DATABASE DESIGN REPORT **Team 2**

Sean Yap Yu Rong, Lim Ding Heng, Toh Yunqi Cheryl, Sun Shengran

IFS4205 Information Security Capstone Project, AY19/20, Semester 1 August 30, 2019

Overview

This project will be a prototype of a health care system. At this stage, it is not possible to achieve relationships expected within a proper health care system. Therefore, to achieve a meaningful and functional health care system, the envisioned health care system is designed to be condensed and compact, with core and essential functionalities. The system is streamlined to achieve the best user experience without neglecting core functionalities.

To give a summary, the envisioned system will rely on trusted administrators to perform registration of users, such as patients, therapists and researchers. After registration, therapists will be able to request permissions from patients to perform his medical duties. Permission requests include two subsets, either account or record permission. It will be assumed that account permissions are necessary for any therapist to access records. More information about permissions will be discussed in more details below.

Similarly to permissions, there is also a system for diagnosis. As explained above, it is not possible to produce a fully encompassing system such that patients have treatments and treatments are performed by doctors. However, we find that it is absolutely necessary in a health care system for therapists to view and set a patient's diagnosis; as such, there is a also functionality for diagnosis to be attributed to a patient to show some illness inflicted him or her for some time period. This is important for therapists to assess as patients' current illnesses are dependent on previous diagnosis. Records can be attributed to multiple diagnosis to show which records are relevant for treating which illness.

There are two databases, one "main" database and one "logging" database. The "main" database performs typical operations related to the system while the "logging" database facilitates the storing of application logs. The "logging" database will also be used to backup transactional and or query logs of the "main" database. This is shown in more detail in the EER diagram and detailed explanations below.

1 Enhanced Entity Relationship (EER) Diagram

1.1 Description

The following is the imagined EER diagram for the proposed system. For simplicity's sake, layers are used to denote separation of databases.

Note that the data-types shown in the diagram may be incorrect. In fact, some of the data-types are placeholders to be decided in the future.

