**Patient Data Encryption and Storage**

**Introduction**

This document outlines the implementation of a patient data encryption and storage system using the Fernet cryptography library and object-oriented programming principles. The system aims to securely store patient data while maintaining its confidentiality and integrity.

**System Overview**

The system comprises two primary components:

**PatientDataEncryption:** This class handles the encryption and decryption of patient data using the Fernet cipher. It provides methods for encrypting data into a secure format and decrypting it back to its original form.

**PatientDatabase:** This class manages the storage and retrieval of patient data. It maintains a list of patients, each represented as a dictionary containing their personal and medical information.

**Detailed Description**

**PatientDataEncryption Class**

The PatientDataEncryption class encapsulates the logic for encrypting and decrypting patient data. It utilizes the Fernet cryptography library to generate a secure key and encrypt data using the generated key.

**Constructor:**

Python

**def \_\_init\_\_(self, key=None):**

**if key is None:**

**self.key = Fernet.generate\_key()**

**else:**

**self.key = key**

**self.cipher = Fernet(self.key)**

The constructor initializes the encryption class with a Fernet key. If no key is provided, a new key is generated. The key is used to create a Fernet cipher object for encryption and decryption operations.

**Encryption Method:**

Python

**def encrypt\_data(self, data):**

**encrypted\_data = self.cipher.encrypt(json.dumps(data).encode())**

**return encrypted\_data**

The encrypt\_data method encrypts the provided data using the Fernet cipher. The data is first converted to JSON format and then encoded into bytes. The encoded data is encrypted using the cipher and returned as encrypted bytes.

**Decryption Method:**

Python

**def decrypt\_data(self, encrypted\_data):**

**decrypted\_data = json.loads(self.cipher.decrypt(encrypted\_data).decode())**

**return decrypted\_data**

The decrypt\_data method decrypts the provided encrypted data using the Fernet cipher. The encrypted data is decrypted using the cipher, and the resulting decrypted bytes are decoded and converted back to a JSON object. The JSON object is returned as the decrypted data.

**PatientDatabase Class**

The PatientDatabase class handles the storage and retrieval of patient data. It maintains a list of patients, each represented as a dictionary containing their personal and medical information.

**Constructor:**

Python

**def \_\_init\_\_(self):**

**self.patients = []**

The constructor initializes the patient database with an empty list of patients.

**Add Patient Method:**

Python

**def add\_patient(self, patient\_data):**

**self.patients.append(patient\_data)**

The add\_patient method adds a new patient to the database. The provided patient data is a dictionary containing the patient's information. The dictionary is appended to the list of patients.

**Display Patients Method:**

Python

**def display\_patients(self):**

**print("Patients in the Database:")**

**for patient in self.patients:**

**print(patient)**

**print("\n")**

The display\_patients method prints a list of all the patients stored in the database. Each patient's information is printed as a dictionary.

**Example Usage**

The provided code snippet demonstrates the usage of the PatientDataEncryption and PatientDatabase classes to encrypt and store patient data, retrieve and decrypt it, and display the decrypted data.

**Conclusion**

The implemented system effectively encrypts patient data using the Fernet cryptography library and stores it securely in a patient database. The system also retrieves and decrypts the encrypted data, ensuring patient data confidentiality and integrity.