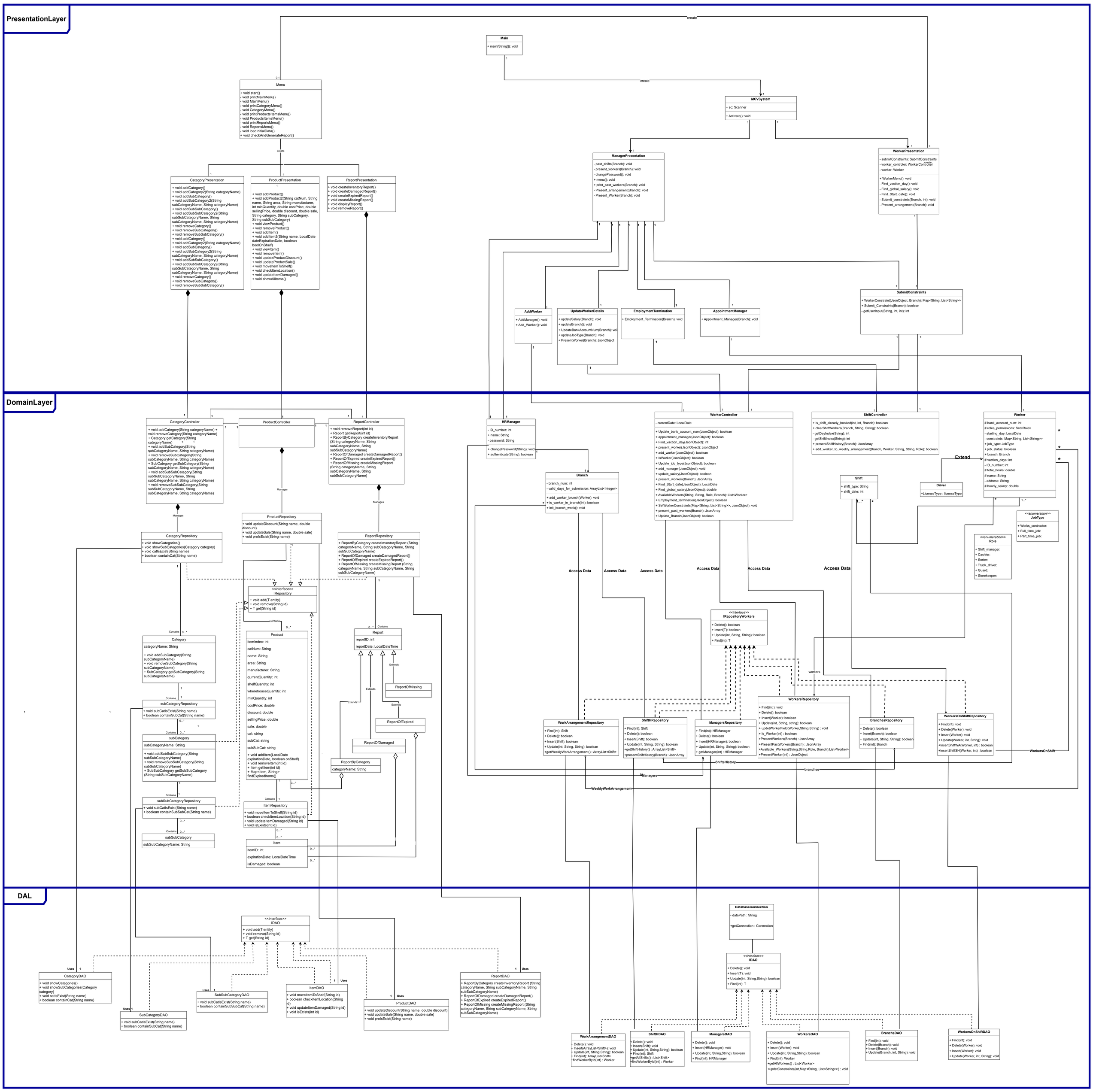
PresentationLayer
DomainLayer
DAL
Main
+ main(String[]): void
create
MCVSystem
+ sc: Scanner
+ Activate(): void
Menu
+ void start()
- void printMainMenu()
- void MainMenu()
- void printCategoryMenu()
- void CategoryMenu()
- void printProductsItemsMenu()
- void ProductsItemsMenu()
- void printReportsMenu()
- void ReportsMenu()
- void loadInitialData()
+ void checkAndGenerateReport()
ManagerPresentation
- past_shifts(Branch): void
- present_workers(Branch): void
- changePassword(): void
+ menu(): void
+ print_past_workers(Branch): void
- Present_arrangement(Branch): void
- Present_Worker(Branch): void
WorkerPresentation
- submitConstraints: SubmitConstraints
- worker_controler: WorkerController
- worker: Worker
+ WorkerMenu(): void
- Find_vaction_day(): void
- Find_global_salary(): void
- Find_Start_date(): void
- Submit_constraints(Branch, int): void
- Present_arrangement(Branch): void
CategoryPresentation
+ void addCategory()
+ void addCategory2(String categoryName)
+ void addSubCategory()
+ void addSubCategory2(String subCategoryName, String categoryName)
+ void addSubSubCategory()
+ void addSubSubCategory2(String subSubCategoryName, String subCategoryName, String categoryName)
+ void removeCategory()
+ void removeSubCategory()
+ void removeSubSubCategory()
+ void addCategory()
+ void addCategory2(String categoryName)
+ void addSubCategory()
+ void addSubCategory2(String subCategoryName, String categoryName)
+ void addSubSubCategory()
+ void addSubSubCategory2(String subSubCategoryName, String subCategoryName, String categoryName)
+ void removeCategory()
+ void removeSubCategory()
+ void removeSubSubCategory()
ProductPresentation
+ void addProduct()
+ void addProduct2(String catNum, String name, String area, String manufacturer, int minQuantity, double costPrice, double sellingPrice, double discount, double sale, String category, String subCategory, String subSubCategory)
+ void viewProduct()
+ void removeProduct()
+ void addItem()
+ void addItem2(String name, LocalDate dateExpirationDate, boolean boolOnShelf)
+ void viewItem()
+ void removeItem()
+ void updateProductDiscount()
+ void updateProductSale()
+ void moveItemToShelf()
+ void checkItemLocation()
+ void updateItemDamaged()
+ void showAllItems()
ReportPresentation
+ void createInventoryReport()
+ void createDamagedReport()
+ void createExpiredReport()
+ void createMissingReport()
+ void displayReport()
+ void removeReport()
AddWorker
+ AddManager(): void
+ Add_Worker(): void
UpdateWorkerDetails
+ updateSalary(Branch): void
+ updateBranch(): void
+ UpdateBankAccountNum(Branch): void
+ updateJobType(Branch): void
+ PresentWorker(Branch): JsonObject
EmploymentTermination
+ Employment_Termination(Branch): void
AppointmentManager
+ Appointment_Manager(Branch): void
SubmitConstraints
+ WorkerConstraint(JsonObject, Branch): Map<String, List<String>>
