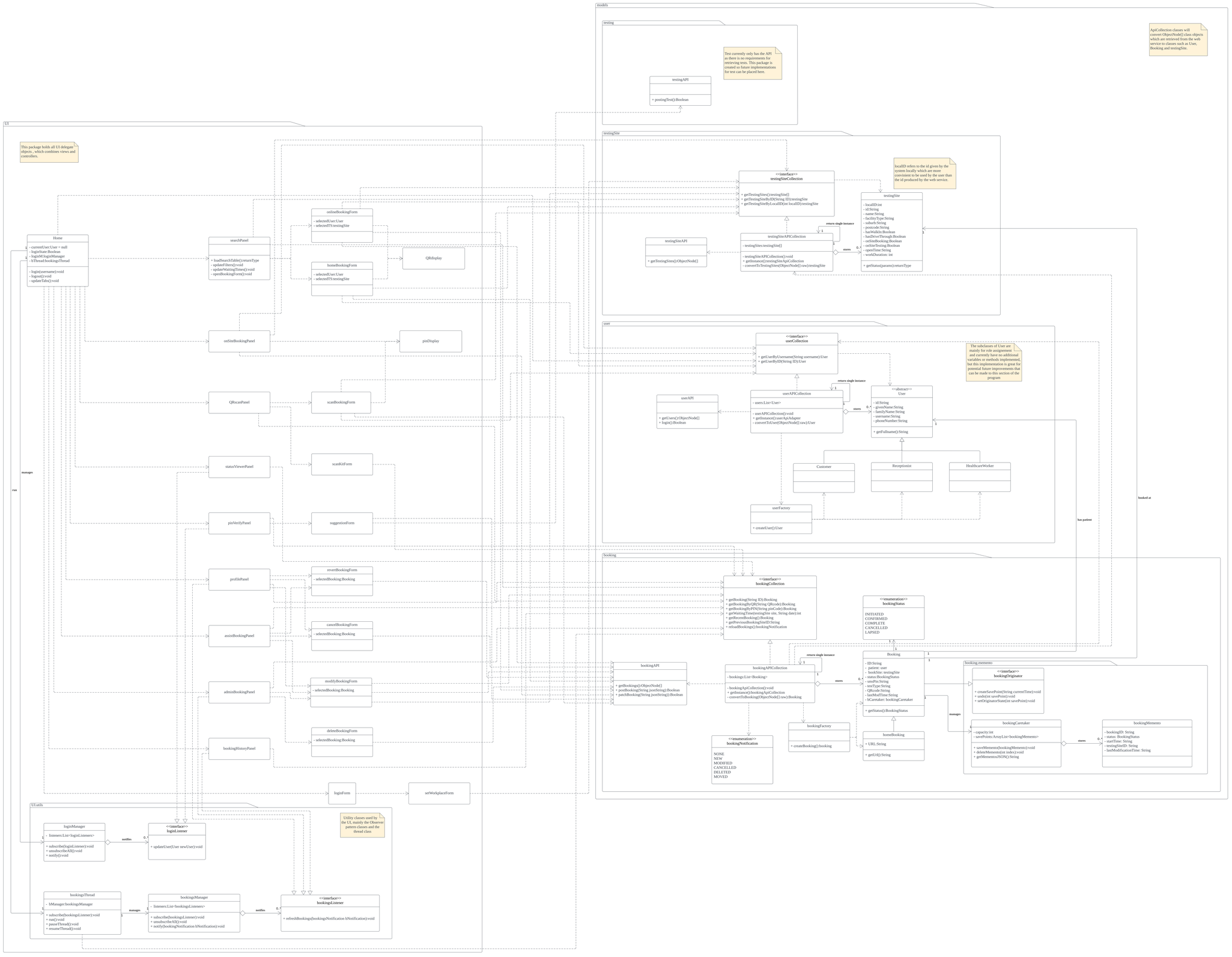
models
testing
Test currently only has the API as there is no requirements for retrieving tests. This package is created so future implementations for test can be placed here.
testingAPI
+ postingTest():Boolean
ApiCollection classes will convert ObjectNode[] class objects which are retrieved from the web service to classes such as User, Booking and testingSite.
UI
This package holds all UI delegate objects , which combines views and controllers.
