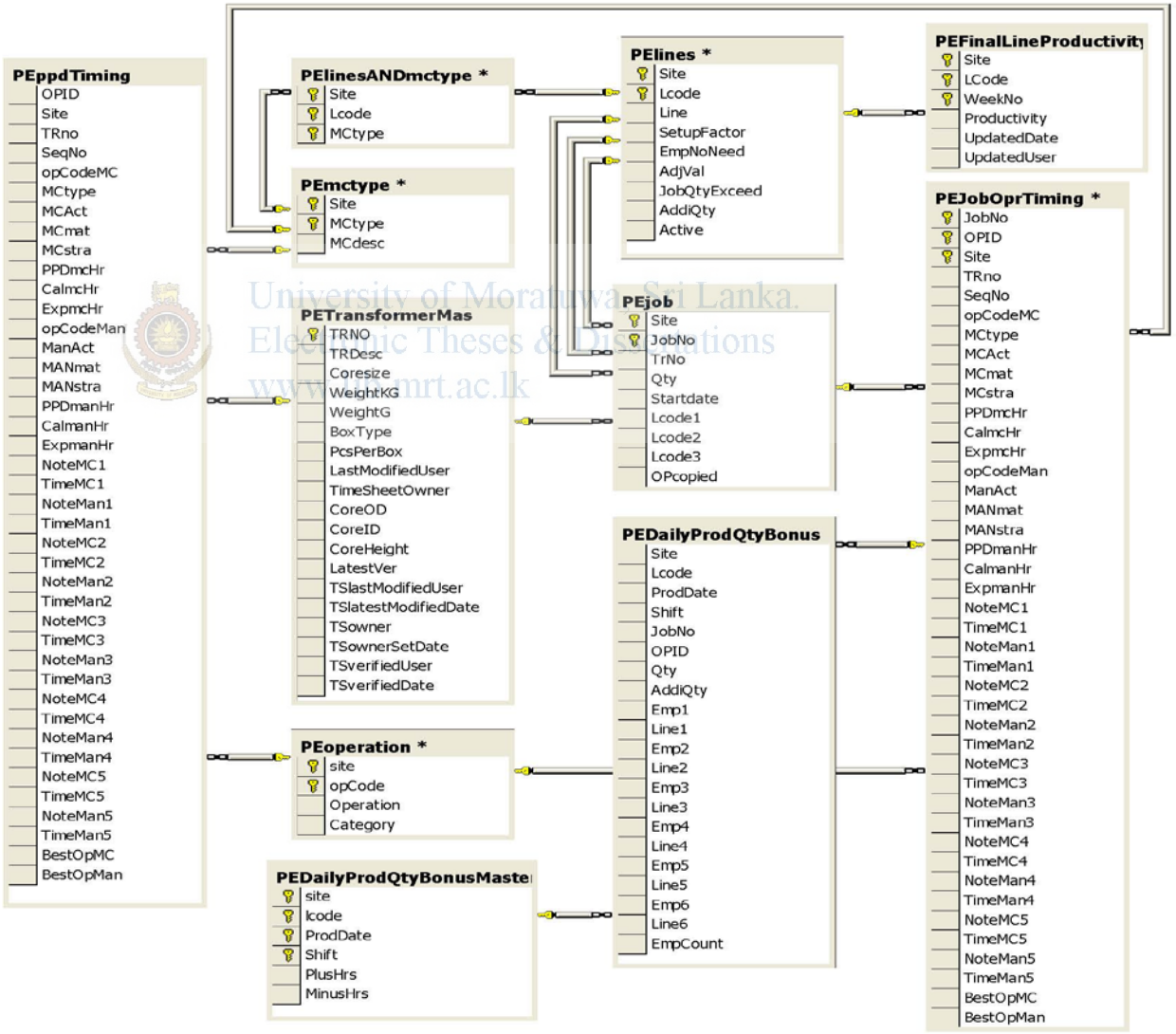

Figure 5.11 : Relation Ship Diagram

5.11 Graphical User Interface (GUI) Design

Technology alone may not win user acceptance and how the user experiences the end product, is the key to acceptance. Hence much attention was paid in the process of user interface designing in order to deliver a user friendly application.

A common template was used in all the interfaces to maintain the consistency. In order to improve the usability of the system, always best practices of GUI designing were used in user interface designing. Some of the focused factors are listed below.

Visibility of system status

Interfaces were designed such a way that system should always keep users informed about what is going on, through appropriate feedback within reasonable time. Keeping buttons in active mode and inactive mode, using different colours, system messages like techniques were used.

Match between system and the real world

All the efforts were taken to ensure that the system speaks the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Real-world conventions were followed while making information appear in a natural and logical order.

User control and freedom

"Exit" facility was provided in all the interfaces in order to provide facility to leave the unwanted state without having to go through an extended dialogue. This is because Users often choose system functions by mistake.

Error prevention

All the possible areas were carefully designed such a way that prevents from problems occurring by giving better error messages in the first place and presented users with a confirmation option before they commit to the action.

Recognition rather than recall

Efforts were taken to minimize the loading of user's memory by making objects, actions, and options visible. Instructions were kept visible or easily retrievable whenever appropriate. Further interfaces were designed such a way that user does not need to remember information from one part of the dialogue to another.

Aesthetic and minimalist design

Irrelevant information or rarely needed information was not included in the dialogues because every extra unit of information diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors

Error messages were expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

5.12 Summary

System Analysis and Designing process was discussed in this chapter. Further, methodologies and techniques which were used for Requirement Analysis, Business Solution Options, Feasibility Study, and Architectural Design also have been discussed in this chapter.

Next chapter discusses about the testing of the solution and implementation methodologies.