**Electronic Health Records Information System**

Emily Connor, [eec52@pitt.edu](mailto:eec52@pitt.edu)

Nicholas Tedesco, [nrt21@pitt.edu](mailto:nrt21@pitt.edu)

Yicheng Zhang, [yiz229@pitt.edu](mailto:yiz229@pitt.edu)

INFSCI2710 Database Management

Dr. Babichenko

April 25th, 2022

Table of Contents

[I. Introduction/Abstract 3](#_Toc101728745)

[II. UML-compliant E-R Model 3](#_Toc101728746)

[III. Business Rules 4](#_Toc101728747)

[IV. Entity/Attribute descriptions 6](#_Toc101728748)

[V. Closing section 10](#_Toc101728749)

# **Introduction/Abstract**

In this project, we have designed an electronic health record information system that is intended to be used by a small hospital system. This system allows users to view their patient information, doctors to view information about the patient, patients to be added to the system, and doctors to add notes about the patient. Relations in the database consist of patient and visit information, medications prescribed and procedures done at the visit, doctor information, as well as insurance and practice details. There are also invoice and payment relations in the database to keep track of payment details.

The target audience for this system is a small hospital system. The benefit of this system is that it enables smaller hospital systems to keep track of all the health records electronically without having to purchase an expensive license from an outside company where the design is more targeted to larger hospital systems. This design will serve as an efficient, small-scale electronic health record information system with lower overhead than current market systems.

# **UML-compliant E-R Model**

Diagram

Description automatically generated

# **Business Rules**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity 1 | Entity 2 | Cardinality (Entity 1 Side) | Cardinality (Entity 2 Side) | Business Rule(s) |
| patients | insurance | 1..\* | 1..\* | A given patient may have one or more health insurance plans (ex: primary plan, secondary plan). A given insurance plan may be associated with one or more patients (ex: family plan). |
| patients | addresses | 1..\* | 1 | A given patient should only have one address on file (i.e., primary address for contact info). A given address may correspond to one or more patients (ex: family living in one house).  Junction table: address\_info |
| patients | visit\_info (visits) | 1 | 1..\* | A given patient may have one or more visits (patient would not be registered if they had zero). A given visit should only correspond to one patient. |
| insurance | companies | 1..\* | 1 | A given insurance plan must only be associated with one insurance company. A given company may issue one or more insurance plans.  Junction table: insurance\_companies |
| companies | addresses | 1 | 1..\* | A given insurance company may have one or more addresses on file (ex: if different branches of insurance company are located at different addresses). A given address must only be associated with one company (i.e., multiple companies do not share same address).  Junction table: address\_info |
| pharmacies | addresses | 1 | 1..\* | A given pharmacy may have one or more addresses on file (ex: CVS has many locations). A given address should only be associated with one pharmacy.  Junction table: address\_info |
| practices | addresses | 1 | 1 | A given practice should only be located at one address. A given address should only correspond to one practice (even if multiple practices are located at same address, floor/apartment number should also be listed).  Junction table: address\_info |
| practices | visit\_info (visits) | 1 | 1..\* | A given practice may host one or more visits (the practice would not be in the system if it hosted zero visits). A given visit may only occur at one practice. |
| prescription\_info (prescriptions) | medications | 1..\* | 1 | A given prescription should only be for one medication. A given medication may be included in one or more prescriptions (the medication would not be in the system if it was not part of any prescriptions). |
| prescription\_info (prescriptions) | doctors | 0..\* | 1 | A given prescription should only come from one doctor’s order. A given doctor may have ordered anywhere from zero to many prescriptions. |
| prescription\_info (prescriptions) | visit\_info  (visits) | 0..\* | 1 | A given prescription should be ordered as a result of only one visit. A given visit may result in anywhere from zero to many prescriptions being ordered. |
| doctors | visit\_info  (visits) | 1 | 1..\* | A given doctor may attend one or more visits (the doctor would not be in the system if they had no visits). A given visit may only be associated with one doctor. |
| procedures | visit\_info  (visits) | 0..\* | 1..\* | A given procedure may be used at one or more visits (the procedure would not be in the system if it was not used in any visits). A given visit may involve anywhere from zero to many procedures.  Junction table: visits\_procedures |
| visit\_info (visits) | invoices | 1 | 1 | A given visit may only generate one invoice. A given invoice should only correspond to one visit. |
| invoices | payments | 1 | 0..\* | A given invoice may correspond to anywhere from zero to many payments (patient may not have paid bill at all yet, or may pay bill in smaller chunks). A given payment should only be for one invoice.  Junction table: invoices\_payments |
| visit\_info (visits) | doctorNotes | 1 | 0..\* | A given visit may correspond to anywhere from zero to many doctor notes (doctor may not feel that a note is necessary, or may enter notes at different points in time for visit). A given doctor note should only correspond to one visit.  Junction table: visit\_doctor\_note |

