**Project Proposal Due Date:** December 15, 2014

**Name of Application:** Patient-Tracking System

**Author:** Fadwa Khalil

**Access information:**

URL: <http://cscie60.dce.harvard.edu/~fkhalil/fp/index.cfm>

User Administrator username: fkhalil

Password: zahranadia77

**Final Project**

**Description of the application:**

you have built including a brief description of the purpose of the application and description of any forms, queries, reports and menus you have built. You should be able to reuse some of the information provided in the proposal and adjust it accordingly.

A friend of mine “Angela Brennan” who is a Licensed Electrologist and skin therapist owns a Clinic in Plaistow, NH called “Davanti Skin Care & Electrolysis”. Currently, Angela keeps her patients records in information cards and fetches those cards every day based on her appointments scheduled on her phone calendar to keep track of every patient status, health condition and the type of treatment she’s providing for that particular patient.

My goal in this project is to replace Angela’s current paper-based form cards with a web-based database system so that Angela and her patients will be benefited from this data collection and information sharing efficiency.

My mission is to enable her to log in new patients, update her already existing patients records/history, schedule appointments online, keep track of history appointments and billing, and benefit from generating different kinds reports.

My vision for this transformation is to make Angela’s life easier, operate faster and in a safer approach and would enhance the interaction between Angela and her patients rather than depending on manual work, which is prone to human error.

Basically, the following should be achieved:

* Workflow automation via Web-based DBM service or process interactions.
* Make business process Web accessible.
* Automate the data collection, billing and data storage process
* Transform the paper-based work into a web-based real time database system.
* Create integrated views, access or transactions to a reliable and consistent information sources.
* Angela will exclusively possess unique and more quality DB content compared to competitors in her region.

**Scope of the study:**

The application will include forms and reports necessary for entering information in the database and to allow reporting.

The following is a list of tentative forms:

1. Patients Registration (add / update).
2. Appointments (add, update, and delete).
3. Billing (add, update, and delete).

The application will include the following reports:

1. All patients information with status (Active/inactive)
2. Appointments history.
3. Billing (per week, per month, per year, etc.).

Other reports include a main menu for the application, with all available options, and search functionality.

**Items to take into consideration:**

* How to insure confidentiality and proper use of the database by using security-protection mechanism to prevent data tampering from hackers and ensure the data integrity.
* User Authentication
* Angela should easily query and analyze each patient’s history information with much higher speed and ease.
* She can collect and manage data by assigning “patient IDs” to each patient, and this number will be used for all transactions on this database.

**Methodology:**

I had a meeting with Angela to discuss with her how this transformation would help her on daily bases. I was able to gather all data required to design the database for this project from her, the owner of this business. However, to preserve the confidentiality of the information, I have listed dummy data to resemble the real data using an online sample data generator.

**Original Work:** I hereby certify that this project was prepared especially for this course, and that this or a similar version has not been submitted to any other course.

**Prototype:**

There will be three major sets of tables forming a relational Database. The different three sets are:

Personal related, medical related and Billing related:

* The personal-related tables will be used to store personal information such as name, age, gender, residence, etc. as well as the status of each patient.
* The medical-related tables will be used to store information such as patient’s medical history, family disease history, allergies, medications/treatments, etc. as well as the treatment Intensity given/recommended
* Billing-related tables will be used to store appointment scheduling and time based billing.

Data types: All three-table categories will be mostly text based (either Character or number or Date).

The ultimate database may contain mostly textual information. Thus, my initial estimate is that I will need about 10-TB of disk space to store all her current information, which would also be sufficient for her future expansion. I might be able to enhance it to store images for example.

**OTHER THINGS TO CONSIDER**

Internet

* Patients can communicate with each other (i.e. “chat room” to discuss their treatment)

Links

* Articles / Reference sites
* Key word search

Services

* Users want to get articles released/posted immediately “forwarded” to their email possibly as a link (i.e. patient wants to read about current articles).
* Users can communicate with Angela through sending email from the website.
* The system sends email notification to patients for appointment confirmation and reminder.

My fully normalized tables with defined Primary Keys: (Note: In phase two of this project when I start creating tables I will be testing the Integrity Rules and the model might encounter some modifications)

