HERB & DISEASE

Submitted by: Fatma Zaman

Table of Contents:

1. Overview:

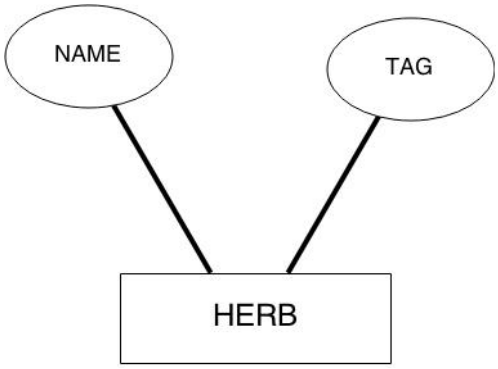
The purpose of this project is to build a system where we can get the detailed info about any given Herb and to Identify which Herb can be used for the cure of a particular disease and viceversa.

2. Database Design:

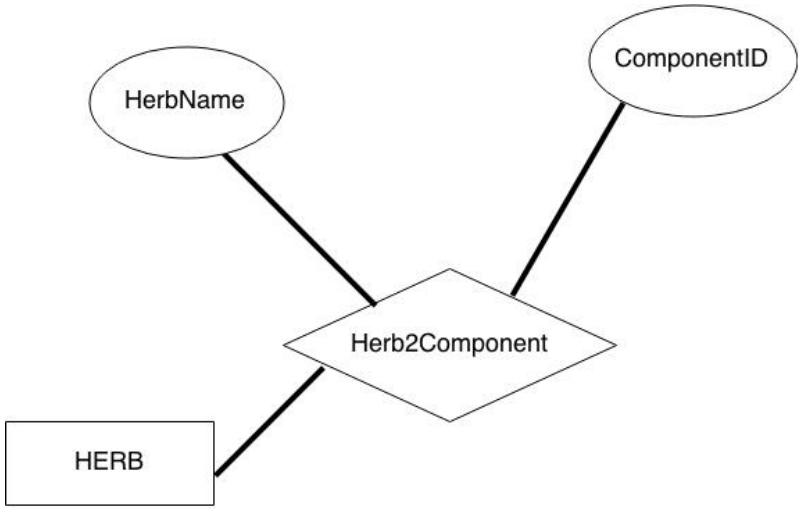
Below ER-diagrams are the overview of the database.

Entity-Relationship Diagrams:

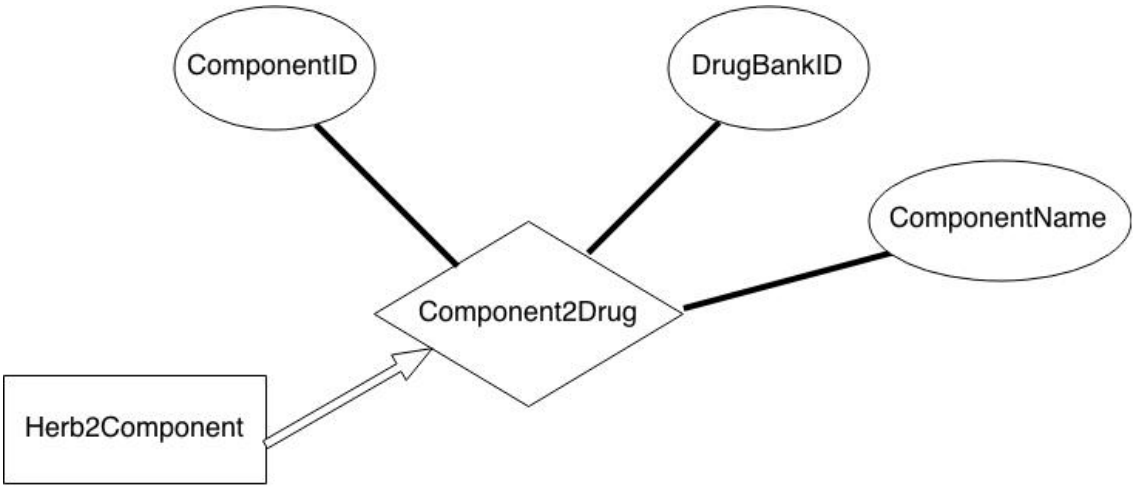
HERB



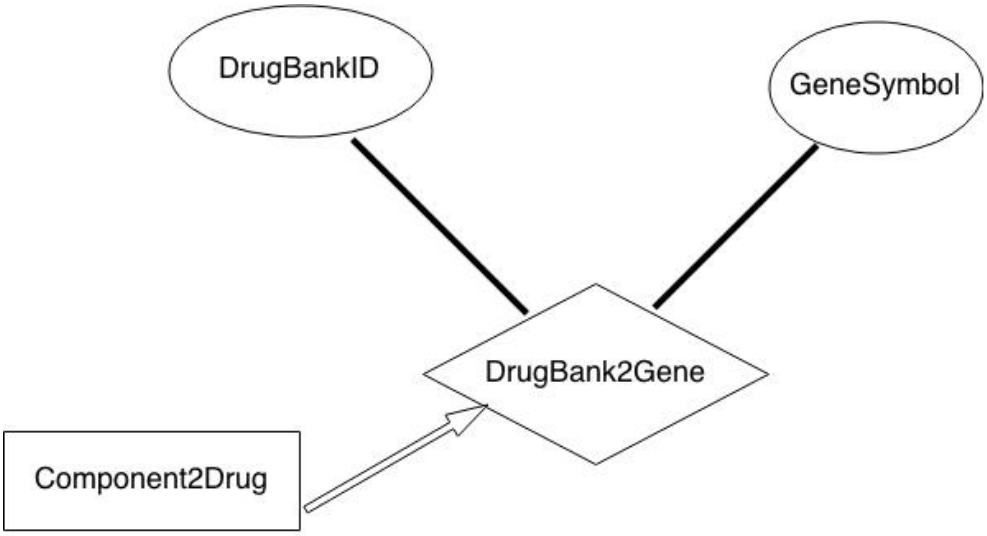
HERB_COMPONENT



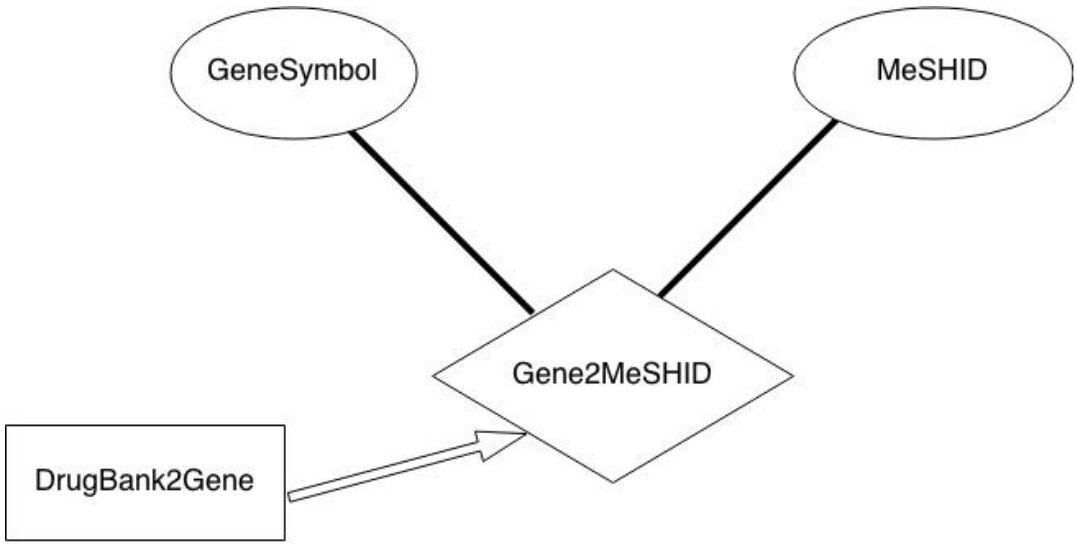
COMPONENT_DRUGBANK



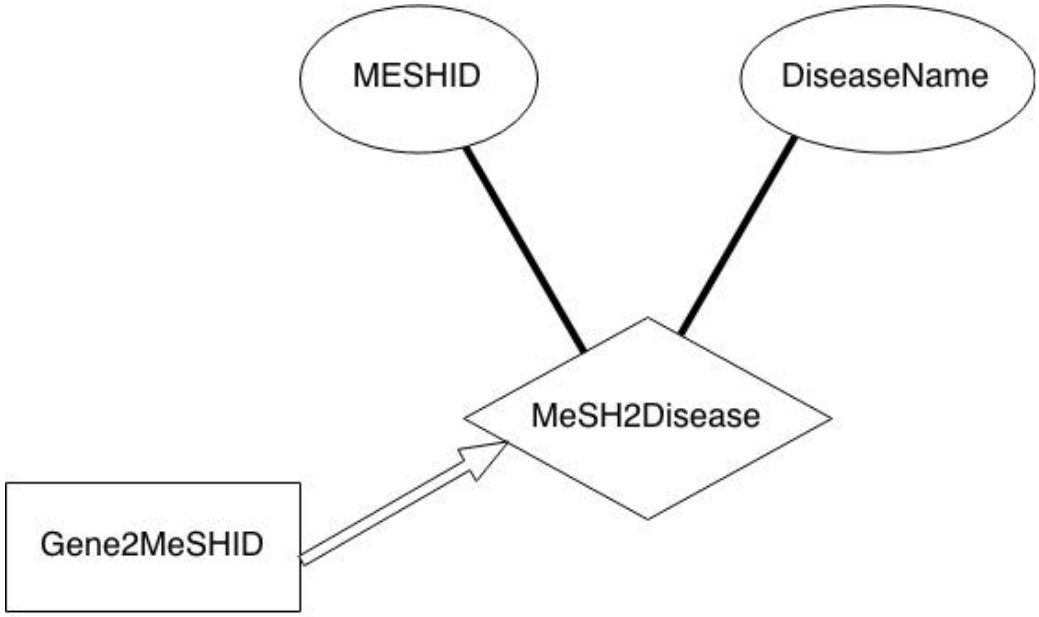
DRUG_GENE



GENE_MESH



MESH_DISEASE



3. Functionality:

This section describes about how this whole process of passing input till the output works.

Starting from the front-end part where there is one text box and user have to fill it with one Herb Name and click on submit. After submitting the Herb Name we will get the respective disease for which this herb is used to cure.

4. Database Tables:

This section describes what data we will store in our database as well as how our relations represent this data.
The following tables will be included in our database. Please refer to the E-R diagrams at the beginning of the document for a higher-level graphical description.

HERB

Herb (*Name*: VARCHAR(100), *Tag*: VARCHAR(12))

foreign key: none
candidate key: Tag
primary key: Name
not null: Name

This tables contains Name and related Tag to of all the Herbs. The Name of the Herb is used as foreign key to the Herb_Component table to get all the components of this Herb.

HERB_COMPONENT

Herb2Comp(*HerbName*: VARCHAR(120), *CID*: VARCHAR(100))

foreign key: HerbName(Herb.Name)
primary key: CID
not null: HerbName, CID

This tables stores all the HerbName and their corresponding components. To get the Component ID,we use Herb name as a foreign key from the Herb table.

COMPONENT_DRUGBANK

Component2DrugID(*CID*: VARCHAR(20), *DID*: VARCHAR(20), *ComponentName:* VARCHAR(100))

foreign key: CID(Herb2Comp.CID)
primary key: DID
not null: CID, DID

This table consists data of all components and their respective DrugBank IDs. using the Component ID(CID) from previous table we get the corresponding Drug bank ID.

