**CHAPTER 3**

**SYSTEM DESIGN**

**3.1 Introduction**

Design is the abstraction of a solution, it is the general description of the solution to a problem without a details. Design is view pattern seen in the analysis phase to be a pattern in a design phase. After design phase we can reduce the time required the implementation

**3.2 Purpose**

For thepurposes of this chapter, pharmacy systems are defined as computer systems designed specifically for pharmacy departmental use, with functionality for the management of pharmacy and dispensing processes, such as medicine labelling, patient medication records, decision support for drug interactions and other.

The pharmacy management system is built for the sake of ensuring effective and clear data saving and manipulating as well as neat work on the pharmacy medical products. This refers the pharmacy management system project highly minimize time and resource by which, searching the medicine data you can get the data in quickest time. And almost the resources are wise used since most actions are done on the pharmacy system. Some of the resources minimized include paper, manpower and related things. The other thing is for storing data's in secure way

So the purpose is clear now that, this system will provide desired medicine information to their front end users. The can search medicine by medicine name and district id. And by this way user can get his nearest shop address and the price of his desired product. That will save time and more efficient. The shops, the system will show will be registered sellers of the system. The system admin or super admin will control the seller ids. Registered sellers can add or delete or modify product information and keep their transaction information on this system. Sellers pay to the super admins for their service.

**3.3 Design Goals**

The main purpose of the application is to automate the existing system of manually maintained records of the counter sales, purchases, reorder levels, Supplier and Customer monetary positions and other related transactions made by the seller. Following are the some goals of the system design

* Provide for mass storage of relevant data.
* Make access to the data easy for the user.
* Provide prompt response to user requests for data.
* Making modifications to the database available immediately.
* Allow for multiple users to be active at one time.
* Protect the data from physical harm and unauthorized access.
* Get alert bout the drugs close to be expired,
* Get Alert about the drugs going to be finish in the stock.
* Having a good statistics part to know how much profit gained daily, monthly or even the specific date that the manager may need to get report about it.

**3.4 Entity Relation Diagram**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals, and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

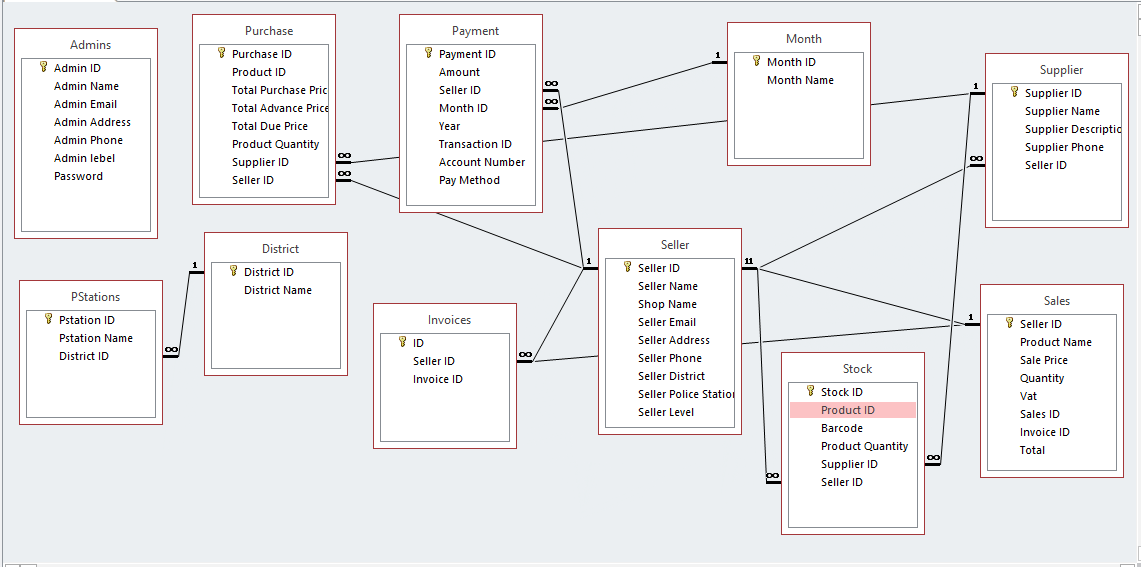


Figure: 3.1 Entity Relation Diagram

**3.5 Data Flow Diagram**

A data flow diagram (DFD) is a graphical representation of the "flow” of data through information systems. DFD's can also be used for the visualization of data processing (structured design). A DFD also known as "bubble chart has the purpose of clarifying system requirements and identifying major transformations. It shows the flow of data through a system. It is a graphical tool because it presents a picture. The DFD may be partitioned into levels that represent increasing information flow and functional detail. Four simple notations are used to complete a DFD.