1.2 Diagram

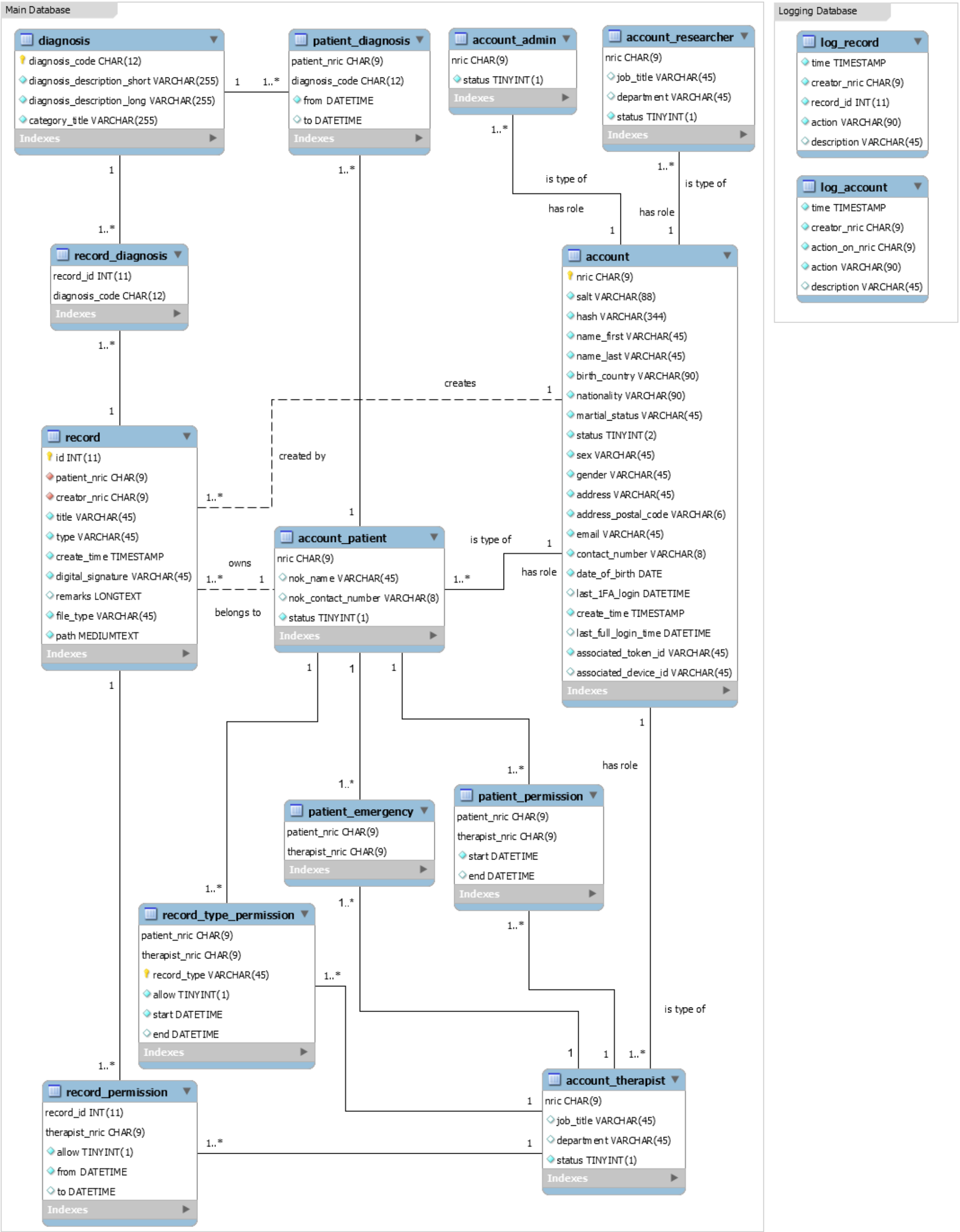

Figure 1: Enhanced Entity Relationship (EER) Diagram

2 Account Tables

Our system caters for 4 different types of accounts, namely patient account, therapist account, researcher account and admin account. An admin is a "superuser" that can register accounts, view all user information and event logs. A patient can view all his therapists, diagnoses history as well as medical records. He can also upload records and set access permissions to all his records and therapists. A therapist can view personal and medical information of his patients, edit record notes and share records with other therapists for comments (if permission granted). He can also upload medical records for his patients. A researcher can search for de-identified medical records of patients and view them for research & analysis purpose.

A user can have multiple accounts with different types, but the system only allows him to sign in as a specific role each time. If a user switches to another role that he has been registered as, a separate web page specifically designed for that role will be displayed. Common attributes (i.e. name, address, email) are stored in the **account** table, and attributes specific to the role (i.e. job title & department for therapist) are stored in the **account_role** tables.

It is important to note that in this design, the **account** table and "role" tables are not normalized as they have a 1-1 relationship (as detailed below) which means they should actually be "merged" into a single **account** table. It is purposefully designed in such a way as roles may have specific information that are only relevant to a role. Additionally, the separation of these tables help in reducing load on the database as less complex queries may be used to perform many actions.

2.1 account

This table is for storing general user information for all types of users. The **account** table has the following attributes: nric, salt, hash, name_first, name_last, birth_country, nationality, marital_status, status, status, sex, gender, address, address_postal_code, email, contact_number, date_of_birth, last_1FA_login, last_full_login_time, create_time, associated_token_id, associated_device_id.

Explanation of some important attributes:

- nric : primary key of the table, uniquely identifies a user. It is important to note that as per, Singapore law, Private Hospitals and Medical Clinics Regulations, the law requires the collection of NRIC, hence this system will be exempt from and will not require compliance to the Personal Data Protection Act (PDPA). Utilizing NRIC as a primary key will also optimize indexing of the database such that queries for data processing and such will be as efficient as possible. It is the only column the system will use to perform searches.

