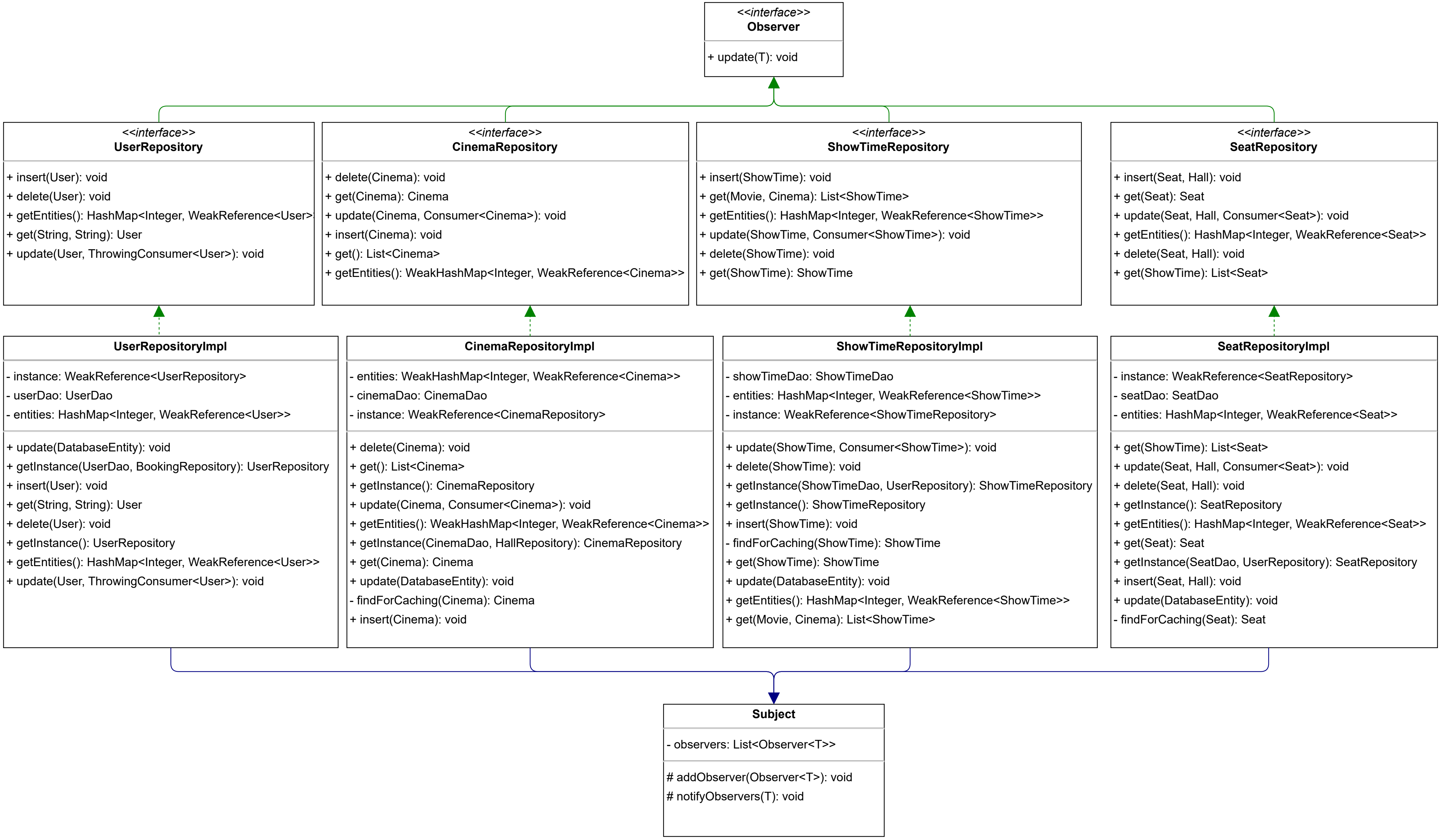
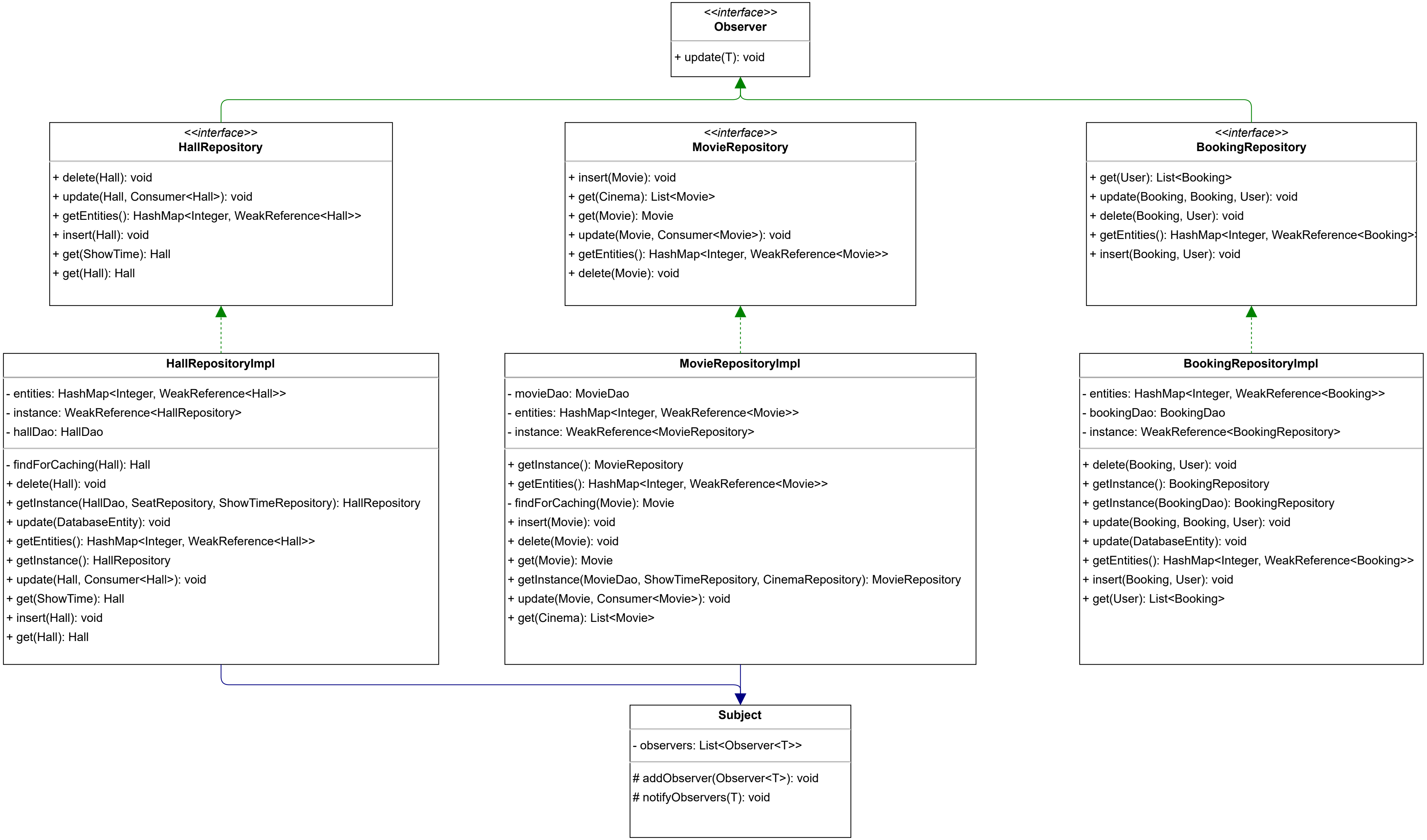

<<interface>>
Observer
+ update(T): void
<<interface>>
UserRepository
+ insert(User): void
+ delete(User): void
+ getEntities(): HashMap<Integer, WeakReference<User>
+ get(String, String): User
+ update(User, ThrowingConsumer<User>): void
<<interface>>
CinemaRepository
+ delete(Cinema): void
+ get(Cinema): Cinema
+ update(Cinema, Consumer<Cinema>): void
+ insert(Cinema): void
+ get(): List<Cinema>
+ getEntities(): WeakHashMap<Integer, WeakReference<Cinema>>
<<interface>>
ShowTimeRepository
+ insert(ShowTime): void
+ get(Movie, Cinema): List<ShowTime>
+ getEntities(): HashMap<Integer, WeakReference<ShowTime>>
+ update(ShowTime, Consumer<ShowTime>): void
+ delete(ShowTime): void
+ get(ShowTime): ShowTime
<<interface>>
SeatRepository
+ insert(Seat, Hall): void
+ get(Seat): Seat
+ update(Seat, Hall, Consumer<Seat>): void
+ getEntities(): HashMap<Integer, WeakReference<Seat>>
+ delete(Seat, Hall): void
+ get(ShowTime): List<Seat>
UserRepositoryImpl
- instance: WeakReference<UserRepository>
- userDao: UserDao
- entities: HashMap<Integer, WeakReference<User>>
+ update(DatabaseEntity): void
+ getInstance(UserDao, BookingRepository): UserRepository
+ insert(User): void
+ get(String, String): User
+ delete(User): void
+ getInstance(): UserRepository
+ getEntities(): HashMap<Integer, WeakReference<User>>
+ update(User, ThrowingConsumer<User>): void
CinemaRepositoryImpl
- entities: WeakHashMap<Integer, WeakReference<Cinema>>
- cinemaDao: CinemaDao
- instance: WeakReference<CinemaRepository>
+ delete(Cinema): void
+ get(): List<Cinema>
+ getInstance(): CinemaRepository
+ update(Cinema, Consumer<Cinema>): void
