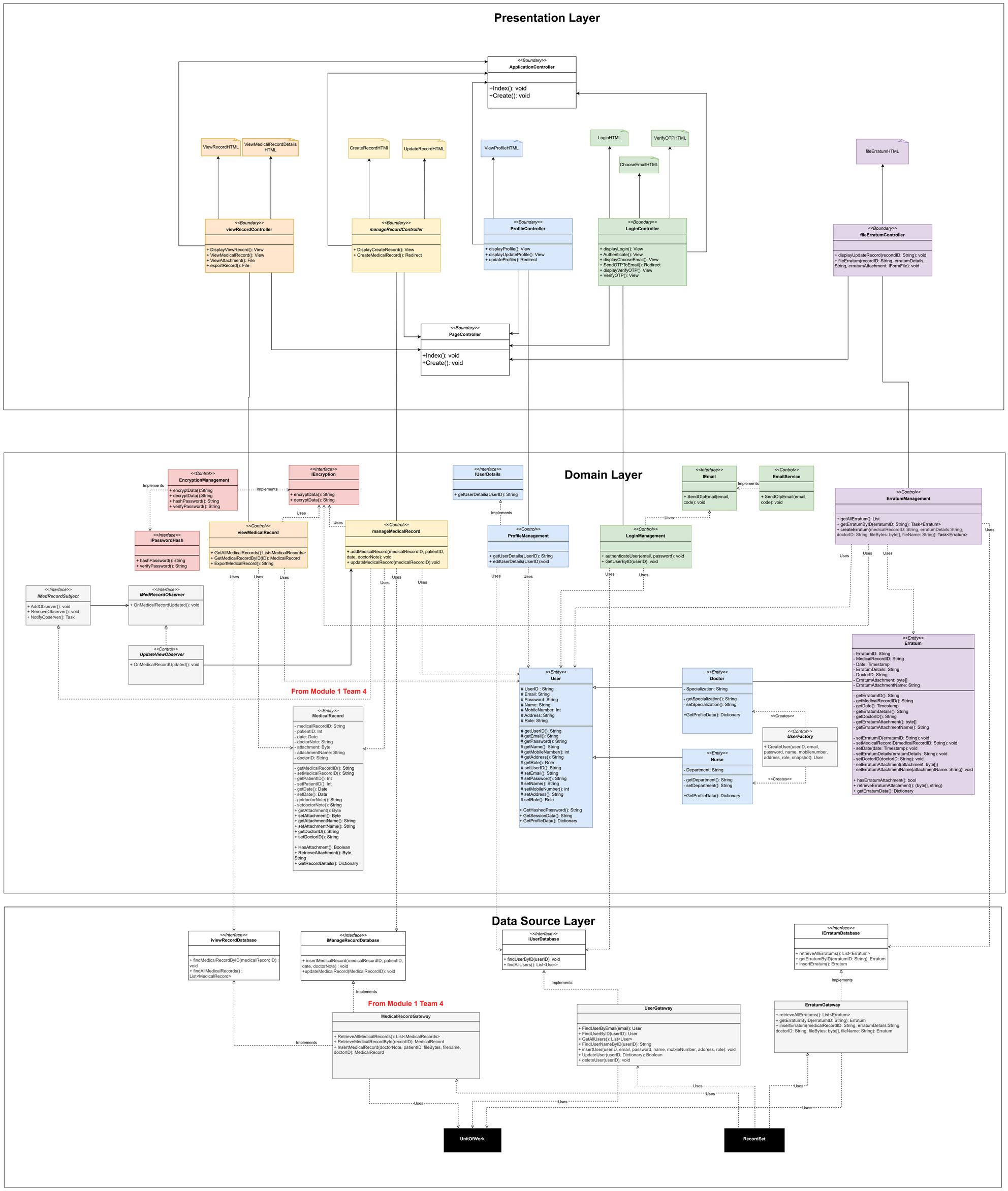
Presentation Layer
<<Boundary>>
ApplicationController
+Index(): void
+Create(): void
ViewRecordHTML
ViewMedicalRecordDetails HTML
CreateRecordHTMl
UpdateRecordHTML
ViewProfileHTML
LoginHTML
VerifyOTPHTML
ChooseEmailHTML
fileErratumHTML
<<Boundary>>
viewRecordController
+ DisplayViewRecord(): View
+ ViewMedicalRecord(): View
+ ViewAttachment(): File
+ exportRecord(): File
<<Boundary>>
manageRecordController
+ DisplayCreateRecord(): View
+ CreateMedicalRecord(): Redirect
<<Boundary>>
ProfileController
+ displayProfile(): View
+ displayUpdateProfile(): View
+ updateProfile(): Redirect
<<Boundary>>
LoginController
+ displayLogin(): View
+ Authenticate(): View
+ displayChooseEmail(): View
+ SendOTPToEmail(): Redirect
+ displayVerifyOTP(): View
+ VerifyOTP(): View
<<Boundary>>
fileErratumController
+ displayUpdateRecord(recortdID: String): void
+ fileErratum(recordID: String, erratumDetails: String, erratumAttachment: IFormFile): void
<<Boundary>>
PageController
+Index(): void
+Create(): void
Domain Layer
<<Control>>
EncryptionManagement
+ encryptData():String
+ decryptData():String
+ hashPassword(): String
+ verifyPassword(): String
Implements
<<Interface>>
IEncryption
+ encryptData(): String
+ decryptData(): String
<<Interface>>
IUserDetails
+ getUserDetails(UserID): String
<<Interface>>
IEmail
+ SendOtpEmail(email, code): void
<<Control>>
EmailService
+ SendOtpEmail(email, code): void
<<Control>>
ErratumManagement
+ getAllErratum(): List
+ getErratumByID(erratumID: String): Task<Erratum>
+ createErratum(medicalRecordID: String, erratumDetails:String, doctorID: String, fileBytes: byte[], fileName: String): Task<Erratum>
<<Interface>>
IPasswordHash
+ hashPassword(): string
+ verifyPassword(): String
<<Control>>
viewMedicalRecord
+ GetAllMedicalRecords():List<MedicalRecords>
+ GetMedicalRecordByID(ID): MedicalRecord
+ ExportMedicalRecord(): String
<<Control>>
manageMedicalRecord
+ addMedicalRecord(medicalRecordID, patientID, date, doctorNote): void
+ updateMedicalRecord(medicalRecordID):void
<<Control>>
ProfileManagement
+ getUserDetails(UserID): String
+ editUserDetails(UserID):void
<<Control>>
LoginManagement
+ authenticateUser(email, password): void
+ GetUserByID(userID): void
Uses
<<Interface>>
IMedRecordSubject
+ AddObserver(): void
+ RemoveObserver(): void
+ NotifyObserver(): Task
<<Interface>>
IMedRecordObserver
+ OnMedicalRecordUpdated(): void
<<Control>>
UpdateViewObserver
+ OnMedicalRecordUpdated(): void
From Module 1 Team 4
<<Entity>>
Erratum
- ErratumID: String
- MedicalRecordID: String
- Date: Timestamp
- ErratumDetails: String
- DoctorID: String
- ErratumAttachment: byte[]
- ErratumAttachmentName: String
- getErratumID(): String
- getMedicalRecordID(): String
- getDate(): Timestamp
- getErratumDetails(): String
- getDoctorID(): String
- getErratumAttachment(): byte[]
- getErratumAttachmentName(): String
- setErratumID(erratumID: String): void
- setMedicalRecordID(medicalRecordID: String): void
- setDate(date: Timestamp): void
- setErratumDetails(erratumDetails: String): void
