

Stores

```
StaffDataStore
- staffList: Map<String, Staff>

+ loadStaffFromCSV(csvFilePath: String): void
+ writeStaffToCSV(csvFilePath: String): void
+ addOrUpdateStaff(staff: Staff): void
+ removeStaff(staffId: String): void
+ getStaffList(): Map<String, Staff>
```

```
InventoryDataStore
- inventoryList: List<Inventory>

+ loadInventoryFromCSV(): void
+ getInventoryList(): List<Inventory>
+ setInventoryList(inventoryList: List<Inventory>): void
+ addInventory(inventory: Inventory): void
+ writeInventoryToCSV(inventoryCsvPath: String): void
+ getLowStockInventory(): List<Inventory>
```

Views

```
PendingAppointmentRequestView
- appointmentRequestService: AppointmentRequestService

+ PendingAppointmentRequestView(scheduleService: ScheduleService, appointmentService: AppointmentService)
+ displayPendingRequests(doctor: Doctor): void
```

```
PatientMedicalRecordView
- medicalRecordService: MedicalRecordService

+ PatientMedicalRecordView(medicalRecordService: MedicalRecordService)
+ printMedicalRecords(patientID: String): void
```

```
AppointmentOutcomeRecordView
- csvFilePath: String

+ loadAndPrintAppointments(): void
+ loadAndPrintPendingAppointments(): boolean
```

```
AdminView
- scanner: Scanner
- adminController: AdministratorController
- adminService: ProjectAdminService

+ AdminView(adminController: AdministratorController, adminService: ProjectAdminService)
+ displayMenu(): void
+ displayListOfStaff(staffList: List<Staff>): void
+ displayAppointments(appointments: List<Appointment>): void
+ displayInventory(inventoryList: List<Inventory>): void
+ getMenuChoice(): int
+ getStaffIDForRemoval(): String
+ getNewMedicationDetails(): Inventory
+ getMedicineNameForRemoval(): String
