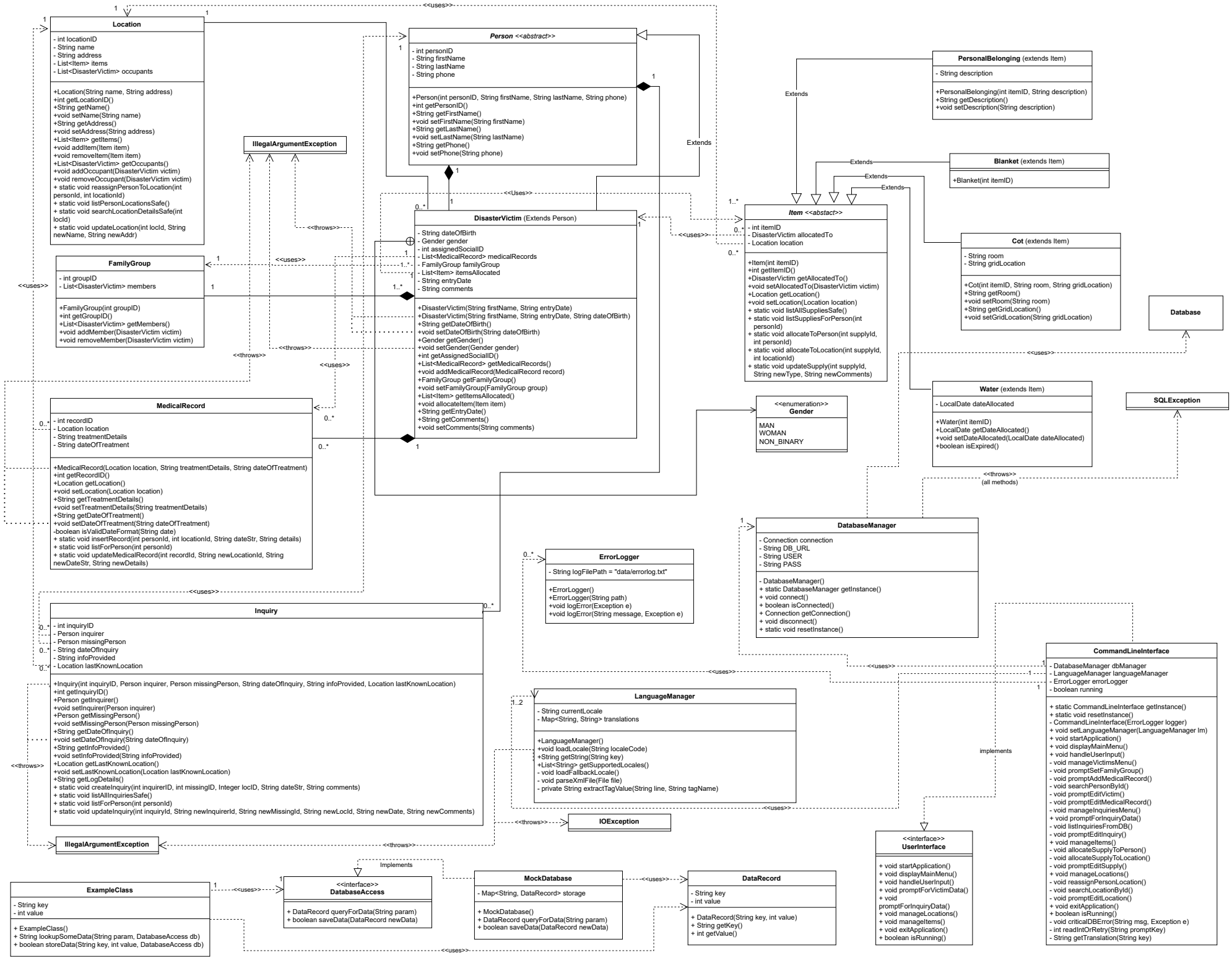
Location
- int locationID
- String name
- String address
- List<Item> items
- List<DisasterVictim> occupants
+Location(String name, String address)
+int getLocationID()
+String getName()
+void setName(String name)
+String getAddress()
+void setAddress(String address)
+List<Item> getItems()
+void addItem(Item item)
+void removeItem(Item item)
+List<DisasterVictim> getOccupants()
+void addOccupant(DisasterVictim victim)
+void removeOccupant(DisasterVictim victim)
+ static void reassignPersonToLocation(int personId, int locationId)
+ static void listPersonLocationsSafe()
+ static void searchLocationDetailsSafe(int locId)
+ static void updateLocation(int locId, String newName, String newAddr)
Person <<abstract>>
- int personID
- String firstName
- String lastName
- String phone
+Person(int personID, String firstName, String lastName, String phone)
+int getPersonID()
+String getFirstName()
+void setFirstName(String firstName)
+String getLastName()
+void setLastName(String lastName)
+String getPhone()
+void setPhone(String phone)
IllegalArgumentException
PersonalBelonging (extends Item)
- String description
+PersonalBelonging(int itemID, String description)
+String getDescription()
+void setDescription(String description)
Blanket (extends Item)
+Blanket(int itemID)
DisasterVictim (Extends Person)
- String dateOfBirth
- Gender gender
- int assignedSocialID
- List<MedicalRecord> medicalRecords
- FamilyGroup familyGroup
- List<Item> itemsAllocated
- String entryDate
- String comments
+DisasterVictim(String firstName, String entryDate)
+DisasterVictim(String firstName, String entryDate, String dateOfBirth)
+String getDateOfBirth()
+void setDateOfBirth(String dateOfBirth)
+Gender getGender()
+void setGender(Gender gender)
+int getAssignedSocialID()
+List<MedicalRecord> getMedicalRecords()
+void addMedicalRecord(MedicalRecord record)
+FamilyGroup getFamilyGroup()
+void setFamilyGroup(FamilyGroup group)
+List<Item> getItemsAllocated()
+void allocateItem(Item item)
+String getEntryDate()
+String getComments()
+void setComments(String comments)
Item <<abstact>>
- int itemID
- DisasterVictim allocatedTo
- Location location
+Item(int itemID)
+int getItemID()
+DisasterVictim getAllocatedTo()
+void setAllocatedTo(DisasterVictim victim)
+Location getLocation()
+void setLocation(Location location)
+ static void listAllSuppliesSafe()
+ static void listSuppliesForPerson(int personId)
+ static void allocateToPerson(int supplyId, int personId)
+ static void allocateToLocation(int supplyId, int locationId)