Home
- currentUser:User = null
- loginState:Boolean
- loginMLoginManager
- bThread:bookingsThread
+ login(username):void
- logout():void
- updateTabs():void
searchPanel
+ loadSearchTable():returnType
- updateFilters():void
- updateWaitingTimes():void
- openBookingForm():void
onlineBookingForm
- selectedUser:User
- selectedTS:testingSite
homeBookingForm
- selectedUser:User
- selectedTS:testingSite
QRdisplay
onSiteBookingPanel
pinDisplay
QRscanPanel
scanBookingForm
statusViewerPanel
scanKitForm
pinVerifyPanel
suggestionForm
profilePanel
revertBookingForm
- selectedBooking:Booking
assistBookingPanel
cancelBookingForm
- selectedBooking:Booking
adminBookingPanel
modifyBookingForm
- selectedBooking:Booking
bookingHistoryPanel
deleteBookingForm
- selectedBooking:Booking
loginForm
setWorkplaceForm
manages
runs
UIutils
Utility classes used by the UI, mainly the Observer pattern classes and the thread class
loginManager
- listeners:List<loginListeners>
+ subscribe(loginListener):void
+ unsubscribeAll():void
+ notify():void
notifies
<<interface>>
loginListener
+ updateUser(User newUser):void
bookingsThread
- bManager:bookingsManager
+ subscribe(bookingsListener):void
+ run():void
+ pauseThread():void
+ resumeThread():void
manages
bookingsManager
- listeners:List<bookingsListeners>
+ subscribe(bookingsListener):void
+ unsubscribeAll():void
+ notify(bookingNotification bNotification):void
notifies
<<interface>>
bookingsListener
+ refreshBookings(bookingsNotification bNotification):void
testingSite
<<interface>>
testingSiteCollection
+ getTestingSites():testingSite[]
+ getTestingSiteByID(String ID):testingSite
+ getTestingSiteByLocalID(int localID):testingSite
localID refers to the id given by the system locally which are more convenient to be used by the user than the id produced by the web service.
testingSite
- localID:int
- id:String
- name:String
- facilityType:String
- suburb:String
- postcode:String
- hasWalkIn:Boolean
- hasDriveThrough:Boolean
- onSiteBooking:Boolean
- onSiteTesting:Boolean
- openTime:String
- workDuration: int
+ getStatus(params):returnType
testingSiteAPI
+ getTestingSites():ObjectNode[]
testingSiteAPICollection
- testingSites:testingSite[]
- testingSiteAPICollection():void
+ getInstance():testingSiteApiCollection
- convertToTestingSites(ObjectNode[] raw):testingSite
return single instance
stores
user
<<interface>>
userCollection
+ getUserByUsername(String username):User
+ getUserByID(String ID):User
The subclasses of User are mainly for role assignement and currently have no additional variables or methods implemented, but this implementation is great for potential future improvements that can be made to this section of the program
userAPI
+ getUsers():ObjectNode[]
+ login():Boolean
userAPICollection
- users:List<User>
- userAPICollection():void
+ getInstance():userApiAdapter
- convertToUser(ObjectNode[] raw):User
<<abstract>>
User
- id:String
- givenName:String
- family(Name:String
- username:String
- phoneNumber:String
+ getFullname():String
Customer
Receptionist
HealthcareWorker
userFactory
+ createUser():User
has patient
booked at
booking
<<interface>>
bookingCollection
+ getBookingsByQR(String QRcode):Booking
+ getBookingsByPIN(String pinCode):Booking
+ getWaitingTime(testingSite site, String date):int
+ getRecentBooking():Booking
+ getPreviousBookingSiteID:String
+ reloadBookings():bookingNotification
<<enumeration>>
bookingStatus
INITIATED
CONFIRMED
COMPLETE
CANCELLED
LAPSED
bookingAPI
+ getBookings():ObjectNode[]
+ postBooking(String jsonString):Boolean
+ patchBooking(String jsonString):Boolean
bookingAPICollection
- bookings:List<Booking>
- bookingApiCollection():void
+ getInstance():bookingApiCollection
- convertToBooking(ObjectNode[] raw):Booking
Booking
- ID:String
- patient: user
- bookSite: testingSite
- status:BookingStatus
- smsPin:String
- testType:String
- QRcode:String
- lastModifTime:String
- bCaretaker: bookingCaretaker
+ getStatus():BookingStatus
homeBooking
+ URL:String
+ getUrl():String
bookingFactory
+ createBooking():booking
<<enumeration>>
bookingNotification
NONE
NEW
MODIFIED
CANCELLED
DELETED
MOVED
booking.memento
<<interface>>
bookingOriginator
+ createSavePoint(String currentTime):void
+ undo(int savePoint):void
+ setOriginatorState(int savePoint):void
bookingCaretaker
- capacity:int
- savePoints:ArrayList<bookingMemento>
+ saveMemento(bookingMemento):void
+ deleteMemento(int index):void
+ getMementosJSON():String
bookingMemento
- bookingID: String
- status: BookingStatus
- startTime: String
- testingSiteID: String
- lastModificationTime: String