+ getEntities(): WeakHashMap<Integer, WeakReference<Cinema>>
+ getInstance(CinemaDao, HallRepository): CinemaRepository
+ get(Cinema): Cinema
+ update(DatabaseEntity): void
- findForCaching(Cinema): Cinema
+ insert(Cinema): void
ShowTimeRepositoryImpl
- showTimeDao: ShowTimeDao
- entities: HashMap<Integer, WeakReference<ShowTime>>
- instance: WeakReference<ShowTimeRepository>
+ update(ShowTime, Consumer<ShowTime>): void
+ delete(ShowTime): void
+ getInstance(ShowTimeDao, UserRepository): ShowTimeRepository
+ getInstance(): ShowTimeRepository
+ insert(ShowTime): void
- findForCaching(ShowTime): ShowTime
+ get(ShowTime): ShowTime
+ update(DatabaseEntity): void
+ getEntities(): HashMap<Integer, WeakReference<ShowTime>>
+ get(Movie, Cinema): List<ShowTime>
SeatRepositoryImpl
- instance: WeakReference<SeatRepository>
- seatDao: SeatDao
- entities: HashMap<Integer, WeakReference<Seat>>
+ get(ShowTime): List<Seat>
+ update(Seat, Hall, Consumer<Seat>): void
+ delete(Seat, Hall): void
+ getInstance(): SeatRepository
+ getEntities(): HashMap<Integer, WeakReference<Seat>>
+ get(Seat): Seat
+ getInstance(SeatDao, UserRepository): SeatRepository
+ insert(Seat, Hall): void
+ update(DatabaseEntity): void
- findForCaching(Seat): Seat
Subject
- observers: List<Observer<T>>
# addObserver(Observer<T>): void
# notifyObservers(T): void


<<interface>>
Observer
+ update(T): void
<<interface>>
HallRepository
+ delete(Hall): void
+ update(Hall, Consumer<Hall>): void
+ getEntities(): HashMap<Integer, WeakReference<Hall>>
+ insert(Hall): void
+ get(ShowTime): Hall
+ get(Hall): Hall
<<interface>>
MovieRepository
+ insert(Movie): void
+ get(Cinema): List<Movie>
+ get(Movie): Movie
+ update(Movie, Consumer<Movie>): void
+ getEntities(): HashMap<Integer, WeakReference<Movie>>