- setDoctorID(doctorID: String): void
- setErratumAttachment(attachment: byte[])
- setErratumAttachmentName(attachmentName: String): void
+ hasErratumAttachment(): bool
+ retrieveErratumAttachment(): (byte[], string)
+ getErratumData(): Dictionary
<<Entity>>
User
UserID : String
Email: String
Password: String
Name: String
MobileNumber: Int
Address: String
Role: String
getUserID(): String
getEmail(): String
getPassword(): String
getName(): String
getMobileNumber(): int
getAddress(): String
getRole(): Role
setUserID(): String
setEmail(): String
setPassword(): String
setName(): String
setMobileNumber(): int
setAddress(): String
setRole(): Role
+ GetHashedPassword(): String
+ GetSessionData(): String
+ GetProfileData(): Dictionary
<<Entity>>
Doctor
- Specialization: String
- getSpecialization(): String
- setSpecialization(): String
+GetProfileData(): Dictionary
<<Entity>>
Nurse
- Department: String
- getDepartment(): String
- setDepartment(): String
+GetProfileData(): Dictionary
<<Creates>>
<<Control>>
UserFactory
+ CreateUser(userID, email, password, name, mobilenumber, address, role, snapshot): User
<<Entity>>
MedicalRecord
- medicalRecordID: String
- patientID: Int
- date: Date
- doctorNote: String
- attachment: Byte
- attachmentName: String
- doctorID: String
- getMedicalRecordID(): String
- setMedicalRecordID(): String
- getPatientID(): Int
- setPatientID(): Int
- getDate(): Date
- setDate(): Date
- getdoctorNote(): String
- setdoctorNote(): String
+ getAttachment(): Byte
+ setAttachment(): Byte
+ getAttachmentName(): String
+ setAttachmentName(): String
+ getDoctorID(): String
+ setDoctorID(): String
+ HasAttachment(): Boolean
+ RetrieveAttachment(): Byte, String
+ GetRecordDetails(): Dictionary
Data Source Layer
<<Interface>>
iviewRecordDatabase
+ findMedicalRecordByID(medicalRecordID) : void
+ findAllMedicalRecords() : List<MedicalRecord>
<<Interface>>
iManageRecordDatabase
+ insertMedicalRecord(medicalRecordID, patientID, date, doctorNote) : void
+updateMedicalRecord(MedicalRecordID): void
<<Interface>>
iUserDatabase
+ findUserByID(userID): void
+ findAllUsers(): List<User>
<<Interface>>
iErratumDatabase
+ retrieveAllErratums(): List<Erratum>
+ getErratumByID(erratumID: String): Erratum
+ insertErratum(): Erratum
Implements
From Module 1 Team 4
MedicalRecordGateway
+ RetrieveAllMedicalRecords(): List<MedicalRecords>
+ RetrieveMedicalRecordById(recordID): MedicalRecord
+ InsertMedicalRecord(doctorNote, patientID, fileBytes, filename, doctorID): MedicalRecord
UserGateway
+ FindUserByEmail(email): User
+ FindUserByID(userID): User
+ GetAllUsers(): List<User>
+ FindUserNameByID(userID): String
+ insertUser(userID, email, password, name, mobileNumber, address, role): void
+ UpdateUser(userID, Dictionary): Boolean
+ deleteUser(userID): void
ErratumGateway
+ retrieveAllErratums(): List<Erratum>
+ getErratumByID(erratumID: String): Erratum
+ insertErratum(medicalRecordID: String, erratumDetails:String, doctorID: String, fileBytes: byte[], fileName: String): Erratum
UnitOfWork
RecordSet