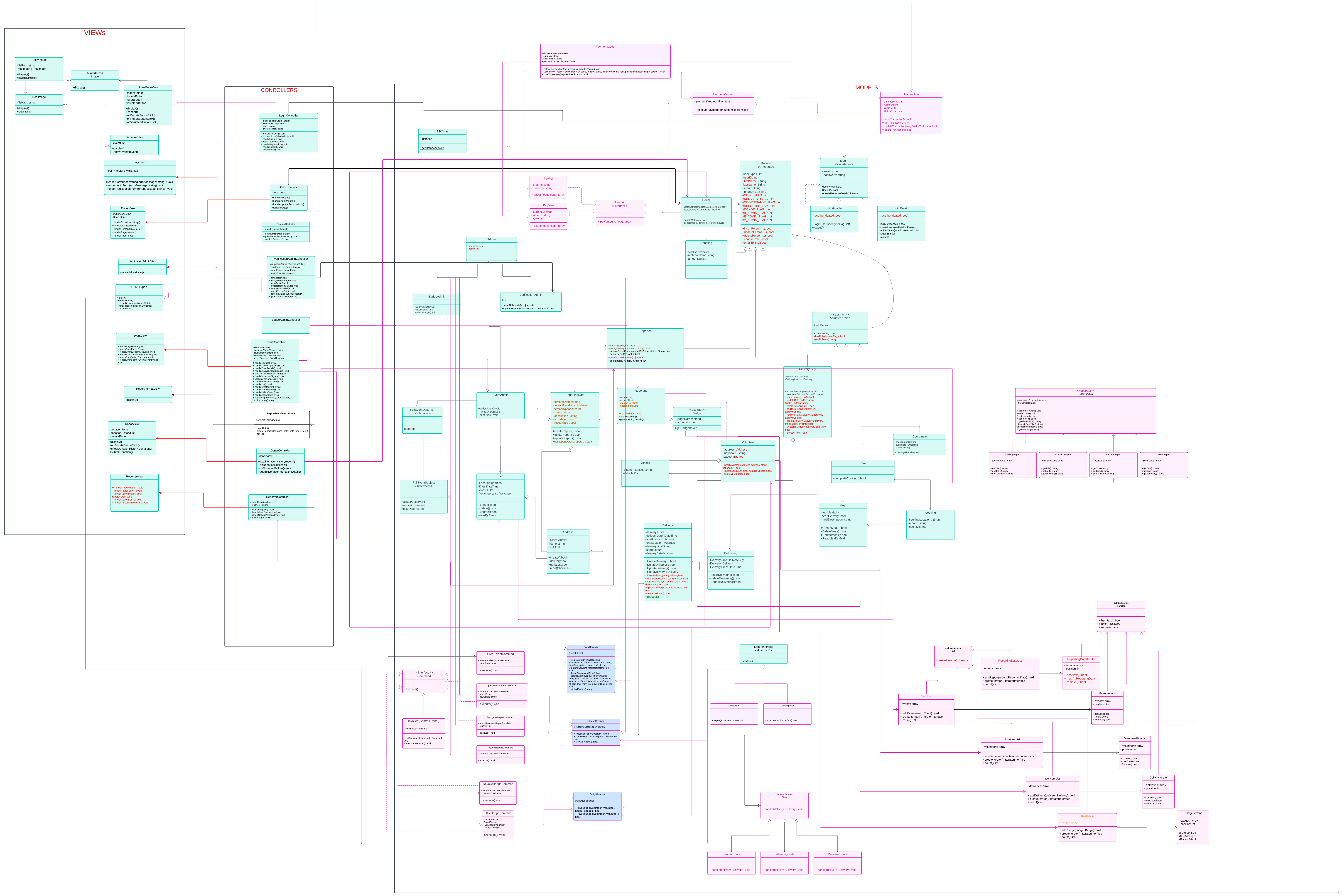
VIEWs
CONROLLERS
MODELS
ProxyImage
-filePath: string
-realImage : RealImage
+display()
+loadRealImage()
RealImage
-filePath: string
+display()
+loadImage()
<<Interface>>
Image
+display()
HomePageView
-image: Image
-donateButton
-reportButton
-volunteerButton
+display()
+ render()
+onDonateButtonClick()
+onReportButtonClick()
+onVolunteerButtonClick()
VolunteerView
-eventList
+display()
+showEvents(event)
LoginView
- loginHandler : withEmail
+renderForm(mode:string,errorMessage: string) : void
- renderLoginForm(errorMessage: string) : void
- renderRegistrationForm(errorMessage: string) : void
DonorView
-DonorView view
-Donor donor
+renderDonationHistory()
+renderDonationForm()
+renderPersonalInfoForm()
+renderPageHeader()
+renderPageFooter()
VerificationAdminView
+renderAdminPanel()
HTMLExport
+ export( )
- renderHeader()
- renderBody(array $reportData)
- renderReportItems(array $items)
- renderFooter()
EventView
+ renderPageHeader(): void
+ renderPageFooter(): void
+ renderEventList(array $events): void
+ renderEventDetails(Event $event): void
+ renderError(string $message): void
+ renderEventForm(?Event $event = null): void
ReportFormatView
+display()
DonorView
-donationForm
-donationHistoryList
-donateButton
+display()
+onDonateButtonClick()
+showDonationHistory(donations)
+submitDonation()
ReporterView
+ renderPageHeader(): void
+ renderPageFooter(): void
+renderReportHistory(array reportHistory):void
+renderReportForm(): void
+renderPersonalInfoForm(): void
LoginController
- loginHandler: LoginHandler
- view: EmailLoginView
- mode: string
- errorMessage: string
+ handleRequest(): void
- processFormSubmission(): void
- handleLogin(): void
+ fetchCountries(): void
- handleRegistration(): void
- handleLogout(): void
- renderPage(): void
DonorController
-Donor donor
+handleRequest()
+handleAddDonation()
+handleUpdatePersonalInfo()
+renderPage()
PaymentController
- model: PaymentModel
+ getPaymentData(): array
+ getUserDetails(email: string): int
+ validatePayment(): void
VerificationAdminController
- verificationAdmin: VerificationAdmin
- reportReceiver: ReportReceiver
- controlPanel: ControlPanel
- adminView: VAdminView
+ handleRequest()
+ recognizeReport(reportID)
+ showAdminPanel()
+ prepareReportData(reports)
+ handlePostAction(action)
+ formatReportData(report)
+ generateActionButtons(reportID)
+ generateSummary(reports)
BadgeAdminController
EventController
- view: EventView
- volunteerView: VolunteerView
- isVolunteerContext: bool
- controlPanel: ControlPanel
- eventReceiver: EventReceiver
+ handleRequest(): void
+ handleUpcomingEvents(): void
+ handleEventDetails(): void
+ handleAjaxVolunteerSignup(): void
+ getUserDetails(email: string): int
+ handleVolunteerSignup(): void
+ validateAdminAccess(): void
+ update(message: string): void
+ handleList(): void
+ handleCreateEvent(): void
+ handleUpdateEvent(): void
+ handleDeleteEvent(): void