# **Entity/Attribute descriptions**

|  |  |  |
| --- | --- | --- |
| patients | | |
| patient\_ID (pk) | INT | Primary ID for patients table |
| first\_name | VARCHAR(50) | Patient’s last name |
| last\_name | VARCHAR(50) | Patient’s first name |
| phone | VARCHAR(30) | Patient’s phone number |
| email | VARCHAR(100) | Patient’s email |
| dob | DATE | Patient’s date of birth |

|  |  |  |
| --- | --- | --- |
| patients\_insurance | | |
| pi\_ID (pk) | INT | Primary ID for patients-insurance junction table |
| fk\_patient\_ID (fk) | INT | Foreign key referencing patients table |
| fk\_insurance\_ID (fk) | INT | Foreign key referencing insurance table |

|  |  |  |
| --- | --- | --- |
| insurance | | |
| insurance\_ID | INT | Primary ID for insurance table |
| policy\_num | VARCHAR(20) | Policy number for health insurance plan |
| deductible | INT | Dollar amount of deductible for insurance plan |
| deductible\_met | ENUM(“yes”, “no”) | Describes whether patient’s deductible is met |
| plan | VARCHAR(50) | Type of health insurance plan (ex: HMO, PPO) |

|  |  |  |
| --- | --- | --- |
| insurance\_companies | | |
| ic\_ID (pk) | INT | Primary ID for insurance-companies junction table |
| fk\_insurance\_ID (fk) | INT | Foreign key referencing insurance table |
| fk\_company\_ID (fk) | INT | Foreign key referencing company table |

|  |  |  |
| --- | --- | --- |
| companies | | |
| company\_ID (pk) | INT | Primary ID for companies table |
| company\_name | VARCHAR(150) | Name of insurance company |

|  |  |  |
| --- | --- | --- |
| address\_info | | |
| relation\_ID (pk) | INT | Primary ID for address info table |
| fk\_address\_ID (fk) | INT | Foreign key referencing addresses table |
| fk\_patient\_ID (fk) | INT (null option) | Foreign key referencing patients table (patient address; optional) |
| fk\_company\_ID (fk) | INT (null option) | Foreign key referencing companies table (insurance company address; optional) |
| fk\_practice\_ID (fk) | INT (null option) | Foreign key referencing practices table (practice address; optional) |
| fk\_pharmacy\_ID (fk) | INT (null option) | Foreign key referencing pharmacies table (pharmacy address; optional) |

|  |  |  |
| --- | --- | --- |
| addresses | | |
| address\_ID (pk) | INT | Primary ID for addresses table |
| address\_line\_1 | VARCHAR(50) | First line of address |
| address\_line\_2 | VARCHAR(50) (null option) | Second line of address (optional) |
| city | VARCHAR(50) | City of address |
| state | VARCHAR(50) | State of address |
| zip | VARCHAR(15) | ZIP code for address |

|  |  |  |
| --- | --- | --- |
| pharmacies | | |
| pharmacy\_ID (pk) | INT | Primary ID for pharmacies table |
| name | VARCHAR(100) | Name of pharmacy |

|  |  |  |
| --- | --- | --- |
| practices | | |
| practice\_ID (pk) | INT | Primary ID for practices table |
| practice\_name | VARCHAR(150) | Name of practice |