+ getMedicineName(): String
+ getMedicineQuantity(): int
+ getLowStockAlertLevel(): int
+ displayChangePassword(hospitalID: String): void
+ getRequestIdForReplenishment(): String
```

```
ViewAndManageInventory
- scanner: Scanner
- inventoryService: InventoryService

+ ViewAndManageInventory(inventoryService: InventoryService)
+ displayMenu(): void
+ getMenuChoice(): int
+ getNewMedicationDetails(): Inventory
+ addNewMedication(): void
+ removeMedication(): void
+ updateStockLevel(): void
+ updateLowStockAlert(): void
+ displayInventory(inventoryList: List<Inventory>): void
+ viewInventory(): void
+ displayListOfStaff(staffList: List<Staff>): void
+ displayAppointments(appointments: List<Appointment>): void
+ getMedicineNameForRemoval(): String
+ getMedicineName(): String
+ getMedicineQuantity(): int
+ getLowStockAlertLevel(): int
```

```
ManageHospStaff
- scanner: Scanner

+ displayMenu(): void
+ getMenuChoice(): int
+ getStaffDetails(): Staff
+ displayListOfStaff(staffList: List<Staff>): void
+ displayAppointments(appointments: List<Appointment>): void
+ displayInventory(inventoryList: List<Inventory>): void
```

```
AppointmentsView
- scanner: Scanner
- appointmentService: IAppointmentService

+ AppointmentsView(appointmentService: IAppointmentService)
+ viewScheduledAppointments(): void
+ displayAppointments(appointments: List<Appointment>): void
+ displayListOfStaff(staffList: List<Staff>): void
+ displayInventory(inventoryList: List<Inventory>): void
+ displayMenu(): void
+ getMenuChoice(): int
```

```
IIPharmacistView
+ displayMenu(): void
+ viewAppointmentOutcomeRecords(): void
```

```
PharmacistView
- scanner: Scanner
- pharmacistController: PharmacistController
- pharmacistService: PharmacistService