+ handleAssignStaff(): void
+ validateInputParams(params: array, required: array): array
ReportTemplateController
-ReportFormatView
+LoadData()
+createReport(title: string, data, dateTime: Date)
+setTitle()
DonorController
-donorView
+loadDonationHistory(userId)
+onDonationSuccess()
+onDonationFailure(error)
+submitDonation(donationDetails)
ReporterController
- view: ReporterView
- reporter: Reporter
+ handleRequest(): void
+ handleFormSubmission(): void
- handleUpdatePersonalInfo(): void
+ renderPage(): void
PaymentModel
- db: DatabaseConnection
- currency: string
- itemNumber: string
- paymentContext: PaymentContext
+ setPaymentMethod(method: string, orderId: ?string): void
+ validateAndProcessPayment(userID: string, orderId: string, donationAmount: float, paymentMethod: string = 'paypal'): array
- storeTransaction(paymentResult: array): void
PaymentContext
- paymentMethod: IPayment
+ executePayment(amount: mixed): mixed
Transaction
- transactionID: int
- $amount: int
- $userID: int
- date: DATETIME
+ storeTransaction(): bool
+ getTransactionID(): int
+ updateTransaction(array $fieldsToUpdate): bool
+ deleteTransaction(): bool
DBConn
+instance
+getInstance():void
PayPal
- orderId: string
- currency: string
+ pay(amount: float): array
PayVisa
- currency: string
- orderId: string
- CVV: int
+ pay(amount: float): array
IPayment
<<interface>>
+ pay(amount: float): array
Donor
-hestoryofMaterialsDonated:list<materials>
-hestotyofMonyDonated:list<Money>
+donateMaterials( ):void
+donateMoney(payment: IPayment):void
Donating
-donationType:enum
-materialName:string
-amount:double
Person
<<Abstract>>
-userTypeID:int
-userID: int
- firstName: String
-lastName: String
- email: String
- phoneNo : String
#COOK_FLAG : int
#DELIVERY_FLAG : int
#COORDINATOR_FLAG : int
#REPORTER_FLAG : int
#DONOR_FLAG : int
#B_ADMIN_FLAG : int
#E_ADMIN_FLAG : int
#V_ADMIN_FLAG : int
+insertPerson(...): bool
+updatePerson(...): bool
+deletePerson(...): bool
+chooseRole():bool
+emailExists():bool
ILogin
<<interface>>
- email: string
- password: string
+login(credentials)
+logout(): bool
+createUser(userDetails):Person
withGoogle
- isAuthenticated: bool
+ loginUser(userTypeFlag: int)
+logout()
withEmail
- isAuthenticated: bool
+login(credentials): bool
+createUser(userDetails):Person
+authenticate(email, password): bool
+logout(): bool
+register()
Admin
-adminID:string
-adminRole
BadgeAdmin
+verifyBadge():void
+giveBadge():void
+revokeBadge():void
VerificationAdmin
bva
+viewAllReports() : [ ] reports
+updateReportStatus(reportID, newStatus):bool
Reporter
+ submitReport(info): bool
+ recognizeReport(reportID: String): bool
+ updateReportStatus(reportID: String, status: String): bool
+deleteReport(reportID):bool
+getAllActiveReports():[ ]reports
+getReportsByUserID($reporterID)
<<abstract>>
VolunteerRoles
#ref: Person
+ chooseRole(): bool
+hasRole(int roleFlag): bool
+getAllRoles(): array
Delivery Guy
-VehicleType : Vehicle
-DeliveryList:List <Delivery>