## FULL DATA MODEL

M

M PDisease M

Disease

1

Medication

1

Associative

Table

Basic Table

Reference Table

Legend

1 Patient 1

1 1

## RECORD DIAGRAMS

1

Package

M M

PBill

M

M

Pvisit

1

**PATIENT Entity**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Patientid** | **firstname** | **lastname** | **Street** | **City** | **State** | **Zip** | **DoB** | **Phone** | **email** | **Gender** | **MaritalStatus** | **Occupation** |
| **(PK)** |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | Earl | Smith | 53635 Green Ridge Lane | Grand Rapids | Michigan | 49560 | 3/14/1954 | (071)801-2921 | esmith0@a8.net | M | Single | Engineer |
| 2 | Beverly | Porter | 10170 Coolidge Alley | Newark | New Jersey | 07195 | 5/31/1957 | (240)676-4467 | bporter1@multiply.com | F | Married | N/A |
| 3 | Samuel | Bowman | 81797 Dakota Point | Detroit | Michigan | 48275 | 10/11/1997 | (077)305-9462 | sbowman2@vk.com | M | Widowed | Teacher |
| 4 | Raymond | Owens | 2730 Mayfield Center | Baton Rouge | Louisiana | 70805 | 8/11/1991 | (589)525-1248 | rowens3@vk.com | M | Divorced | Doctor |
| 5 | Linda | Washington | 96 Susan Terrace | Shawnee Mission | Kansas | 66225 | 4/24/1977 | (089)883-2701 | lwashington4@si.edu | F | Single | Nurse |
| 6 | Aaron | Howell | 67 Goodland Point | Pensacola | Florida | 32511 | 4/18/1949 | (946)765-3119 | ahowell5@loc.gov | M | Married | Engineer |
| 7 | Judy | Knight | 7 Kingsford Junction | Newport News | Virginia | 23612 | 2/27/1963 | (351)011-3331 | jknight6@cmu.edu | F | Widowed | N/A |
| 8 | Evelyn | Jenkins | 7 High Crossing Pass | Baltimore | Maryland | 21281 | 3/15/2013 | (224)940-3850 | ejenkins7@hostgator.com | F | Divorced | Teacher |
| 9 | Johnny | Castillo | 0702 Novick Trail | Katy | Texas | 77493 | 6/15/1999 | (201)005-2877 | jcastillo8@myspace.com | M | Single | Doctor |
| 10 | Alan | Kim | 43130 Shopko Way | Schenectady | New York | 12325 | 4/4/1953 | (462)610-7696 | akim9@kickstarter.com | M | Married | Nurse |

Data Type:

|  |  |
| --- | --- |
| **Patientid (PK)** | primary key, CHAR(4) |
| **firstname** | VARCHAR2(20), REQUIRED field |
| **lastname** | VARCHAR2(20), REQUIRED field |
| **Street** | VARCHAR2(30) |
| **City** | Char(20) |
| **State** | Varchar2(10) |
| **Zip** | varchar2(10) |
| **DoB** | Date, REQUIRED field |
| **Phone** | VARCHAR2(20) |
| **email** | VARCHAR2(50) |
| **Gender** | Char(1) |
| **MaritalStatus** | VARCHAR2(10) |
| **Occupation** | VARCHAR2(20) |

**PDisease Entity**

|  |  |  |
| --- | --- | --- |
| **Patientid** | **DiseaseID** | **MedID (FK)** |
| **(FK) (PK)** | **(PK) (FK)** |  |
| 1 | 7 | 0001 |
| 1 | 2 | 0002 |
| 3 | 7 | 0003 |
| 4 | 4 | 0004 |
| 4 | 7 | 0005 |
| 5 | 6 | 0001 |
| 2 | 7 | 0002 |
| 2 | 7 | 0003 |
| 1 | 1 | 0004 |
| 1 | 2 | 0005 |

Data Type:

|  |  |
| --- | --- |
| **Patientid (PK) (FK)** | Part of a composite primary key as well as a foreign key referencing the Patient |
| **DiseaseID (PK) (FK)** | Part of a composite primary key as well as a foreign key referencing the Disease |
| **MedID (FK)** | foreign key referencing the Medication, number (4,0) |

**Pvisit Entity**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **VisitID** | **Patientid** | **LastVisit** | **VisitDuration** | **NextVisit** | **Status** | **Intensity** |
| **(PK)** | **(FK)** |  |  |  |  |  |
| 1 | 1 | 02/14/2014 | 1 | 02/17/2014 | false | 1 |
| 2 | 1 | 02/16/2014 | 1.5 | 02/30/2014 | false | 2 |
| 3 | 2 | 03/14/2014 | 0.5 | 03/17/2014 | true | 3 |
| 4 | 2 | 08/01/2014 | 1 | 08/17/2014 | true | 3 |
| 5 | 3 | 06/20/2014 | 1.5 | 06/30/2014 | true | 2 |
| 6 | 3 | 04/21/2014 | 0.5 | 07/06/2012 | true | 2 |
| 7 | 4 | 06/29/2013 | 1 | 03/10/2014 | true | 2 |
| 8 | 4 | 12/27/2011 | 1.5 | 03/31/2014 | false | 1 |
| 9 | 5 | 08/15/2013 | 0.5 | 11/02/2013 | true | 1 |
| 10 | 5 | 09/23/2011 | 1 | 04/22/2011 | true | 1 |

Data Type:

|  |  |
| --- | --- |
| **Patientid (FK) (PK)** | Part of a composite primary key as well as a foreign key referencing the Patient |
| **VisitID (PK)** | primary key, use a sequencer, size 11 |
| **VisitDuration (PK) (FK)** | Number(2,1) Part of a composite primary key as well as a foreign key referencing the price |
| **LastVisit** | Date |
| **NextVisit** | Date |
| **Status** | Char(5) |
| **Intensity** | Char(1) |

