Databases for developers Task 2 (10 studypoints)

Designing a Prescription reminder. Hand in for the 6th of March.

For this assignment, please write a small report with diagrams and parts of the report you deem necessary. - 20 pages maximum - won't be changed after 9:30 on the 17th

Background information.

A pharmacist has described a problem with especially elderly patients forgetting to renew their receipts for medicine, as it doesn't necessarily happen automatically, even for chronic diseases, and they don't automatically get a reminder by text or email when getting their medicine.
As seen on the official app from the pharmacy they do get some sort of help,but it is not explicitly written if they get a notification when their receipt is about to expire, which leads to them standing at the counter, unable to get their medicine.
For some cases, this is dire, and while the pharmacists won't let people die out of refusal to hand out life saving medicine such as inhalators, it would help senior citizens with quality of life if they were reminded to renew their receipts.

How will you develop software and underlying databases to accommodate this issue?

For this assignment you can either design and implement your own solution to the problem, implement the one suggested below, or design a variant.

The suggested solution/take on the approach:

The pharmacy has requested an implementation for a service that can send out reminders to the patients whose prescriptions are about to expire by mail or texts.
For sending mails in Java or Python, a free STMP server for testing is provided.

For proof of concept, it is suggested to be using dummy Gmail accounts for sending out mails in order to streamline testing rather than your personal accounts.

For this task, they have requested the use of a PostGres Database in this task, as they prefer open source software.

Deliverables/What we expect you to hand in:

Mandatory:

- A working prototype that can send emails to a patient found in the database.
- Automated emails sent out when a patient's receipt is about to expire.

A functional, normalized (3rd degree) database backend that stores the following information:

Must have:

1. Persons
2. Patients
3. Prescriptions
    - How many handouts remain
4. Doctors who prescribed them and to which patient.
5. Pharmacies

Nice to have:

When and by who the information for a patient journal was Last accessed

- Log this as you see fit.

Database requirements:

Meaningful data representing people, such as real people names, (randomly generated), is nice to have - but in order to show your database optimization, we need a rather large sample size of people in the database and their receipts.

Optional: Compare your queries, before and after optimizations (Think indexing)

Optional:

A logging system that details who has been sent out notifications for receipts about to expire and who the last pharmaceut to be in contact with the patient was.

Optional extra tasks:

Design a system architecture emphasizing security and the SOLID principles.

- Multiple user access - can others access data before you have read it?
- Transactions - make sure that you don't accidentally send reminders to people with fresh receipts.
- Transparency in who accesses what data.
- Distributed databases for multiple pharmacies - experiment with multiple schemas is advised.

For clarity - 2 to 3 schemas to represent more than one data entity is sufficient.

Tips:

This is a complex task and you are expected to implement a program that gives insight in the different information. This means, actually code.

We don’t expect a fully fledged prototype, but we do expect at least a functional database or working functionalities.

We are mostly concerned with your “why” argumentation and why your solution is designed the way it is.