+ ChooseDelivery(DelivryID: int): bool
+ completeDelivery(DelivryID: int): void
+insertDeliveryGuy(): bool
+updateDeliveryGuy(array $fieldsToUpdate):bool
+deleteDeliveryGuy(): bool
+addToDeliveryList(Delivery $delivery):bool
+removeFromDeliveryList(Delivery $delivery): bool
+assignDelivery(Delivery $delivery, string $deliveryTime): bool
+unassignDelivery(Delivery $delivery): bool
+chooseRole(): bool
EventAdmin
+collectData():void
+fundBuyers():void
+sendAide():void
ReportingData
-personInName:string
-personINaddress: Address
-personINphoneno: int
- status : enum
- description : string
- is_deleted: bool
- recognized : bool
+createReport(): bool
+deleteReport(): bool
+updateReport(): bool
+getReportDetails(reportID): data
Reporting
-reportID : int
-reporterID:int
-created_at : Date
- updated_at: Date
+updateTimestamp()
+addReporting()
+getReportingDetails()
<<abstract>>
Badge
-BadgeName: string
-badgeLvl: string
+getBadge():void
Coordinator
-coordinatorID:string
-eventDate : DateTime
-eventID:string
+ manageUtensils(): void
FullEventObserver
<<interface>>
update()
Volunteer
- address: Address
- nationalID:string
-badge: Badges
+insertVolunteer(Address address, string nationalID): bool
+updateVolunteer(array fieldsToUpdate): bool
+deleteVolunteer(): bool
Vehicle
-LisencPlateNo: string
-VehicleID:int
Cook
+completeCooking():bool
FullEventSubject
<<interface>>
registerObservert()
removeObserver()
notifyObservers()
Event
-Location:address
-Date:DateTime
-eventID:int
-Volanteers:list<Volanteer>
+create():bool
+delete():bool
+update():bool
+read():Event
Meal
-noofMeals:int
-needDelivery: bool
-mealDescription: string
+CreateMeal(): bool
+DeleteMeal(): bool
+UpdateMeal(): bool
+ReadMeal():Meal
Cooking
-cookingLocation : Enum
-mealID:string
-cookID:string
Address
-AddressID:Int
-name:string
-P_ID:int
+create():bool
+delete():bool
+update():bool
+read():Address
Delivery
-deliveryID: int
-deliveryDate: DateTime
-startLocation: Adress
-endLocation: Address
-deliveryGuyID: int
-status enum
-deliveryDetails: string
+CreateDelivery(): bool
+DeleteDelivery(): bool
+UpdateDelivery(): bool
+ReadDelivery():Delivery
+insertDelivery(string deliveryDate, string startLocation, string endLocation, int $deliveryGuyID, string status , string deliveryDetails): bool
+updateDelivery(array fieldsToUpdate): bool
+deleteDelivery(): bool
+request()
Delivering
-DeliveryGuy: DeliveryGuy
-Delivery: Delivery
-DeliveryTime: DateTime
+insertDelivering():bool
+deleteDelivering():bool
+updateDelivering():bool
<<abstract>>
ReportTemplate
- $exporter: ExporterInterface
- $reportData: array
+ generateReport(): void
- collectData(): void
# getHeader(): string
# getFooter(): string
# getTimestamp(): string
abstract # getTitle(): string
abstract # getBody(): array
# getSummary(): string
DeliveryReport
- $deliveryData: array
DonationReport
- $donationData: array
ReporterReport
- $reportData: array
EventReport
- $eventData: array