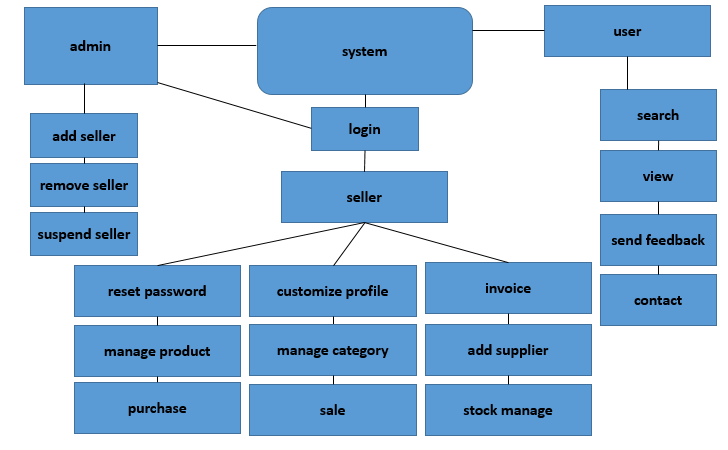


Figure 3.2: Data Flow Diagram

**3.6 Use Case Diagram**

Use case diagram are usually referred to as behavior diagrams used to describe a s et of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

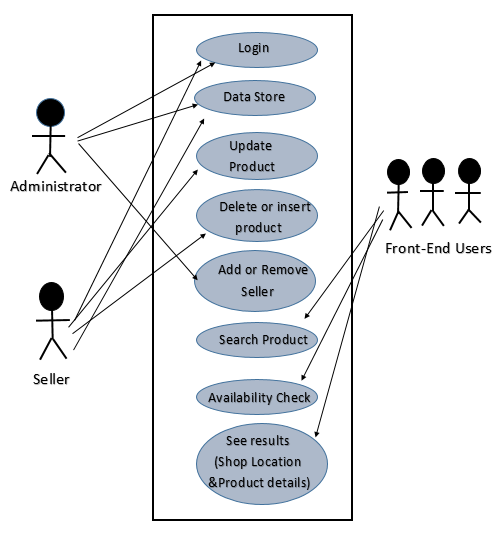


Figure 3.3: Use Case Diagram

**3.7 System Context Diagram**

A system context diagram (SCD) in engineering is a diagram that defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it. This diagram is a high level view of a system. It is similar to a block diagram. System Context Diagrams represent all external entities that may interact with a system. Such a diagram pictures the system at the center, with no details of its interior structure, surrounded by all its interacting systems, environments and activities. The objective of the system context diagram is to focus attention on external factors and events that should be considered in developing a complete set of systems requirements and constraints. System context diagrams are used early in a project to get agreement on the scope under investigation.

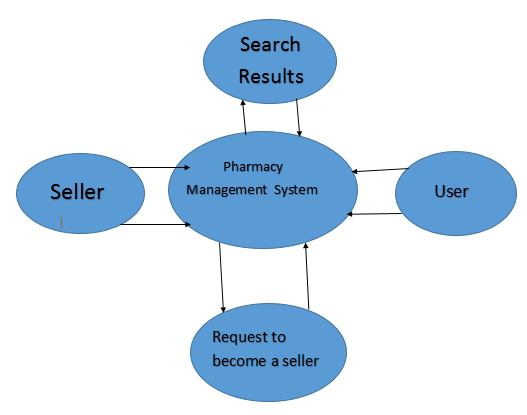


Figure 3.4: Context Diagram

**3.8 Data Dictionary**

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data. The data dictionary is a crucial component of any relational database.

**Districts:** {district id + district name}

**Invoices:** {id + seller id + invoice id}

**Pstations:** {pstation id + pstation name + district id}

**Purchase:** {purchase id + product id + total purchase price + total advance price + total due price + product quantity + supplier id + seller id}

**Seller:** {seller id + seller name + shop name + seller email+ seller address + seller phone + seller district + seller police Station + seller label}

**Stock:** {stock id + product id + barcode + product quantity + supplier id + seller id}

**Suppliers:** {supplier id + supplier name + supplier description + supplier phone + seller id +}

**Admins:** {admin id + admin name + admin email + admin address + admin phone + admin label + password}

**Month:** {month id + month name}

**Payment:** {payment id + amount + seller id + month id + year + transaction id + account number + pay method}

**Sales:** {product name + seller id + sale price + quantity + vat + sales id +invoice id + total}