|  |  |  |
| --- | --- | --- |
| prescription\_info | | |
| prescription\_ID (pk) | INT | Primary ID for prescription info table |
| fk\_medication\_ID (fk) | INT | Foreign key referencing medications table (medication which was ordered) |
| fk\_pharmacy\_ID (fk) | INT | Foreign key referencing pharmacies table (pharmacy where prescription was ordered) |
| fk\_visit\_ID (fk) | INT | Foreign key referencing visits table (visit where prescription was ordered) |
| fk\_doctor\_Id (fk) | INT | Foreign key referencing doctors table (doctor who ordered prescription) |

|  |  |  |
| --- | --- | --- |
| medications | | |
| medication\_ID (pk) | INT | Primary ID for medications table |
| med\_name | VARCHAR(100) | Name of medication |

|  |  |  |
| --- | --- | --- |
| doctors | | |
| doctor\_ID (pk) | INT | Primary ID for doctors table |
| first\_name | VARCHAR(50) | Doctor’s first name |
| last\_name | VARCHAR(50) | Doctor’s first name |
| type | VARCHAR(50) | Type of doctor (ex: cardiologist) |

|  |  |  |
| --- | --- | --- |
| visit\_info | | |
| visit\_ID (pk) | INT | Primary ID for visit info table |
| fk\_patient\_ID (fk) | INT | Foreign key referencing patients table (patient who visit was for) |
| fk\_doctor\_ID (fk) | INT | Foreign key referencing doctors table (doctor who attended visit) |
| fk\_practice\_ID (fk) | INT | Foreign key referencing practices table (practice where visit was conducted) |
| fk\_invoice\_ID (fk) | INT | Foreign key referencing invoices table (invoice which was generated for visit) |
| visit\_date | DATE | Date of the visit |

|  |  |  |
| --- | --- | --- |
| invoices | | |
| invoice\_ID (pk) | INT | Primary ID for invoices table |
| amount | DOUBLE | Invoice amount (i.e., cost of visit) |
| is\_paid | TINYINT(1) | Whether or not the invoice is currently paid |

|  |  |  |
| --- | --- | --- |
| invoices\_payments | | |
| ip\_ID (pk) | INT | Primary ID for invoices-payments junction table |
| fk\_invoice\_ID (fk) | INT | Foreign key referencing invoices table |
| fk\_payment\_ID (fk) | INT | Foreign key referencing payments table |

|  |  |  |
| --- | --- | --- |
| payments | | |
| payment\_ID (pk) | INT | Primary ID for payments table |
| payment\_type | VARCHAR(50) | Form of the payment (ex: insurance, mastercard, discover) |
| amount | DOUBLE | Payment amount |
| time\_stamp | DATETIME | Time stamp (date and time) for payment |

|  |  |  |
| --- | --- | --- |
| visits\_procedures | | |
| vp\_ID (pk) | INT | Primary ID for visits-procedures junction table |
| fk\_visit\_ID (fk) | INT | Foreign key referencing visits table |
| fk\_procedure\_ID (fk) | INT | Foreign key referencing procedures table |

|  |  |  |
| --- | --- | --- |
| procedures | | |
| procedure\_ID (pk) | INT | Primary ID for procedures table |
| name | VARCHAR(100) | Name of procedure |
| description | VARCHAR(500) | Brief description of procedure |

|  |  |  |
| --- | --- | --- |
| visit\_doctor\_note | | |
| vdn\_ID (pk) | INT | Primary ID for visit\_info-doctorNotes junction table |
| fk\_visit\_ID (fk) | VARCHAR(50) | Foreign key referencing visit\_info table |
| fk\_doctorNote\_ID (fk) | VARCHAR(50) | Foreign key referencing doctorNotes table |

|  |  |  |
| --- | --- | --- |
| doctorNotes | | |
| doctorNote\_ID (pk) | INT | Primary ID for doctorNotes table |
| note | VARCHAR(5000) | Doctor’s note |

# **Closing section**

This project allowed our team to gain real hands-on experience designing a database system and user interface. I think the most important insight that our team gained is that even “simple” systems involve a lot of planning and can become very complicated very quickly. A lot of detail goes into creating the tables to ensure that all the relevant data is captured in the most efficient way. Additionally, our team learned that sometimes your first design ends up not accounting for all the relevant information or contains data that is not relevant, so you must be able to be agile and adjust the system to account for all design constraints as you move forward in the design process. Overall, our team worked effectively and efficiently to create this electronic health record information system.