**PBill Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Patientid** | **BillID** | **VisitID** | **PackageID** |
| **(FK) (PK)** | **(PK)** | **(FK) (PK)** | **(FK)** |
| 1 | 1 | 1 | 2 |
| 1 | 2 | 2 | 3 |
| 2 | 1 | 3 | 1 |
| 2 | 2 | 4 | 2 |
| 3 | 1 | 2 | 2 |
| 3 | 2 | 2 | 2 |
| 4 | 1 | 7 | 2 |
| 4 | 2 | 8 | 3 |
| 5 | 1 | 9 | 1 |
| 5 | 2 | 10 | 2 |

Data Type:

|  |  |
| --- | --- |
| Patientid  (FK) (PK) | Part of a composite primary key as well as a foreign key referencing the Patient |
| VisitID  (FK) (PK) | Part of a composite primary key as well as a foreign key referencing the PVisit |
| BillID (FK) | foreign key referencing the Price table |
| PackageID | foreign key referencing the Package table Primary key, CHAR(11) |

**Package Entity**

|  |  |  |
| --- | --- | --- |
| **PackageID** | **Duration** | **Price** |
| **(PK)** |  |  |
| 1 | 0.5 | 35 |
| 2 | 1 | 65 |
| 3 | 1.5 | 95 |

Data Type:

|  |  |
| --- | --- |
| **PackageID** (PK) | Primary key, CHAR(11) |
| visitDuration (PK) | Number(2,1) |
| Price | NUMBER(10, 2), cannot be negative, default of 0.00 |

**Disease Entity**

|  |  |
| --- | --- |
| **DiseaseID** | **Disease** |
| **(PK)** |  |
| 1 | Diabetes |
| 2 | STD |
| 3 | Allergy |
| 4 | Heart Disease |
| 5 | Hepatitis |
| 6 | Other |
| 7 | None |

Data Type:

|  |  |
| --- | --- |
| **DiseaseID (PK)** | Primary key, CHAR(2) |
| **Disease** | VARCHAR2(20) |

**Medication Entity**

|  |  |
| --- | --- |
| **MedID** | **MedType** |
| **(PK)** |  |
| 0001 | Med1 |
| 0002 | Med2 |
| 0003 | Med3 |
| 0004 | Med4 |
| 0005 | Med5 |

Data Type:

|  |  |
| --- | --- |
| **MedID (PK)** | Primary key, CHAR(4) |
| **MedType** | VARCHAR2(20) |

**A listing of bugs** you were unable to fix, and an action plan on how to fix them. If you are forthcoming in describing the issues we will face, you are less likely to lose too many points. Bugs your graders find when testing your application are those likely to cost you more points.

**Special features** you implemented. A portion of your grade is based on the implementation of special features (see below). You will receive points for them as long as the features work and they are documented in the report. The TAs will not be scouting for them.

**A full data model of your database design including record diagrams** as we have done in class. This is, you must include the ERD and record diagrams. Do not submit any of the intermediate steps in ERA database design.

**The database schema** including the create table statements, and the select \* from each table. You may use the DDL sample we presented in class as a template.

**The application code** (e.g., ColdFusion .cfm code). **Screen shots** of some of the screens of your application, showing database interaction. Think of it as a

walk through of your project.

Make sure that all the pieces work together before embarking on the finished result. For example, if your application requires an email or a file download capability, try out that functionality early in your application development. While we do not expect you to become an expert with the systems you choose, the idea is to demonstrate at least the course content through a sound example of a Web-based database application.

**GRADING GUIDELINES**

Grading the final project will emphasize the following:

**Database design and Implementation (100 points)**

The database design should represent fully normalized tables with effective table relationships and enforced referential integrity. Data integrity should be enforced in the form of validation rules, integrity constraints, and appropriate datatypes. There should be sufficient data in every table to demonstrate the full range of information management needs of this application. The implementation should include at least two of the following: triggers, transactions, procedures. For example, (a) two triggers, (b) one trigger and a procedure, or (c) two transactions. You must describe those in the final project report in order to receive credit for them.

**Client Application (80 points)**

The database application should offer presentable, user friendly, and informative features which allow the information to be managed while insuring that information inserted/updated/deleted does not violate any integrity constraints you have built into the database design. For example, consider implementing drop down lists (combo boxes) to your forms to offer a choice where the user would have little chance of knowing the set of acceptable values. The application should provide a presentable user interface, in the form of a menu. A good way of testing this portion is to ask a friend to test it without providing him any explanation of what to enter. Experience tells us many students do poorly in this section.

**Special Features (25 points)**

The application must include special features for enhancement. For example, adding a search tool to your application, or an email or file download functionality. **The definition of a special feature is one that was not required for any assignment.** The number of features will depend on the type of features you implement. Quality is more important than quantity. For example, securing your application is more time consuming than adding a search tool. Securing your application will most likely be enough, but implementing a search tool will need one or two additional special features. Use your best judgment. You may have to do some research to satisfy this requirement. **To earn credit, your report must include a description of the special features.**

**Report (25 points)**

The final report should clearly state the problem your information system solves, describe it accurately, list the most relevant queries, triggers, transactions, procedures, and special features, basics of the client application, and present it in good taste.