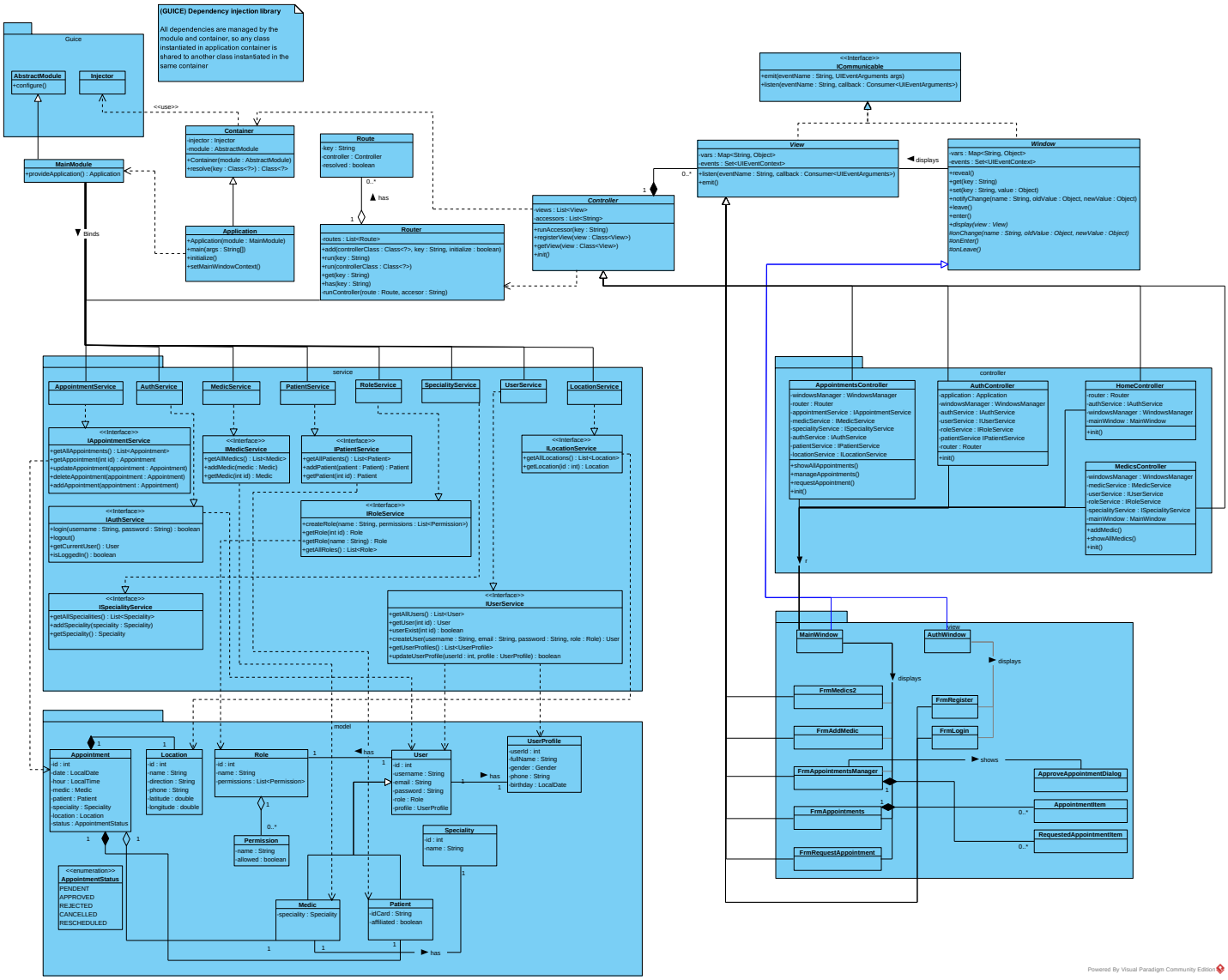
(GUICE) Dependency injection library
All dependencies are managed by the module and container, so any class instantiated in application container is shared to another class instantiated in the same container
Guice
AbstractModule
+configure()
Injector
<<use>>
MainModule
+provideApplication() : Application
binds
Container
-injector : Injector
-module : AbstractModule
+Container(module : AbstractModule)
+resolve(key : Class<?>) : Class<?>
Application
+Application(module : MainModule)
+main(args : String[])
+initialize()
+setMainWindowContext()
Route
-key : String
-controller : Controller
-resolved : boolean
0..*
has
1
Router
-routes : List<Route>
+add(controllerClass : Class<?>, key : String, initialize : boolean)
+run(key : String)
+run(controllerClass : Class<?>)
+get(key : String)
+has(key : String)
-runController(route : Route, accesor : String)
<<Interface>>
ICommunicable
+emit(eventName : String, UIEventArguments args)
+listen(eventName : String, callback : Consumer<UIEventArguments>)
View
-vars : Map<String, Object>
-events : Set<UIEventContext>
+listen(eventName : String, callback : Consumer<UIEventArguments>)
+emit()
displays
Window
-vars : Map<String, Object>
-events : Set<UIEventContext>
+reveal()
+get(key : String)
+set(key : String, value : Object)
+notifyChange(name : String, oldValue : Object, newValue : Object)
+leave()
+enter()
+display(view : View)
#onChange(name : String, oldValue : Object, newValue : Object)
#onEnter()
#onLeave()
Controller
-views : List<View>
-accessors : List<String>
+runAccessor(key : String)
+registerView(view : Class<View>)
+getView(view : Class<View>)
+init()
service
AppointmentService
AuthService
MedicService
PatientService
RoleService
SpecialityService
UserService
LocationService
<<Interface>>
IAppointmentService
+getAllAppointments() : List<Appointment>
+getAppointment(int id) : Appointment
+updateAppointment(appointment : Appointment)
+deleteAppointment(appointment : Appointment)
+addAppointment(appointment : Appointment)
<<Interface>>
IMedicService
+getAllMedics() : List<Medic>
+addMedic(medic : Medic)
+getMedic(int id) : Medic
<<Interface>>
IPatientService
+getAllPatients() : List<Patient>
+addPatient(patient : Patient) : Patient
+getPatient(int id) : Patient
<<Interface>>
ILocationService
+getAllLocations() : List<Location>
+getLocation(id : int) : Location
<<Interface>>
IAuthService
+login(username : String, password : String) : boolean
+logout()
+getCurrentUser() : User
+isLoggedIn() : boolean
<<Interface>>
IRoleService