+ Submit_Constraints(Branch): boolean
- getUserInput(String, int, int): int
CategoryController
+ void addCategory(String categoryName) + void removeCategory(String categoryName)
+ Category getCategory(String categoryName)
+ void addSubCategory(String subCategoryName, String categoryName)
+ void removeSubCategory(String subCategoryName, String categoryName)
+ SubCategory getSubCategory(String subCategoryName, String categoryName)
+ void addSubSubCategory(String subSubCategoryName, String subCategoryName, String categoryName)
+ void removeSubSubCategory(String subSubCategoryName, String subCategoryName, String categoryName)
ProductController
ReportController
+ void removeReport(int id)
+ Report getReport(int id)
+ ReportByCategory createInventoryReport (String categoryName, String subCategoryName, String subSubCategoryName)
+ ReportOfDamaged createDamagedReport()
+ ReportOfExpired createExpiredReport()
+ ReportOfMissing createMissingReport (String categoryName, String subCategoryName, String subSubCategoryName)
Manages
ProductRepository
+ void updateDiscount(String name, double discount)
+ void updateSale(String name, double sale)
+ void prolsExist(String name)
ReportRepository
+ ReportByCategory createInventoryReport (String categoryName, String subCategoryName, String subSubCategoryName)
+ ReportOfDamaged createDamagedReport()
+ ReportOfExpired createExpiredReport()
+ ReportOfMissing createMissingReport (String categoryName, String subCategoryName, String subSubCategoryName)
CategoryRepository
+ void showCategories()
+ void showSubCategories(Category category)
+ void catIsExist(String name)
+ boolean containCat(String name)
<<interface>>
IRepository
+ void add(T entity)
+ void remove(String id)
+ T get(String id)
Contains
Category
categoryName: String
+ void addSubCategory(String subCategoryName)
+ void removeSubCategory(String subCategoryName)
+ SubCategory getSubCategory(String subCategoryName)
subCategoryRepository
+ void subCatIsExist(String name)
+ boolean containSubCat(String name)
subCategory
subCategoryName: String
+ void addSubSubCategory(String subSubCategoryName)
+ void removeSubSubCategory(String subSubCategoryName)
+ SubCategory getSubSubCategory (String subSubCategoryName)
subSubCategoryRepository
+ void subCatIsExist(String name)
+ boolean containSubSubCat(String name)
subSubCategory
subSubCategoryName: String
Product
itemIndex: int
catNum: String
name: String
area: String
manufacturer: String
qurrentQuantity: int
shelfQuantity: int
wherehouseQuantity: int
minQuantity: int
costPrice: double
discount: double
sellingPrice: double
sale: double
cat: string
subCat: string
subSubCat: string
+ void addItem(LocalDate expirationDate, boolean onShelf)
+ void removeItem(int id)
+ Item getItem(int id)
+ Map<Item, String> findExpiredItems()
ItemRepository
+ void moveItemToShelf(String id)