- userService: UserService

+ PharmacistView(pharmacistController: PharmacistController, pharmacistService: PharmacistService, userService: UserService)
+ start(user: User): void
+ registerPharmacist(user: User): Pharmacist
+ updatePrescriptionStatus(): void
+ viewAppointmentOutcomeRecords(): void
+ displayMenu(): void
+ getUserInput(): int
+ changePassword(): void
```

```
MedicalRecordView
+ displayPatientInfo(appointment: Appointment)
```

```
AllocatedAppointmentView
+ displayPatientInfo(appointment: Appointment)
```

```
AppointmentHistoryView
+ displayPatientInfo(appointment: Appointment)
```

```
PatientView
+ showPatientDetails(patient: Patient): void
+ showSuccessMessage(message: String): void
+ showErrorMessage(message: String): void
+ display(patient: Patient): void
```

```
PersonalScheduleView
- scheduleService: ScheduleService

+ PersonalScheduleView(scheduleService: ScheduleService)
+ displayPersonalSchedule(doctor: Doctor): void
```

```
SubmittedRequestView
- pharmacistView: InventoryService

+ SubmittedRequestView(pharmacistView: InventoryService)
+ displayRequests(pharmacist: Pharmacist): void
```

```
UpcomingAppointmentView
- scheduleService: ScheduleService

+ UpcomingAppointmentsView(scheduleService: ScheduleService)
+ displayUpcomingAppointments(doctor: Doctor): void
```

```
DoctorView
+ displayMenu(): void
+ showSuccessMessage(message: String): void
+ showErrorMessage(message: String): void
```

```
DoctorView
- scanner: Scanner
- doctorController: DoctorController
- userService: UserService
- appointmentRequestService: AppointmentRequestService
- scheduleService: ScheduleService
- medicalRecordService: MedicalRecordService
- appointmentService: AppointmentService
