

Introduction

I have looked into your system and I identified few things that you need to address.

First your system is a web based applications helps in interaction between the patient and the medical officers and also other patients .some of the previous flows that you have highlighted in the previous system include:

- miss of drugs by patients
- stigmatization from other people
- long queue to take a drug

based on this few stuff, you wanted a system that will be user friendly and secure also.

The main study area of your system is data encryption and security.

System requirements

the system is expected to have three components.

- The master/admin modules
- the medical officer module
- the patient module

the master/admin module

the master/admin module is responsible for the enter operation of the system,thus some of the functionality of the admin module include:

- edit/delete/update patient details
- update drug availability
- update various places to pick up the drug
- edit/delete/update medical officer information

the medical officer module

the medical officer will be the one interacting with the patient most of the time thus his module will be responsible with the following functions:

- update patient information
- check on drug availability and update them
- check on drug pick up station and update it
- schedule and create appointment
- create reports of the system
- update patient financial status

the patient module

the patient module will have the following functions

- update patient details
- check on drugs availability
- book patient visits]
- view own patient reporting

Security

we are also aware that you needed to integrate security into your project . So it will be basically ideal to talk about it and how we are thinking to go about it.

The major security issues when related to web applications include:

1. secret keys
2. sql injection
3. cross-site request forgery

4. session hijacking
5. logs

first I need to say that ill be using django for the development of the online ticket reservation system so while tackling the above security issues ill be relating them to the django application.

- **Secret keys**

the secret key is very essential in django as it is used to hash the system by hashing I mean developing algorithm that encrypt things such as passwords etc. in django under the project under settings we have a SECRET_KEY variable which holds the secret key used to hash passwords, thus it is recommended to store this secret key in the environment variables which can not be easily accessed by hackers.

Other functions of secret key in django apart from hashing passwords are:

- used in hashing sessions
- used in hashing messages
- used in cryptography signing.

For understanding more on secret key check this YouTube video <https://www.youtube.com/watch?v=WvjOqx2nVgk>

- **SQL injection**

SQL injection is a web vulnerability that allows an attacker to interfere with the queries that an application makes into the database.

For more information on how to go about the sql injection in django you can check the following youtube video <https://www.youtube.com/watch?v=8uMbSooxSr8>

- **cross-site request forgery**

cross -site forgery also known as (CSRF) is a web vulnerability the allows attackers to induce users to perform actions that they do not intend to perform. It allows attackers to partially circumvent the same origin policy, which is designed to prevent different website from interacting with each other.

More on how to secure form with CSRF token in django can be found in this video :

<https://www.youtube.com/watch?v=iJmezMBJqEs>

- **session hijacking**

session hijacking sometime known as cookie hijacking is the exploitation of valid computer session to gain unauthorized access to information or services in computer system.

More about session hijacking can be found on this video <https://www.youtube.com/watch?v=RNEiDrppnzw>

- **logs**

logs can be used by hackers to identify system vulnerability and use those vulnerability to be able to be able to exploit and destroy a given system.

To prevent logs from showing up in django application we need to disable debugging in settings. That is DEBUG = FALSE

class diagrams