- salt and hash : since it is not safe to directly store the password of a user, we store salt and hashed password instead so that the system is able to check the password by combining it with the salt, hash it and compare with the hashed password. Therefore, even if the database server has been compromised, user password will not be disclosed to attacker. Also, since salt is used in the hashing, it is almost impossible for attacker to perform dictionary attack or to know which users are using the same password.

- status : user account status. Since it is represented by 2 bits, 0 (00) stands for account disabled, 1 (01) stands for account enabled with MFA, and 2 (10) stands for account enabled without MFA. If user account is disabled, the user cannot log in as any role. Otherwise the system will check the status from all account role tables to determine which roles the user can log in as.

- last_1FA_login : the most recent time that the user attempted to log in using his password (first factor authentication).

- last_full_login_time : the most recent time that the user attempted to log in using his NFC (second factor authentication). Having last_1FA_login and last_full_login_time attributes allows us to check the time interval between the two factors authentication so that the system can get alerts if the time interval is longer than a certain period.

- associated_token_id : hashed id of the near-field communicator (NFC) that is associated with the user. To log in, user scans his NFC using his phone and sends the id to the web server for authentication. Server checks whether the id matches the associated_token_id of that user.

- associated_device_id : hashed id of the mobile device that user is using for login. For the first time user logs in using his mobile device, the system will register his phone by providing an associated_device_id.

User can only log in using this mobile device since then. This is to prevent attackers from logging in using a separate device. If user wishes to change the mobile device, he will need to ask admin to deactivate (delete) the previous associated_device_id in order to register a new one.

2.2 account_admin

This table is for storing admin specific information. It has the following attributes: nric (primary key), status

2.3 account_patient

This table is for storing patient specific information. It has the following attributes: nric (primary key), nok_name, nok_contact_number, status

2.4 account_researcher

This table is for storing researcher specific information. It has the following attributes: nric (primary key), job_title, department, status

2.5 account_therapist

This table is for storing therapist specific information. It has the following attributes: nric (primary key), job_title, department, status

3 Record and Diagnosis Tables

Our system keeps all the medical records of the patients. Each record can be categorized into one specific medical type (i.e. blood pressure, heartbeat...) with a file type (i.e. .img, .docx, .pdf...). Both patient and his therapists can upload records, but all the records uploaded belong to the patient only.

Each patient may have multiple diagnoses at the same time or different periods of time. And each medical record of the patient is associated with one or more diagnoses. Therefore, it is convenient for the patient to view all the relevant records for a specific diagnosis.

Our system also maintains a list of diagnosis with different levels of description for patients & therapists reference.

3.1 record

This table is for storing patient medical records information. It has the following attributes: id (primary key), patient_nric (foreign key), creator_nric (foreign key) , title, type, create_time, digital_signature, remarks, file_type, path

Explanation of some important attributes:

- digital_signature: Contains the digital signature of the record content. It is used to check the authenticity of the viewer as well as the integrity of the record. If the record content and its signature do not match, user is not allowed to view the content.

- remarks: Contains annotations & suggestions provided by the therapist. multiple therapists can edit this note if permission granted by the patient (record owner).
- path: Contains the relative path of the record stored in the file server. This path will be obfuscated in some way to prevent attackers from seeing the real record path. It will most likely contain hashes of NRIC along with created_date.
- type and file_type: type represents the medical type of the record (i.e. heartbeat, blood pressure), while file_type represents the format of the records file (i.e. docx, pdf, img).