- doctorService: DoctorService

+ DoctorView(doctorController: DoctorController, doctorService: DoctorService, userService: UserService, scheduleService: ScheduleService, appointmentService: AppointmentService)
+ start(user: User): void
+ displayMenu(): void
+ manageAvailabilityMenu(user: User): void
+ manageAppointmentRequestsMenu(doctor: Doctor): void
+ managePatientRecordsMenu(): void
+ getUserInput(): int
+ showSuccessMessage(message: String): void
+ showErrorMessage(message: String): void
+ changePassword(): void
```

```
UIView
- userService: UserService
- loggedInHospitalID: String
- WELCOME_MESSAGE: static final String
- SEPARATOR: static final String

+ UIView(userService: UserService)
+ displayLogin(): void
+ attemptLogin(scanner: Scanner): boolean
+ promptToExitOrRetry(scanner: Scanner): boolean
+ navigateToRoleSpecificPage(user: User, role: UserRole): void
+ navigateToPatientPage(user: User): void
+ navigateToPharmacistPage(user: User): void
+ navigateToDoctorPage(user: User): void
+ navigateToAdministratorPage(user: User): void
+ displayChangePassword(): void
```

```
MedicalInventoryView
- inventoryService: InventoryService

+ MedicalInventoryView(inventoryService: InventoryService)
+ displayInventory(inventoryList: List<Inventory>): void
+ displayInventory(inventoryList: List<Inventory>): void
+ showErrorMessage(message: String): void
+ showSuccessMessage(message: String): void
```

Entry point of application

```
HMS.App
+ main()
```

State that is up to date with the data files

Controls the users' actions according to their input. Uses Services and Views through the interfaces to control the user experience and the flow of the application.

```
ReplenishmentController