# getTitle(): string
# getBody(): array
# getSummary(): string
CreatEventCommad
- eventReceiver: EventReceiver
- eventData: array
+execute(): void
EventReceiver
# event: Event
+ createEvent(eventDate: string, eventLocation: Address, eventName: string, eventDescription: string, reqCooks: int, reqForDelivery: int, reqCoordinators: int): bool
+ deleteEvent(eventID: int): bool
+ updateEvent(eventID: int, eventDate: string, eventLocation: Address, eventName: string, eventDescription: string, reqCooks: int, reqForDelivery: int, reqCoordinators: int): bool
+ fetchAllEvents(): array
<<interface>>
ICommand
+execute()
UpdateReportStatusCommand
- facadReciver: ReportReceiver
- reportID: int
- newStatus: string
+execute(): void
RecognizeReportCommand
- reportReceiver: ReportReceiver
- reportID: int
+ execute(): void
ReportReceiver
# reportingData: ReportingData
+ recognizeReport(reportID): mixed
+ updateReportStatus(reportID, newStatus): bool
+ viewAllReports(): array
Invoker (ControlePanel)
- command: ICommand
+ setCommand(command: ICommand): void
+ executeCommand(): void
ViewAllReportsCommand
- facadReciver: ReportReceiver
+ execute(): void
RevokeBadgeCommad
- facadReciver: FacadReciver
- volunteer: Volunteer
+execute():void
BadgeReceiver
#badge: Badges
+ giveBadge(volunteer: Volunteer, badge: Badges): bool
+ revokeBadge(volunteer: Volunteer): bool
GIveBadgeCommad
- facadReciver: FacadReciver
- volunteer: Volunteer
- badge: Badges
+execute(): void
ExportInterface
<<Interface>>
+ export(...)
CsvExporter
+ export(array $reportData): void
JsonExporter
+ export(array $reportData): void
<<interface>>
State
+ handle(delivery: Delivery): void
PendingState
DeliveringState
DeliveredState
+ handle(delivery: Delivery): void
<<interface>>
Listt
+createIterator(): Iterator
ReportingDataList
- reports: array
+ addReport(report: ReportingData): void
+ createIterator(): IteratorInterface
+ count(): int
ReportingDataIterator
- reports: array
- position: int
+ hasNext(): bool
+ next(): ReportingData
+ remove(): bool
<<Interface>>
Iterator
+ hasNext(): bool
+ next(): Delivery
+ remove(): void
EventList
- events: array
+ addEvent(event: Event): void
+ createIterator(): IteratorInterface
+ count(): int
EventIterator
- events: array
- position: int
+hasNext():bool
+Next():Event
+Remove():bool
VolunteerList
- volunteers: array
+ addVolunteer(volunteer: Volunteer): void
+ createIterator(): IteratorInterface
+ count(): int
VolunteerIterator
- volunteers: array
- position: int
+hasNext():bool
+Next():Volunteer
+Remove():bool
DeliveryList
- deliveries: array
+ addDelivery(delivery: Delivery): void
+ createIterator(): IteratorInterface
+ count(): int
DeliviryIterator
- deliveries: array
- position: int
+hasNext():bool
+Next():Delivery
+Remove():bool
BadgeList
- badges: array
+ addBadge(badge: Badge): void
+ createIterator(): IteratorInterface
+ count(): int
BadgeIterator
- badges: array
- position: int
+hasNext():bool
+Next():Badge
+Remove():bool