+ boolean checkItemLocation(String id)
+ void updateItemDamaged(String id)
+ void isExists(int id)
Item
itemID: int
expirationDate: LocalDateTime
isDamaged: boolean
Report
reportID: int
reportDate: LocalDateTime
Extends
ReportOfMissing
ReportOfExpired
ReportOfDamaged
ReportByCategory
categoryName: String
HRManager
- ID_number: int
+ name: String
- password: String
+ changePassword(String): void
+ authenticate(String): boolean
Branch
- branch_num: int
- valid_days_for_submission: ArrayList<Integer>
+ add_worker_brunch(Worker): void
+ is_worker_in_branch(int): boolean
+ init_branch_week(): void
WorkerController
- currentDate: LocalDate
+ Update_bank_account_num(JsonObject): boolean
+ appointment_manager(JsonObject): boolean
+ Find_vaction_day(JsonObject): int
+ present_worker(JsonObject): JsonObject
+ add_worker(JsonObject): boolean
+ IsWorker(JsonObject): boolean
+ Update_job_type(JsonObject): boolean
+ add_manager(JsonObject): void
+ update_salary(JsonObject): boolean
+ present_workers(Branch): JsonArray
+ Find_Start_date(JsonObject): LocalDate
+ Find_global_salary(JsonObject): double
+ AvailableWorkers(String, String, Role, Branch): List<Worker>
+ Employment_termination(JsonObject): boolean
+ SetWorkerConstraints(Map<String, List<String>>, JsonObject): void
+ present_past_workers(Branch): JsonArray
+ Update_Branch(JsonObject): boolean
ShiftController
+ is_shift_already_booked(int, int, Branch): boolean
+ clearShiftWorkers(Branch, String, String): boolean
- getDayIndex(String): int
- getShiftIndex(String): int
+ presentShiftHistory(Branch): JsonArray
+ add_worker_to_weekly_arrangement(Branch, Worker, String, String, Role): boolean
Worker
bank_account_num: int
roles_permissions: Set<Role>
- starting_day: LocalDate
- constraints: Map<String, List<String>>
+ job_type: JobType
+ job_status: boolean
+ branch: Branch
# vaction_days: int
- ID_number: int
total_hours: double
name: String
- address: String
hourly_salary: double
Extend
Driver
+LicenseType : licenseType
Shift
+ shift_type: String
+ shift_date: int
<<enumeration>>
JobType
+ Works_contractor:
+ Full_time_job:
+ Part_time_job:
<<enumeration>>
Role
+ Shift_manager:
+ Cashier:
+ Sorter:
+ Truck_driver:
+ Guard:
+ Storekeeper:
Access Data
<<interface>>
IRepositoryWorkers
+ Delete(): boolean
+ Insert(T): boolean
+ Update(int, String, String): boolean
+ Find(int): T
workers
WorkArrangementRepository
+ Find(int): Shift
+ Delete(): boolean
+ Insert(Shift): boolean
+ Update(int, String, String): boolean
+getWeeklyWorkArrangement() : ArrayList<Shift>
ShiftHRepository
+ Find(int): Shift
+ Delete(): boolean
+ Insert(Shift): boolean
+ Update(int, String, String): boolean
+getShiftHistory() : ArrayList<Shift>
+presentShiftHistory(Branch) : JsonArray
ManagersRepository
+ Find(int): HRManager
+ Delete(): boolean
+ Insert(HRManager): boolean
+ Update(int, String, String): boolean