- service: ReplenishmentService

+ ReplenishmentController()
+ sendReplenishmentRequest(medicines: List<String>): List<ReplenishmentRequest>
+ viewPendingRequests(): List<ReplenishmentRequest>
+ viewPastRequests(): List<ReplenishmentRequest>
+ viewPendingRequestsAdmin(): List<ReplenishmentRequest>
+ approveRequest(requestId: String): void
+ rejectRequest(requestId: String): void
```

```
AppointmentController
- appointmentService

+ AppointmentController()
+ scheduleAppointment(appointment: Appointment): boolean
+ cancelAppointment(appointmentId: String): boolean
+ rescheduleAppointment(appointmentId: String, newAppointment: Appointment): boolean
+ viewScheduledAppointments(): List<Appointment>
+ getAppointment(appointmentId: String): Appointment
+ recordAppointmentOutcome(appointmentId: String, serviceProvided: String, prescribedMedications: List<Medication>, prescribedQuantities: List<Integer>, consultationNotes: String): void
+ getAvailableSlots(doctorId: String, date: LocalDate): List<LocalDateTime>
+ updateMedicationStatus(appointmentId: String): void
```

```
DoctorController
- doctorService: DoctorService
- appointmentService: AppointmentService
- scheduleService: ScheduleService
- appointmentRequestService: AppointmentRequestService
- medicalRecordService: MedicalRecordService
- userService: UserService
- medicalRecordsView: PatientMedicalRecordView
- viewPendingAppointmentRequest: PendingAppointmentRequestView
- personalScheduleView: PersonalScheduleView
- viewUpcomingAppointments: UpcomingAppointmentsView
- appointmentOutcomeRecordView: AppointmentOutcomeRecordView

+ DoctorController(doctorService: DoctorService, scheduleService: ScheduleService, medicalRecordService: MedicalRecordService, appointmentService: AppointmentService)
