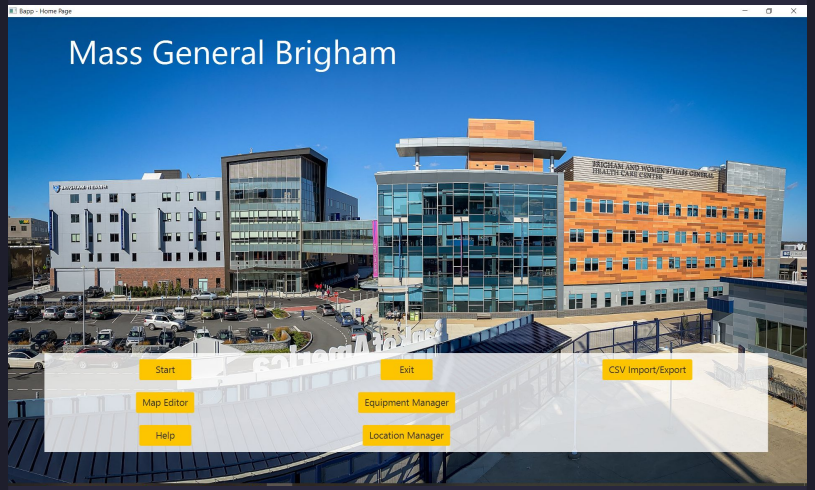
Software Application for Brigham & Women's Hospital