+ getManager(int) : HRManager
WorkersRepository
+ Find(int): void
+ Delete(): boolean
+ Insert(Worker): boolean
+ Update(int, String, string): boolean
+ updetWorkerField(Worker,String,String) : void
+ Is_Worker(int) : boolean
+PresentWorkers(Branch) : JsonArray
+PresentPastWorkers(Branch) : JsonArray
+Available_Workers(String,String,Role, Branch):List<Worker>
+PresentWorker(int) : JsonObject
BranchesRepository
+ Delete(): boolean
+ Insert(Branch): boolean
+ Update(int, String, String): boolean
+ Find(int): Branch
WorkersOnShiftRepository
+ Find(int): void
+ Delete(Worker): void
+ Insert(Worker): void
+ Update(Worker, int, String): void
+InsertShiftWA(Worker, int) : boolean
+InsertShiftSH(Worker, int) : boolean
WorkersOnShift
ShiftsHistory
branches
Managers
WeeklyWorkArrangement
<<interface>>
IDAO
+ void add(T entity)
+ void remove(String id)
+ T get(String id)
Uses
CategoryDAO
+ void showCategories()
+ void showSubCategories(Category category)
+ void catIsExist(String name)
+ boolean containCat(String name)
SubCategoryDAO
+ void subCatIsExist(String name)
+ boolean containSubCat(String name)
SubSubCategoryDAO
+ void subCatIsExist(String name)
+ boolean containSubSubCat(String name)
ItemDAO
+ void moveItemToShelf(String id)
+ boolean checkItemLocation(String id)
+ void updateItemDamaged(String id)
+ void isExists(int id)
ProductDAO
+ void updateDiscount(String name, double discount)
+ void updateSale(String name, double sale)
+ void prolsExist(String name)
ReportDAO
+ ReportByCategory createInventoryReport (String categoryName, String subCategoryName, String subSubCategoryName)
+ ReportOfDamaged createDamagedReport()
+ ReportOfExpired createExpiredReport()
+ ReportOfMissing createMissingReport (String categoryName, String subCategoryName, String subSubCategoryName)
DatabaseConnection
- dataPath : String
+getConnection : Connection
<<interface>>
IDAO
+ Delete(): void
+ Insert(T): void
+ Update(int, String,String): boolean
+ Find(int): T
WorkArrangementDAO
+ Delete(): void
+ Insert(ArrayList<Shift>): void
+ Update(int, String,String): boolean
+ Find(int): ArrayList<Shift>
+findWorkerById(int) : Worker
ShiftHDAO
+ Delete(): void
+ Insert(Shift): void
+ Update(int, String,String): boolean
+ Find(int): Shift
+getAllShifts() : List<Shift>
+findWorkerById(int) : Worker
ManagersDAO
+ Delete(): void
+ Insert(HRManager): void
+ Update(int, String,String): boolean
+ Find(int): HRManager
WorkersDAO
+ Delete(): void
+ Insert(Worker): void
+ Update(int, String,String): boolean
+ Find(int): Worker
+getAllWorkers() : List<Worker>
+updetConstraints(int,Map<String, List<String>>) : void
BranchsDAO
+ Find(int): void
+ Delete(Branch): void
+ Insert(Branch): void
+ Update(Branch, int, String): void
WorkersOnShiftDAO
+ Find(int): void
+ Delete(Worker): void
+ Insert(Worker): void
+ Update(Worker, int, String): void