+ delete(Movie): void
<<interface>>
BookingRepository
+ get(User): List<Booking>
+ update(Booking, Booking, User): void
+ delete(Booking, User): void
+ getEntities(): HashMap<Integer, WeakReference<Booking>
+ insert(Booking, User): void
HallRepositoryImpl
- entities: HashMap<Integer, WeakReference<Hall>>
- instance: WeakReference<HallRepository>
- hallDao: HallDao
- findForCaching(Hall): Hall
+ delete(Hall): void
+ getInstance(HallDao, SeatRepository, ShowTimeRepository): HallRepository
+ update(DatabaseEntity): void
+ getEntities(): HashMap<Integer, WeakReference<Hall>>
+ getInstance(): HallRepository
+ update(Hall, Consumer<Hall>): void
+ get(ShowTime): Hall
+ insert(Hall): void
+ get(Hall): Hall
MovieRepositoryImpl
- movieDao: MovieDao
- entities: HashMap<Integer, WeakReference<Movie>>
- instance: WeakReference<MovieRepository>
+ getInstance(): MovieRepository
+ getEntities(): HashMap<Integer, WeakReference<Movie>>
- findForCaching(Movie): Movie
+ insert(Movie): void
+ delete(Movie): void
+ get(Movie): Movie
+ getInstance(MovieDao, ShowTimeRepository, CinemaRepository): MovieRepository
+ update(Movie, Consumer<Movie>): void
+ get(Cinema): List<Movie>
BookingRepositoryImpl
- entities: HashMap<Integer, WeakReference<Booking>>
- bookingDao: BookingDao
- instance: WeakReference<BookingRepository>
+ delete(Booking, User): void
+ getInstance(): BookingRepository
+ getInstance(BookingDao): BookingRepository
+ update(Booking, Booking, User): void
+ update(DatabaseEntity): void
+ getEntities(): HashMap<Integer, WeakReference<Booking>>
+ insert(Booking, User): void
+ get(User): List<Booking>
Subject
- observers: List<Observer<T>>
# addObserver(Observer<T>): void
# notifyObservers(T): void