**UNIVERSITY OF IBADAN**

**DEPARTMENT OF COMPUTER SCIENCE**

**CSC 747 (ADVANCED DATABASE SYSTEM) GROUP ASSIGNMENT**

**ON**

**HEALTH CARE CENTER**

**SUMBITTED BY**

**GROUP 9 MEMBERS**

1. **ORONIRAN OLUFUMILOLA 182817**
2. **UKAIRO UGONNA IDIKA 209047**
3. **LAWAL OLALEKAN BABATUNDE 147118**
4. **MUSTAPHA ABDULRASHEED**
5. **SANGOTOYE OLUDAMIPE AYOTUNIDE 211245**
6. **FOLORUNSO OLAIDE BEATRICE 208475**
7. **BALOGUN SAHEED ABIODUN 211202**

**INTRODUCTION**

Today the health care sector faces serious and increasing problems concerning limited resources for effective disease prevention management. In most of our Western societies, the “acute care”  
paradigm has led to the design of “find-it, fix-it” health systems, but they are not meeting the changes in demand of care services prompted by the aging population. The economic and social burden is pressing heavily on governments, health care providers and citizens. Emerging proposals are stressing the imperative need to redesign the provision of services in ways that are more flexible. Technology, especially mobile applications, seems to be the cornerstone that will make this transformation possible.

**BASIC REQUIREMENTS OF THE SYSTEM**

* Search for patient
* Register staff
* Update, staff record, patients
* View all types of records
* Assign access rights and privileges to the system users

**THE FOLLOWING ARE THE DESIRED FUNCTIONALITIES OF THE NEW SYSTEM**

1. The system should accept have submissions in form or raw patients, staff, and drug supply at the submitting point.
2. The system should perform analysis of financial, drug inventory, patients and drug supply.
3. The system should generate reports on requests
4. The system should only allow the administrator to delete records in the database.

**CONCEPTUAL SCHEMA FOR THE HEALTH CARE SYSTEM**

1. **USER** = (`USER\_ID` int not null auto\_increment, `USERNAME` varchar(45) not null, `PASSWORD` varchar(45) not null, `EMAIL` varchar(45) not null, `USER\_CATEGORY` varchar(45) not null, `privilege` varchar(45) not null, `F\_NAME` varchar(45) null, `L\_NAME` varchar(45) null, `sex` varchar(45) null, `DOB` varchar(45) null, `MARITAL\_STATUS` varchar(45) null, `COUNTRY` varchar(45) null, `STATE` varchar(45) null, `ADDRESS` varchar(45) null, `PHONE` varchar(45) not null, `PHOTO` varchar(45) null, `BLOOD\_GROUP` varchar(45) null, `GENOTYPE` varchar(45) null, `WEIGHT` varchar(45) null, `HEIGHT` varchar(45) null, `DISABILITY` varchar(45) null, `LAST\_ACTIVE` datetime not null default current\_timestamp on update current\_timestamp, `PARENT\_ID` int null)
2. **MEDICAL\_HISTORY=** (`DIAGNOSIS\_ID` int not null auto\_increment, `USER\_ID` int not null, `SYMPTOMS` varchar(45) null, `TEST\_TYPE` varchar(45) null, `TEST\_RESULT` varchar(45) null, `PRESCRIPTION` varchar(45) null, `DOCTOR` varchar(45) not null, `DATE` datetime not null default current\_timestamp, `invoice` varchar(45) null)
3. **INVENTORY =** ( `ID` int not null, `PRODUCT\_ID` varchar(45) null, `SUPPLIER\_ID` varchar(45) null, `QTY\_AVAILABLE` varchar(45) null, `PRODUCT\_CATEGORY` varchar(45) null, `cost/qty` varchar(45) null, `BATCH\_NUM` varchar(45) null, `SUPPLY\_DATE` varchar(45) null, `MFG\_DATE` varchar(45) null, `BB\_DATE` varchar(45) null, `PRODUCT\_NAME` varchar(45) null, primary key (`id`))
4. **SUPPLIER** = (`SUPPLIER\_ID` int not null auto\_increment, `SUPPLIER\_NAME` varchar(45) null, `PHONE` varchar(45) null, `address` varchar(45) null, `EMAIL` varchar(45) null, `SUPPLIER\_ACCNO` varchar(45) null, `SUPPLIER\_BANK` varchar(45) null, `SUPPLIER\_PRODUCT` varchar(45) null, primary key (`SUPPLIER\_ID`))