3.2 diagnosis

This table is for storing diagnosis information as according to the International Statistical Classification of Diseases, revision 10, (ICD-10). It has the following attributes: diagnosis_code (primary key), diagnosis_description_short, diagnosis_description_long, category_title

3.3 record_diagnosis

This table is for storing record-diagnosis association. Each record can be associated with one or more diagnoses. It has the following attributes: record_id (foreign key), diagnosis_code (foreign key)

primary key: (record_id, diagnosis_code)

3.4 patient_diagnosis

This table is for storing patient diagnoses information. Each patient may have one or more diagnoses for certain periods of time. It has the following attributes: patient_nric (foreign key), diagnosis_code (foreign key), start, end

primary key: (patient_nric, diagnosis_code, start, end)

4 Permission and Emergency Tables

Our system maintains three permission tables, namely **patient_permission** table, **record_type_permission** table and **record_permission** table. They represent the three tiered permission system that we have developed for this unique project.

patient_permission table specifies a treatment period between a patient and a therapist. This is also considered as a permission granted period when patient and therapist can view the personal information of each other. The therapist can end the treatment by setting the end time, and both parties are unable to view their personal information since then. However, patient can still see the basic information of the therapists (i.e. name, department) in order to know who have had treatments with him before, and so can the therapist.

record_type_permission table and **record_permission** table set record access permission at different levels. **record_type_permission** allows therapist to view specific type of records (i.e. blood pressure, x-ray) of the patient, while **record_permission** allows therapist to view only a specific record granted by the patient. However, **record_permission** takes deterministic role, such that if a certain record is granted access by **record_type_permission** but disallowed by **record_permission**, then the therapist do not have access to the record. Both permissions are granted to the therapist for a certain period of time. Thus, therapist is unable to view the records after the period.

For a therapist to view a record, the system will check for permissions in the following order.

1. If therapist is authorised into system; account is “active” and has therapist role.
2. And if therapist has related record to patient in table, **patient_permission**, and if the permission has not “expired”, such as treatment of patient has ended, or revoked.
3. And if therapist is given permission of “allow” to view the specific record in table, **record_type_permission**.
4. Else if therapist is given fine-grained permission of “allow” to view the specific record in table, **record_permission**.

In this system, if the therapist is given “allow” permission to view a record_type of “X” in table, **record_type_permission**, the fine-grained permission of “disallow” in table, **record_permission**, will disallow the therapist from accessing the record. In this system, a patient is able to provide total fine-grained access to therapists.

In addition, in order to handle comatose patient who is unable to manually set permissions by himself, our system also maintains a **patient_emergency** table to assign a therapist to the patient. The system also helps to check & update permissions of the therapist to ensure that he can access all the medical records of the patient. This access will be revoked once the treatment is over or the patient recovers and sets the permission manually.

4.1 record_permission

This table is for storing record access permissions of therapist for a certain period of time. It has the following attributes: record_id (foreign key), therapist_nric (foreign key), allow, start, end

primary key : (record_id, therapist_nric)

4.2 record_type_permission

This table is for storing access permissions of a specific type of patient records for a certain period of time. It has the following attributes: patient_nric (foreign key), therapist_nric (foreign key), record_type, allow, start, end

primary key : (patient_nric, therapist_nric, record_type)

4.3 patient_permission

This table is for storing patient personal information permissions of therapist for a certain period of time. It has the following attributes: patient_nric (foreign key), therapist_nric (foreign key), start, end

primary key : (patient_nric, therapist_nric)

4.4 patient_emergency

This table is for storing patient-therapist association when patient is in emergency mode (i.e. comatose). It has the following attributes: patient_nric (primary key), therapist_nric (foreign key)

5 Logging Tables

There are two types of logging tables in our system. **log_record** table contains logs of events happening on a record, while **log_account** table contains logs of events happening on an account. From the logs, we are able to know which person did what action on which record/account at when.

5.1 log_account

This table is for storing event information related to accounts. It has the following attributes: time (primary key), creator_nric (foreign key), action_on_nric (foreign key), action, description

5.2 log_record

This table is for storing event information related to records. It has the following attributes: time (primary key), creator_nric (foreign key), record_id (foreign key), action, description

-End-