+ pendingAppointmentsView(doctor: Doctor): void
+ personalScheduleView(doctor: Doctor): void
+ upcomingAppointmentsView(doctor: Doctor): void
+ acceptAppointmentRequest(doctor: Doctor): void
+ declineAppointmentRequest(doctor: Doctor): void
+ setToUnavailable(doctor: Doctor): void
+ setToUnavailable(doctor: Doctor): void
+ blockSlots(doctor: Doctor): void
+ unblockSlots(doctor: Doctor): void
+ newPatientDiagnosis(): void
+ newPatientPrescription(): void
+ appointmentOutcomeRecordView(doctor: Doctor): void
+ viewAppointmentOutcomeRecords(doctor: Doctor): boolean
+ changePassword(hospitalID: String, oldPassword: String, newPassword: String): void
```

```
UserController
- userService: UserService
- userView: UIView

+ UserController(userService: UserService)
+ run(): void
+ attemptLogin(scanner: Scanner): boolean
+ navigateToRoleSpecificPage(user: User, role: UserRole): void
+ navigateToPatientPage(user: User): void
+ navigateToPharmacistPage(user: User): void
+ navigateToDoctorPage(user: User): void
+ changePassword(hospitalID: String, oldPassword: String, newPassword: String): boolean
```

```
AdministratorController
- adminService: ProjectAdminService
- appointmentService: AppointmentService
- adminView: AdminView
- inventoryService: InventoryService
- userService: UserService
- replenishmentService: ReplenishmentService
- loggedInHospitalID: String

+ AdministratorController(appointmentService: AppointmentService, adminService: ProjectAdminService, userService: UserService, patientService: PatientService)
+ start(loggedInHospitalID: String): void
+ manageHospitalStaff(): void
+ managePatients(): void
+ manageInventory(): void