+createRole(name : String, permissions : List<Permission>)
+getRole(int id) : Role
+getRole(name : String) : Role
+getAllRoles() : List<Role>
<<Interface>>
ISpecialityService
+getAllSpecialities() : List<Speciality>
+addSpeciality(speciality : Speciality)
+getSpeciality() : Speciality
<<Interface>>
IUserService
+getAllUsers() : List<User>
+getUser(int id) : User
+userExist(int id) : boolean
+createUser(username : String, email : String, password : String, role : Role) : User
+getUserProfiles() : List<UserProfile>
+updateUserProfile(userId : int, profile : UserProfile) : boolean
controller
AppointmentsController
-windowsManager : WindowsManager
-router : Router
-appointmentService : IAppointmentService
-medicService : IMedicService
-specialityService : ISpecialityService
-authService : IAuthService
-patientService : IPatientService
-locationService : ILocationService
+showAllAppointments()
+manageAppointments()
+requestAppointment()
+init()
AuthController
-application : Application
-windowsManager : WindowsManager
-authService : IAuthService
-userService : IUserService
-roleService : IRoleService
-patientService IPatientService
-router : Router
+init()
HomeController
-router : Router
-authService : IAuthService
-windowsManager : WindowsManager
-mainWindow : MainWindow
+init()
MedicsController
-windowsManager : WindowsManager
-medicService : IMedicService
-userService : IUserService
-roleService : IRoleService
-specialityService : ISpecialityService
-mainWindow : MainWindow
+addMedic()
+showAllMedics()
+init()
r
View
MainWindow
AuthWindow
displays
FrmMedics2
FrmRegister
FrmAddMedic
FrmLogin
shows
FrmAppointmentsManager
ApproveAppointmentDialog
FrmAppointments
AppointmentItem
RequestedAppointmentItem
FrmRequestAppointment
model
Appointment
-id : int
-date : LocalDate
-hour : LocalTime
-medic : Medic
-patient : Patient
-speciality : Speciality
-location : Location
-status : AppointmentStatus
Location
-id : int
-name : String
-direction : String
-phone : String
-latitude : double
-longitude : double
Role
-id : int
-name : String
-permissions : List<Permission>
Permission
-name : String
-allowed : boolean
User
-id : int
-username : String
-email : String
-password : String
-role : Role
-profile : UserProfile
UserProfile
-userId : int
-fullName : String
-gender : Gender
-phone : String
-birthday : LocalDate
Speciality
-id : int
-name : String
<<enumeration>>
AppointmentStatus
PENDENT
APPROVED
REJECTED
CANCELLED
RESCHEDULED
Medic
-speciality : Speciality
Patient
-idCard : String
-affiliated : boolean
Powered By Visual Paradigm Community Edition