+ static void updateSupply(int supplyId, String newType, String newComments)
Cot (extends Item)
- String room
- String gridLocation
+Cot(int itemID, String room, String gridLocation)
+String getRoom()
+void setRoom(String room)
+String getGridLocation()
+void setGridLocation(String gridLocation)
Database
FamilyGroup
- int groupID
- List<DisasterVictim> members
+FamilyGroup(int groupID)
+int getGroupID()
+List<DisasterVictim> getMembers()
+void addMember(DisasterVictim victim)
+void removeMember(DisasterVictim victim)
Water (extends Item)
- LocalDate dateAllocated
+Water(int itemID)
+LocalDate getDateAllocated()
+void setDateAllocated(LocalDate dateAllocated)
+boolean isExpired()
SQLException
MedicalRecord
- int recordID
- Location location
- String treatmentDetails
- String dateOfTreatment
+MedicalRecord(Location location, String treatmentDetails, String dateOfTreatment)
+int getRecordID()
+Location getLocation()
+void setLocation(Location location)
+String getTreatmentDetails()
+void setTreatmentDetails(String treatmentDetails)
+String getDateOfTreatment()
+void setDateOfTreatment(String dateOfTreatment)
-boolean isValidDateFormat(String date)
+ static void insertRecord(int personId, int locationId, String dateStr, String details)
+ static void listForPerson(int personId)
+ static void updateMedicalRecord(int recordId, String newLocationId, String newDateStr, String newDetails)
<<enumeration>>
Gender
MAN
WOMAN
NON_BINARY
DatabaseManager
- Connection connection
- String DB_URL
- String USER
- String PASS
- DatabaseManager()
+ static DatabaseManager getInstance()
+ void connect()
+ boolean isConnected()
+ Connection getConnection()
+ void disconnect()
+ static void resetInstance()
ErrorLogger
- String logFilePath = "data/errorlog.txt"
+ErrorLogger()
+ErrorLogger(String path)
+void logError(Exception e)
+void logError(String message, Exception e)
Inquiry
- int inquiryID
- Person inquirer
- Person missingPerson
- String dateOfInquiry
- String infoProvided
- Location lastKnownLocation
+Inquiry(int inquiryID, Person inquirer, Person missingPerson, String dateOfInquiry, String infoProvided, Location lastKnownLocation)
+int getInquiryID()
+Person getInquirer()
+void setInquirer(Person inquirer)
+Person getMissingPerson()
+void setMissingPerson(Person missingPerson)
+String getDateOfInquiry()
+void setDateOfInquiry(String dateOfInquiry)
+String getInfoProvided()
+void setInfoProvided(String infoProvided)
+Location getLastKnownLocation()
+void setLastKnownLocation(Location lastKnownLocation)
+String getLogDetails()
