**1. Introduction:**

1.1 **Purpose** -

Telephone directories serve a vital purpose in the organization of basic information of people such as name, contact details and their home addresses. Accumulation of this information in an orderly and accessible manner is necessary to make it useful to the everyday user. The document encompasses the requirements involved in a Telephone directory systems and identifies their functional and non-functional nature.

Conventionally, telephone directories are in the printed form and involves wastage of paper. The Telephone Directory System software helps by avoiding manual labour and paper work and presents itself in a compact form for the user's convenience.

1.2 **Scope –**

The proposed software is the Telephone Directory System (TDS) software.

* The scope of this software extends to any person to get contact details or address information of other people living in the region and store the records in an efficient manner.
* It allows a user to maintain privacy, retrieve desired information quickly and enables them to save the contacts they would like to refer for future purposes.
* The motive of the software caters to maximizing the number of unique person records in the directory and minimizing the resource costs, ensuring reachability.

1.3 **Overview** -

On the whole, the TDS software deals with person identification details with the use of modules dedicated for specific purposes.

They serve to achieve the following:

* Allowing the user to register their entry with the administration.
* Provide service of making search queries to the users ( general users and admins).
* Make provisions to the users to delete/take down their information to uphold their privacy.
* Ensuring that the addition of person records and their modification by the user takes place through the admin to prevent addition of incorrect/invalid information records.

This document describes the software product, user characteristics, specific requirements (functional and non-functional) and interfaces required.

1.4 **Definitions -**

a. Telephone Directory – A telephone directory, also known as a telephone book, telephone address book, phone book, or the white/yellow pages, is a listing of telephone [subscriber](https://en.wikipedia.org/wiki/Subscriber)s in a geographical area or subscribers to services provided by the organization that publishes the directory.

b. GUI or Graphical User Interface – A visual way of interacting with a computer using windows, icons, menus, OS, etc.

c. Database – It is the collection of schemas, tables, queries, reports and other objects organized to model aspects of reality.

2. **Overall Description -**

2.1 **Product Perspective -**

* A cross-platform self-contained and independent software product.
* Web-based access system.
* Integrated modules to share and exchange information in real-time.
* Inclusion of facilities of registration, login, addition, modification, deletion of telephone records.
* Modular concept and flexible in choice of modules per application.
* Allowing access only to specific modules for user-specific applications.

2.2 **Product Description**

* Login - To enable authorized access to the software.
* Checks to see if the user is already registered upon entry of their details, otherwise they are directed towards registration.
* The system generates search results for queries and incorportes the feature of saving contacts.
* Users can access the records from any location due to the web-based environment.

2.3 **User Characteristics -**

* User-friendly product which meets the ergonomic requirements.
* User must have general computer skills and should be comfortable in the English language.
* Technically competent in the general purpose applications on the computer.

2.4 **Abbreviations -**

**OS** – Operating Systems

**DBMS** – Database Management System

**Info** – Information

**Admin** – Administration

**HL7** – Health Level 7

**SQL** – Structured Query Language

**IDE** – Integrated Development Environment

**GUI** – Graphical User Interface

**SOAP** – Simple Object Access Protocol

**PHN** – Personal Health Number

**LAN** – Local Area Network

3. **Specific Requirements**

3.1 **Functional Requirements -**

3.1.1 **Reception Module**

a. Introduction – This module handles various enquires about the patients admission and discharge details. It handles Doctor consultation and scheduling, fees and time allocation for appointments and treatments.

Submodules are:

- Doctor Visit Schedule

- Doctor Appointment Schedule

- Patient Enquiry

- Patient History

b. Input – Personal details of patients, illness type, minor health related inquiries ( for new entries ), PHN for regular patients.

c. Processing – To store input details into the SQL database using DBMS or check PHN with database corresponding to the patient for verification.

d. Output – Time and name of Doctor allotted for the allotment.

e. Attributes – Patient Name, PHN, Age, Address, Medical History, etc.

f. Others – To record patient queries and feedback.