DRUG_GENE

Drug2Gene(*DID*: VARCHAR(20), *GSymbol*: VARCHAR(20))

foreign keys: DID(Component2DrugID.DID)
primary key: GSymbol
not null: DID, GSymbol

In this table, we are storing information of drugbank ID and its corresponding Gene Symbol. This Gene symbol will lead us to the MeSHID in the next table.

GENE_MESH

Gene2MeSH(*GSymbol*: VARCHAR(50), *MESHID:* VARCHAR(50))

foreign keys: GSymbol(Drug2Gene.GSymbol)
primary key: MESHID
not null: MESHID, GSymbol

This table stores the Information of Gene Symbol and its respective MESHID. This MESHID can be used to get the Disease in the next table.

MESH_DISEASE

MESH2Disease(*MESHID*: VARCHAR(20), *DiseaseName*: VARCHAR(120))

foreign keys: MESHID(Gene2MESH.MESHID)
primary key: DiseaseName
not null: MESHID, DiseaseName

Here in this table we finally get the disease name using its MESHID from the previous table. This table stores MESHID and its corresponding Disease Name.

5. SQL Queries:

Query for getting Disease from Herb

When a user puts Herb name into the text box and clicks on submit button he gets to see the Disease name to which this herb is related to.

SELECT CID from Herb2Comp where HerbName='$HerbName'

After getting the Components from the “Herb2Comp” table, We get the Gene Symbol by joining table “Component2DrugID” and “Drug2Gene”.

SELECT * from Drug2Gene inner join Component2DrugID on Component2DrugID.CID='$CID[0]' && Drug2Gene.DID=Component2DrugID.DID

Getting Gene Symbol will lead us to MESHID and then this MESHID will then lead us to Disease name. We get this by joining tables “Gene2MESH” and “MESH2Disease”

SELECT * from MESH2Disease inner join Gene2MeSH on Gene2MeSH.GSymbol='$GeneSymbol' && MESH2Disease.MESHID=Gene2MeSH.MESHID

6. Database Initialization Queries:

The following four SQL queries are used to initialize the database tables:

```
CREATE TABLE IF NOT EXISTS `Herb` (
  `Name` varchar(100) CHARACTER SET armscii8 NOT NULL DEFAULT '',
  `Tag` varchar(12) DEFAULT NULL,
  PRIMARY KEY (`Name`),
  KEY `Tag` (`Tag`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `Herb2Comp` (
  `HerbName` varchar(120) CHARACTER SET armscii8 NOT NULL,
  `CID` varchar(100) CHARACTER SET armscii8 NOT NULL,
  PRIMARY KEY (`HerbName`,`CID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `Component2DrugID` (
  `CID` varchar(20) CHARACTER SET armscii8 NOT NULL,
  `DID` varchar(20) CHARACTER SET armscii8 NOT NULL,
  `ComponentName` varchar(100) CHARACTER SET armscii8 NOT NULL,
  PRIMARY KEY (`CID`),
  KEY `DID` (`DID`,`ComponentName`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `Drug2Gene` (
  `DID` varchar(20) CHARACTER SET armscii8 NOT NULL,
  `GSymbol` varchar(20) CHARACTER SET armscii8 NOT NULL,
  PRIMARY KEY (`DID`,`GSymbol`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `Gene2MeSH` (
  `GSymbol` varchar(50) CHARACTER SET armscii8 NOT NULL,
  `MESHID` varchar(50) CHARACTER SET armscii8 NOT NULL,
  PRIMARY KEY (`GSymbol`,`MESHID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `MESH2Disease` (
  `MESHID` varchar(20) CHARACTER SET armscii8 NOT NULL DEFAULT '',
  `DiseaseName` varchar(120) CHARACTER SET armscii8 DEFAULT NULL,
  PRIMARY KEY (`MESHID`),
  KEY `Disease Name` (`DiseaseName`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```