+ manageAppointments(): void
+ changePassword(hospitalID: String, oldPassword: String, newPassword: String): boolean
+ viewPendingReplenishmentRequests(): void
+ declineReplenishmentRequest(requestId: String): void
```

```
PatientController
- patientService: PatientService
- appointmentService: AppointmentService
- allocatedAppointmentView: AllocatedAppointmentView
- appointmentHistoryView: AppointmentHistoryView
- medicalRecordView: MedicalRecordView

+ viewPatientDetails(patient: Patient): void
+ viewAllocatedAppointments(patient: Patient): void
+ viewAppointmentHistory(patient: Patient): void
+ viewMedicalRecords(patient: Patient): void
+ updateContactInformation(patient: Patient): void
+ changePassword(patient: Patient): boolean
+ createAppointment(patient: Patient): void
+ cancelAppointment(patient: Patient): void
+ rescheduleAppointment(patient: Patient): void
+ getHospitalID(): String
+ viewPastRecords(patient: Patient): void
```

```
PharmacistController
- inventoryService: InventoryService
- appointmentService: AppointmentService
- replenishmentService: ReplenishmentService
- userService: UserService
- appointmentOutcomeRecordView: AppointmentOutcomeRecordView
- csvFile: String

+ PharmacistController(inventoryService: InventoryService, appointmentService: AppointmentService, replenishmentService: ReplenishmentService, userService: UserService)
+ viewMedicationInventory(): void
+ submitReplenishmentRequest(): void
+ viewReplenishmentRequests(): void
+ updatePrescription(): void
+ viewAppointmentOutcomeRecords(): void
+ changePassword(): void
```

```
BillingController
+ BillingController()
+ calculateBill(appointmentId: String): String
+ payBill(appointmentId: String): boolean