+ static void createInquiry(int inquirerID, int missingID, Integer locID, String dateStr, String comments)
+ static void listAllInquiriesSafe()
+ static void listForPerson(int personId)
+ static void updateInquiry(int inquiryId, String newInquirerId, String newMissingId, String newLocId, String newDate, String newComments)
LanguageManager
- String currentLocale
- Map<String, String> translations
+LanguageManager()
+void loadLocale(String localeCode)
+String getString(String key)
+List<String> getSupportedLocales()
- void loadFallbackLocale()
- void parseXmlFile(File file)
- private String extractTagValue(String line, String tagName)
CommandLineInterface
- DatabaseManager dbManager
- LanguageManager languageManager
- ErrorLogger errorLogger
- boolean running
+ static CommandLineInterface getInstance()
+ static void resetInstance()
- CommandLineInterface(ErrorLogger logger)
+ void setLanguageManager(LanguageManager lm)
+ void startApplication()
+ void displayMainMenu()
+ void handleUserInput()
- void manageVictimsMenu()
- void promptSetFamilyGroup()
- void promptAddMedicalRecord()
- void searchPersonById()
- void promptEditVictim()
- void promptEditMedicalRecord()
- void manageInquiriesMenu()
+ void promptForInquiryData()
- void listInquiriesFromDB()
- void promptEditInquiry()
+ void manageItems()
- void allocateSupplyToPerson()
- void allocateSupplyToLocation()
- void promptEditSupply()
+ void manageLocations()
- void reassignPersonLocation()
- void searchLocationById()
- void promptEditLocation()
+ void exitApplication()
+ boolean isRunning()
- void criticalDBError(String msg, Exception e)
- int readIntOrRetry(String promptKey)
- String getTranslation(String key)
IOException
IllegalArgumentException
<<interface>>
UserInterface
+ void startApplication()
+ void displayMainMenu()
+ void handleUserInput()
+ void promptForVictimData()
+ void promptForInquiryData()
+ void manageLocations()
+ void manageItems()
+ void exitApplication()
+ boolean isRunning()
ExampleClass
- String key
- int value
+ ExampleClass()
+ String lookupSomeData(String param, DatabaseAccess db)
+ boolean storeData(String key, int value, DatabaseAccess db)
<<interface>>
DatabaseAccess
+ DataRecord queryForData(String param)
+ boolean saveData(DataRecord newData)
MockDatabase
- Map<String, DataRecord> storage
+ MockDatabase()
+ DataRecord queryForData(String param)
+ boolean saveData(DataRecord newData)
DataRecord
- String key
- int value
+ DataRecord(String key, int value)
+ String getKey()
+ int getValue()
<<uses>>
<<Uses>>
<<throws>>
Extends
Implements
implements
(all methods)