**COMPOSITE ENTITIES**

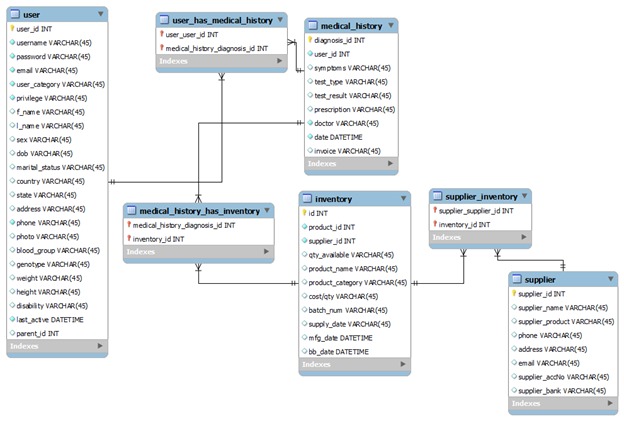
1. **SUPPLIER\_INVENTORY**`=(`SUPPLIER\_ID`,`INVENTORY\_ID`, index `fk\_supplier\_has\_inventory\_inventory1\_idx` (`inventory\_id` asc),index `fk\_supplier\_has\_inventory\_supplier\_idx`(`supplier\_supplier\_id`asc),constraint`fk\_supplier\_has\_inventory\_supplier`foreignkey(`supplier\_supplier\_id`).healthcare`supplier` (`supplier\_id`)on delete no actionon update no action, constraint `fk\_supplier\_has\_inventory\_inventory1`foreign key (`inventory\_id`) references `healthcare`.`inventory` (`id`)on delete no actiononupdate no action)engine = innodb;

2. **HEALTHCARE`USER\_HAS\_MEDICAL\_HISTORY** = (`user\_user\_id`, `medical\_history\_diagnosis\_id`),index`fk\_user\_has\_medical\_history\_medical\_hisory1\_idx`(`medical\_history\_diagnosis\_id`asc),index`fk\_user\_has\_medical\_history\_user1\_idx` (`user\_user\_id` asc),constraint `fk\_user\_has\_medical\_history\_user1`foreign key (`user\_user\_id`)references `healthcare`.`user` (`user\_id`)on delete no actionon update no action,constraint `fk\_user\_has\_medical\_history\_medical\_history1`foreign key(`medical\_history\_diagnosis\_id`)references `healthcare`.`medical\_history` (`diagnosis\_id`)on delete no actionon update no action)engine = innodb;

3. **MEDICAL\_HISTORY\_HAS\_INVENTORY**= (`MEDICAL\_HISTORY\_DIAGNOSIS\_ID`, `INVENTORY\_ID`), index `fk\_medical\_history\_has\_inventory\_inventory1\_idx` (`inventory\_id` asc), index `fk\_medical\_history\_has\_inventory\_medical\_history1\_idx`(`medical\_history\_diagnosis\_id` asc),constraint `fk\_medical\_history\_has\_inventory\_medical\_history1`

foreign key (`medical\_history\_diagnosis\_id`)references `healthcare`.`medical\_history`(`diagnosis\_id`)constraint`fk\_medical\_history\_has\_inventory\_inventory1`foreign key (`inventory\_id`)references `healthcare`.`inventory` (`id`)on delete no actionon update no action) engine = innodb;

**ENTITY RELATIONSHIP DIAGRAM (ERD) FOR THE HEALTH CARE SYSTEM**

****

**CONCLUSION**

This is a simple and typical healthcare database design. We have common entities and their attributes to handle the general operations of a healthcare establishment.

The database system would be able to handle requirements such as:

* Search for patients who visit the healthcare center
* Registers new employees of the center
* Update staff record and patients
* View all types of records that are entered into the database
* Assign different levels of access rights and privileges to the system users