+ getPendingPayments(allAppointments: List<Appointment>): List<Appointment>
```

Models

```
Staff
- id: String
- name: String
- role: String
- gender: String
- age: int

+ Staff(id: String, name: String, role: String, gender: String, age: int)
```

```
User
- hospitalID: String
- password: String = password
- role: UserRole

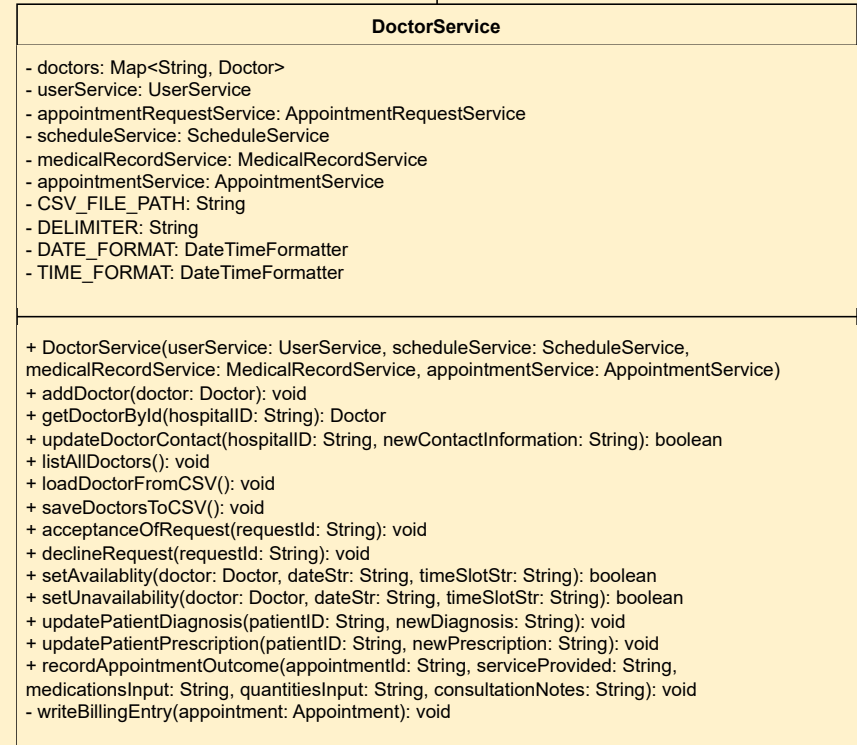
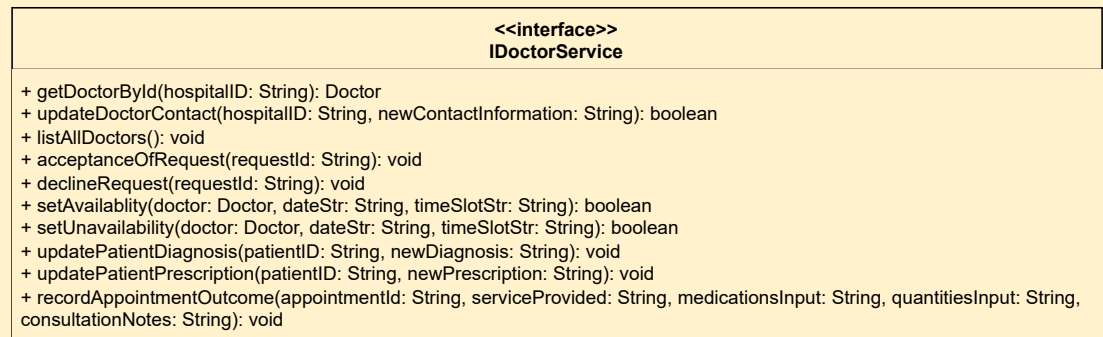
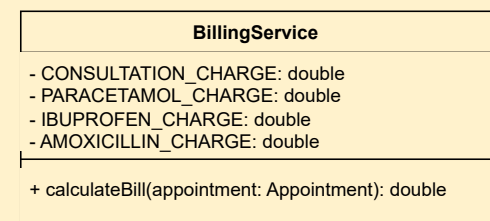
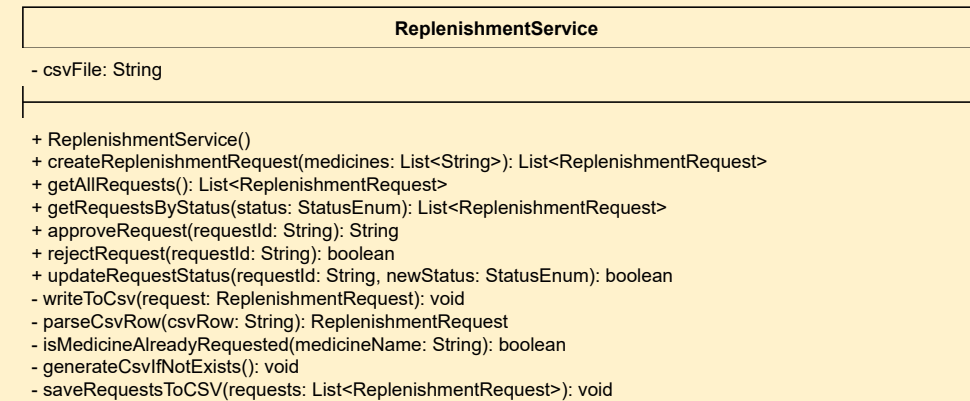
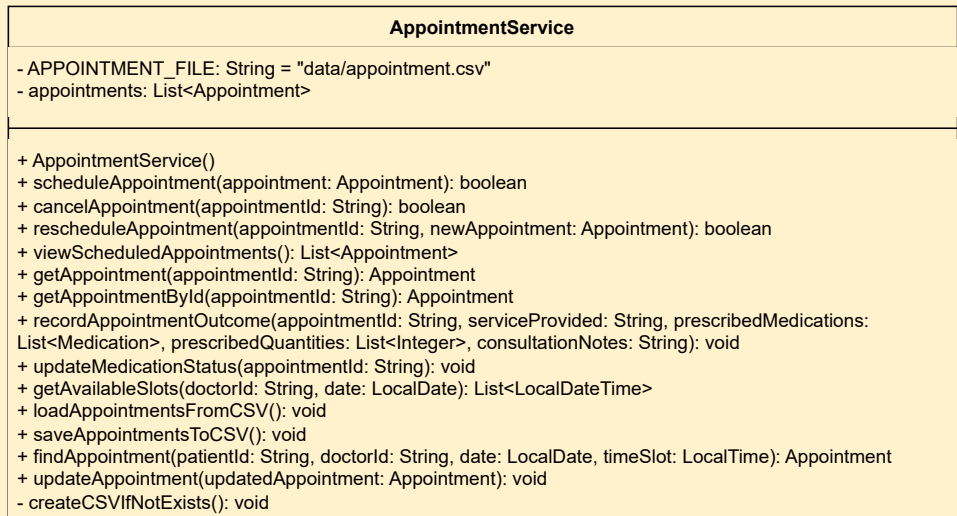
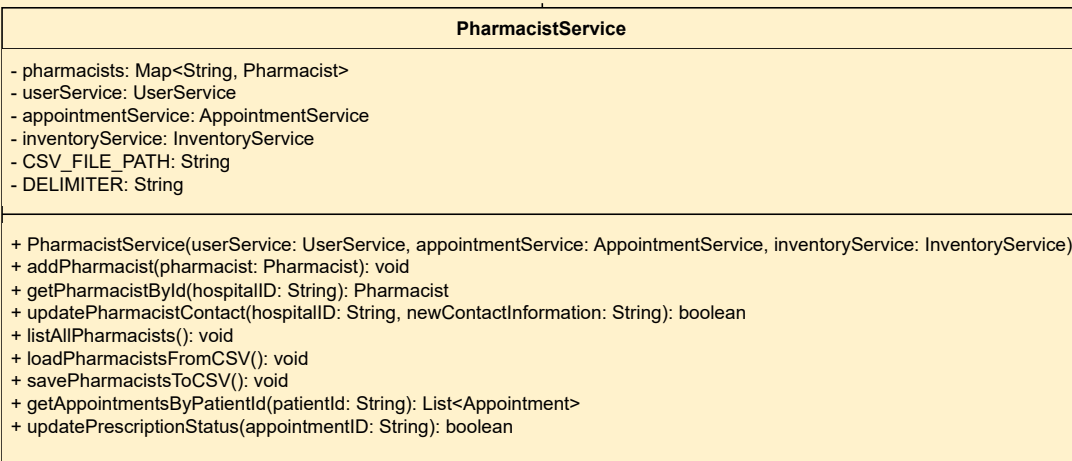
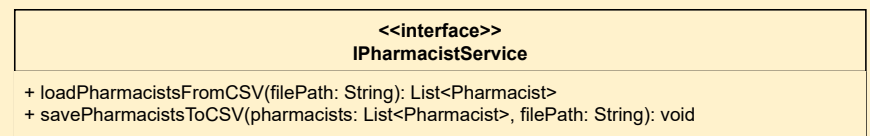
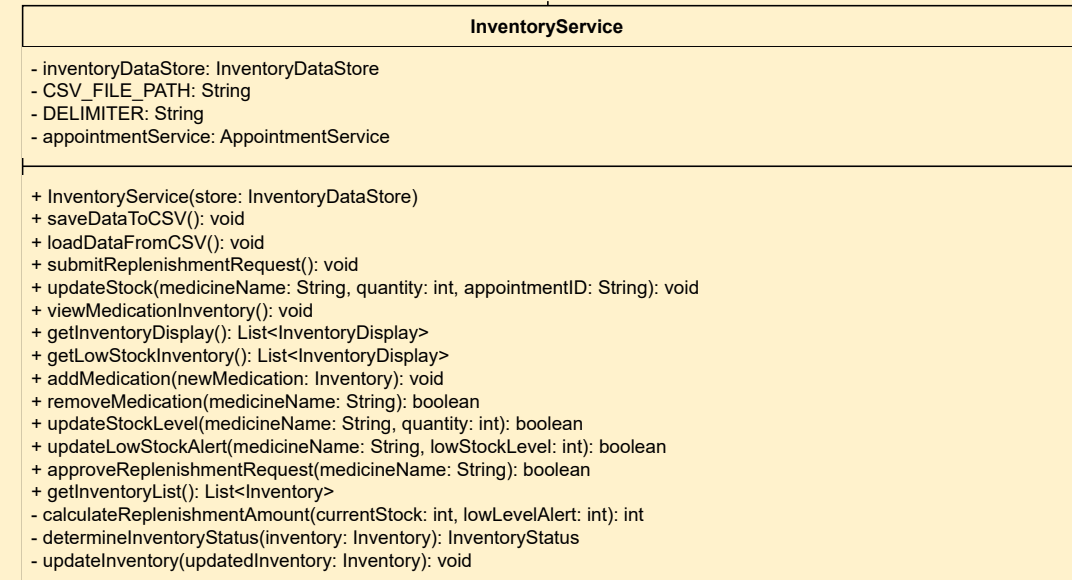
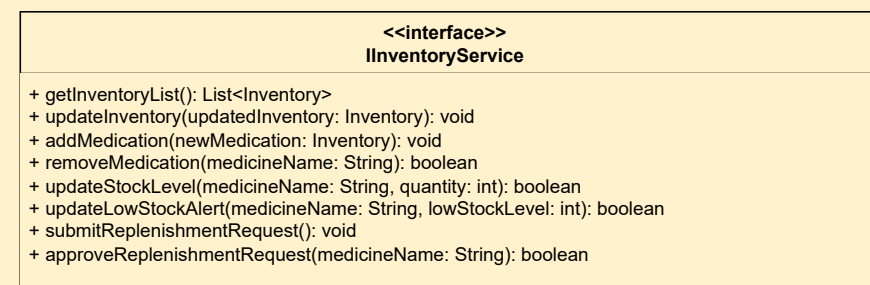
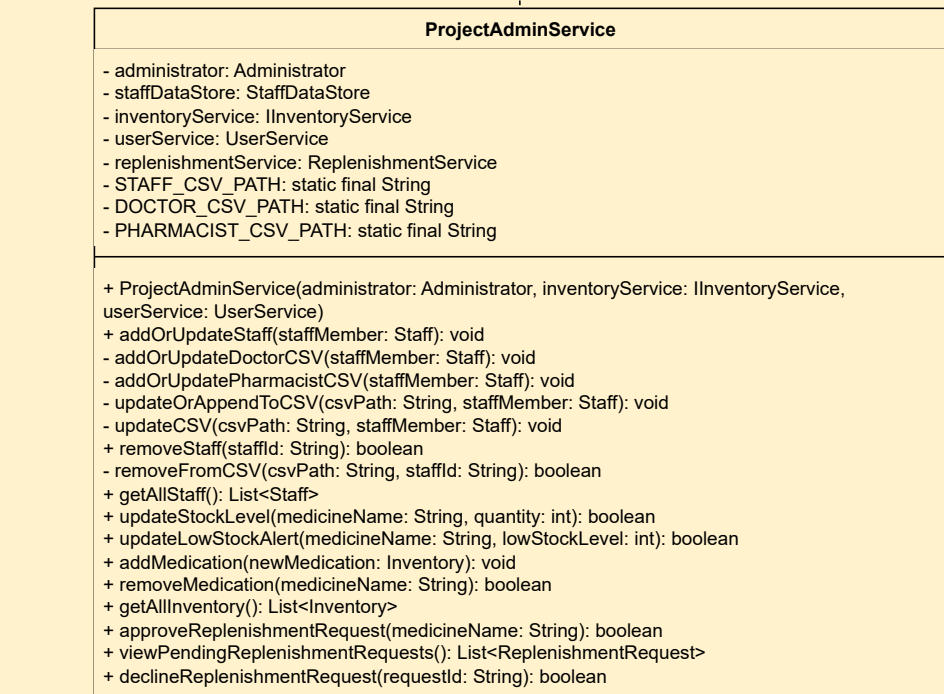
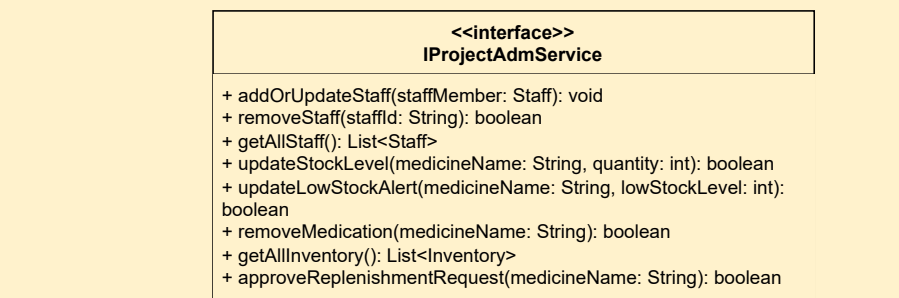
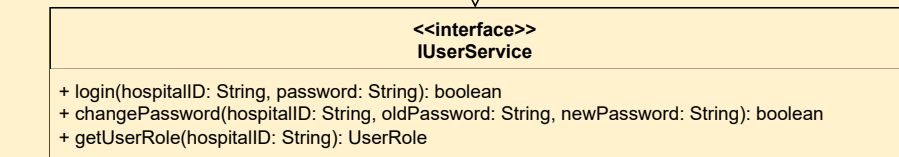
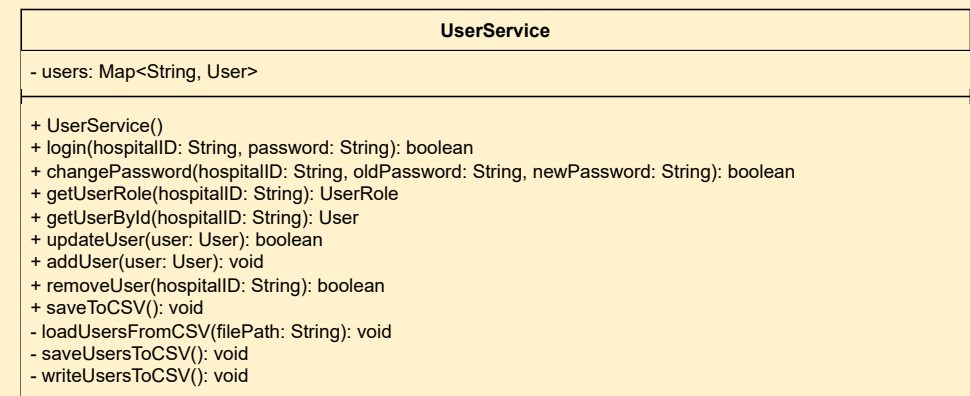
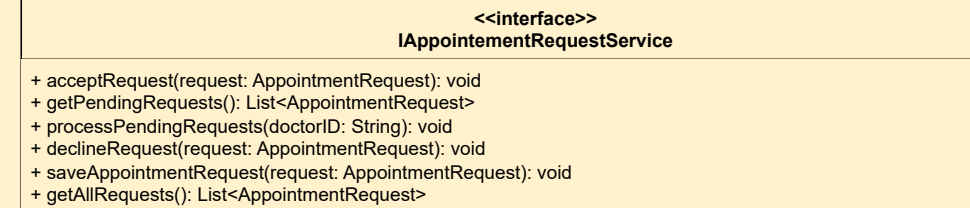
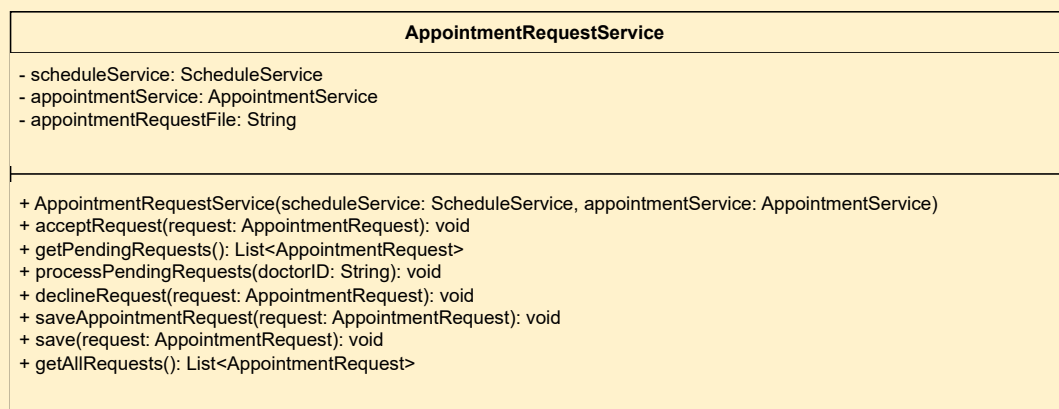
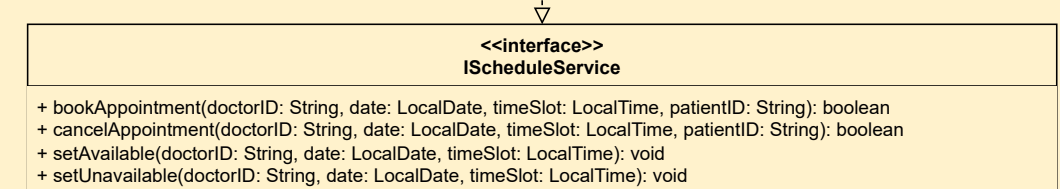
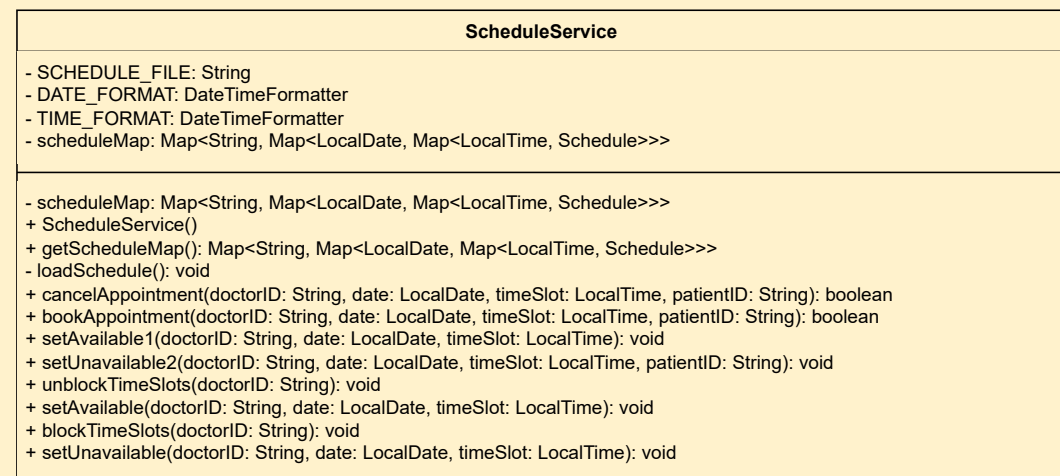
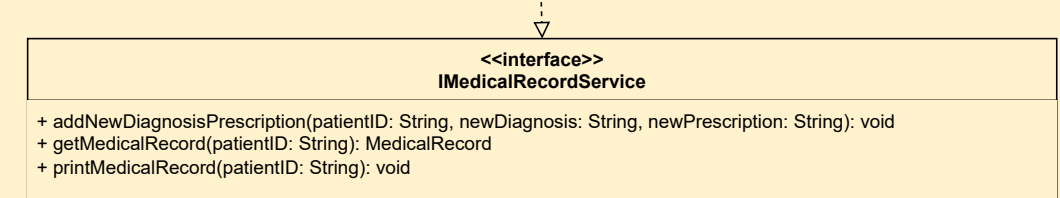
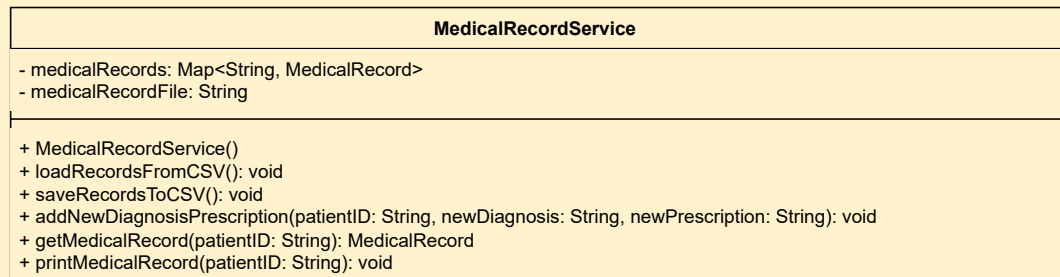
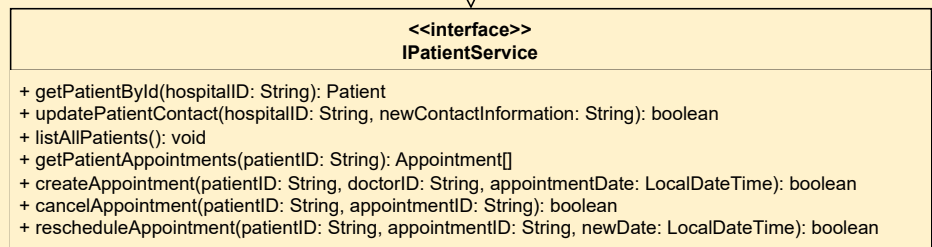
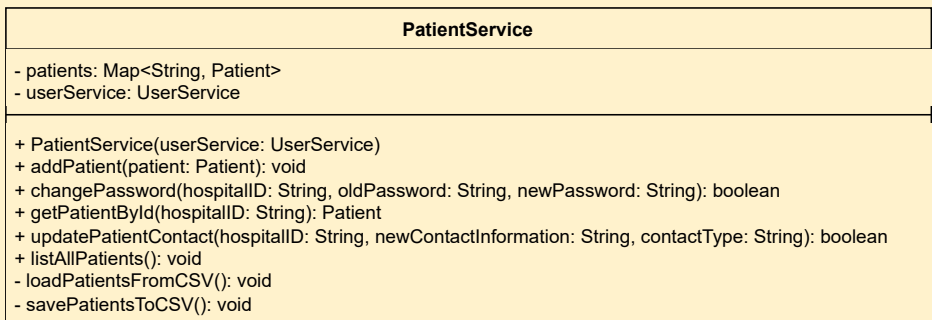
+ User(hospitalID: String, password: String = Package.User.password, name: String, email: String)
+ getHospitalID(): String
+ setHospitalID(): boolean
```

```
Administrator
- hospitalStaff: HashMap<String, Staff>

+ Administrator(hospitalID: String, password: String, role: String)
+ getHospitalStaff(): HashMap<String, Staff>
+ setHospitalStaff(hospitalStaff: HashMap<String, Staff>): void
```


Controllers

Services



Interacts with the database to perform specific business logic functionality. Each service implements an interface.