3.1.2 **Administration Module**

a. Introduction – This module handles all entry details for hospital requirement such as consultation detail, doctor specialization, consultancy fees, staff salary, transaction and service charges.

b. Input – To input employee name, department and specialization.

c. Processing – To tally input with the DB and look through employee records.

d. Output – Enquired details of the employee along with consultation fees.

e. Attributes – Employee Name, ID No, Type, Qualifications, History, etc.

f. Others – To record employee details and referrals of doctors.

3.1.3 **Registration/ Discharge Module -**

a. Introduction – This module helps in registering information about patients and handling both inpatients and outpatients' queries. A unique PHN is generated for each patient after registration. This helps in establishing relationship with customer and tracking medical history of the patient.

b. Input – Detailed information of patient such as Name, Age, Address, Medical Certificates, past medical records, illness type and duration, doctor preference, inpatient or outpatient, ward and bed type ( for inpatients ), to avail nurses for full-time or not.

c. Processing – To take the user input and store it in an empty record of the DB for inpatients and outpatients separately. Look for ward vacancies and availability of services or discharge and deleted accommodation status.

d. Output – Allotment of health consultant, ward, nurse name and qualification details, facilities provided in the chosen room type.

e. Attributes – Name, Age, Address, Medical records, Health Problem, Doctor Name, Doctor Type, Room No and Type, Nurse ID and Name, etc.

f. Others – To accommodate as many patients as possible in the wards and treatment rooms.

3.1.4 **Laboratory Module -**

a. Introduction – This module enables the maintenance of requests by the experts and generation of test results for services such as X-ray tests, etc. Requests can be made from multiple sources. It is linked with the registration module and billing module.

b. Input – Doctor ID and password, PHN of patient and treatment type.

c. Processing – To authenticate expert, obtain patient record using PHN, verify health responses to allergies and look for past test results.

d. Output – To produce test results such as blood pressure, allergies, blood group, test remarks by the expert.

e. Attributes – Doctor ID, Doctor password, PHN, Treatment Type, Height, Weight, Blood Pressure, Medical Certificates, etc.

f. Others – To provide a summary of the result to the patient and linking the cost to the billing module. It contains the facility of viewing lab equipment, their maintenance history and usage past, which can be viewed by authorized personnel.

3.1.5 **Billing Module -**

a. Introduction – This module sums all the costs and transactions made by the patients, and also develops purchase records for the hospital resources.

b. Input – PHN of patient.

c. Processing – To tally the patient's PHN, obtain record of transactions made and services availed and produce bill report.

d. Output – Total cost including service charges.

e. Attributes – Patient Name, PHN, etc.

f. Others – To add bills to patients' records and hospital database.

3.2 **Non-Functional Requirements –**

3.2.1 **Reliability –**

* The HMS software should be available 24/7 (incorporating scope for cases of emergency).
* It should present real-time information of the hospital functioning.

3.2.2 **Security –**

* The software should maintain a database containing username and logins for employees and PHNs for patients to establish authentication prior to use of product.
* The payment gateway should be rid of loopholes to restrict undesired interventions.

3.2.3 **Performance –**

* The queries made by users must be readily retrieved from the database.
* It should be able to handle multiple users working on different modules from various locations in the hospital.
* Traffic-independent responses have to be brought about.

3.2.4 **Interoperability –**

* The software should be developed in a manner such that it can be easily integrated with other devices/software in hospitals.

3.2.5 **Usability –**

* The HMS software must be user-friendly and follow an interactive GUI.
* The web interface should be simple to navigate to ensure user convenience.

4. **External Interface -**

4.1 **User Interface -**

The HMS software:

* Provides a user-friendly interface to the user with a simple GUI and follows an organized format which is intuitive to use.
* Contains hospital header at all point to maintain consistency.
* Has client application of Windows-based programming.
* Contains server application implementation through runtime hosts.

4.2 **Hardware Interface -**

* Intel core i3-i7.
* Minimum 256MB RAM onboard.
* 1TB Hard Disk for.

4.3 **Software Interface -**

* Operating systems – Windows, Android, iOS, Symbian, etc.
* Programming language – Microsoft's .NET, C#.
* Web technology – ASP NET 2.0.
* Back-end – SQL server.
* Web-server – Apache.

4.4 **Communication Interface -**

* LAN.
* Must be connected with the central database.
* Remote serial devices.