BEFORE & AFTER: APPLICATION

HOME PAGE

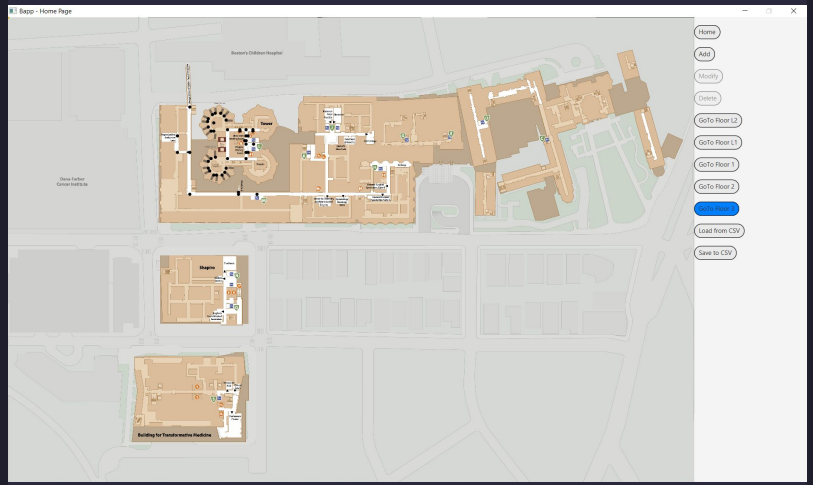
BEFORE



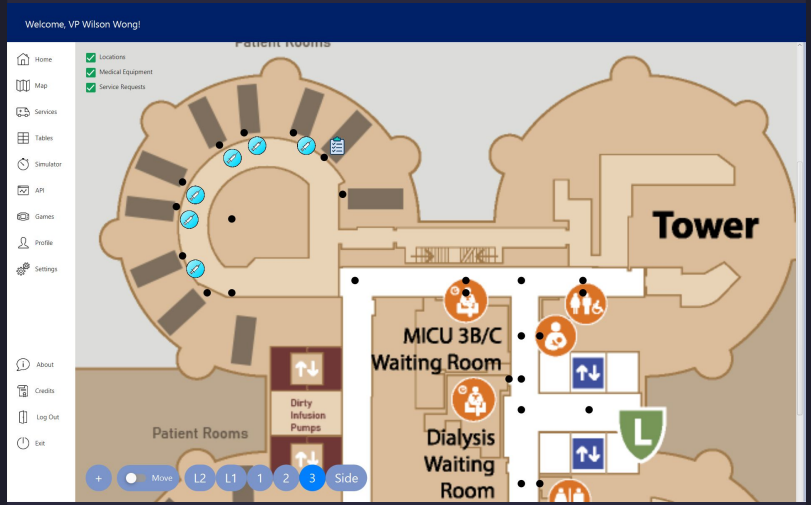
AFTER

MAP EDITOR

BEFORE

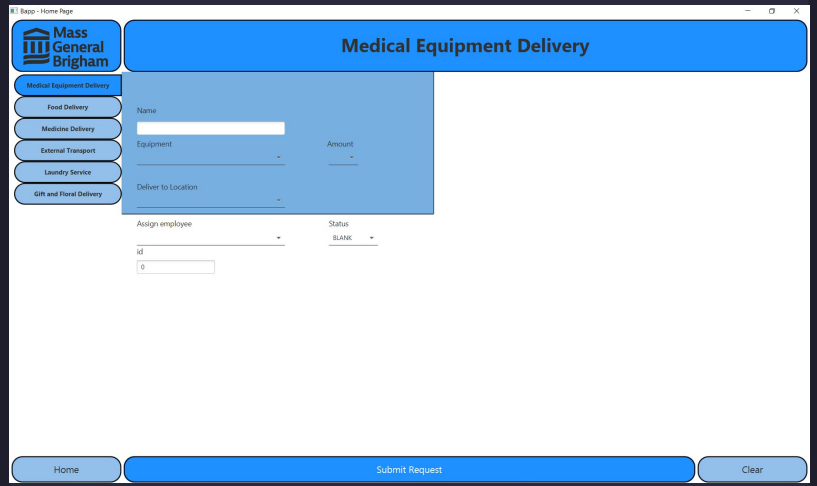


AFTER

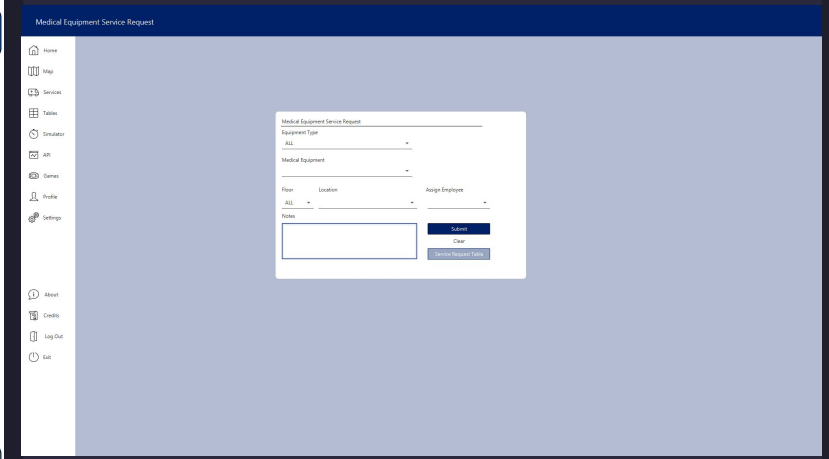


MEDICAL EQUIPMENT SERVICE REQUEST

BEFORE



AFTER



Goal of Team and My Role

Team

- Team of 10 people applying Agile Scrum methodology.
- Developed a desktop application for the hospital employees to access and use.
- Application focuses on creating and carrying out Service Requests.
- Database tables are maintained and implemented in an easy-to-use graphical interface
- Conducted daily scrums, sprint plannings and sprint reflections.

My Role

- Designed and developed back-end system including database management system and restore/backup subsystem.
- Implemented embedded and client-server database using Apache Derby and remote cloud-based database using MongoDB.
- As Assistant Lead Software Engineer, coordinated the other software engineers, particularly the back-end team, and performed code reviews through Git workflows.

118

JAVA CLASSES

5

INTERFACES

3

SUBSYSTEMS

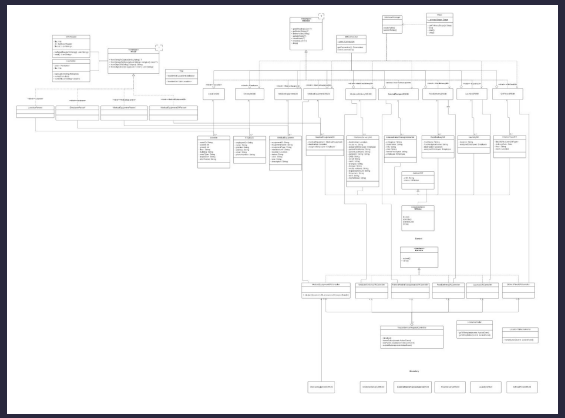
15,424

LINES OF JAVA CODE

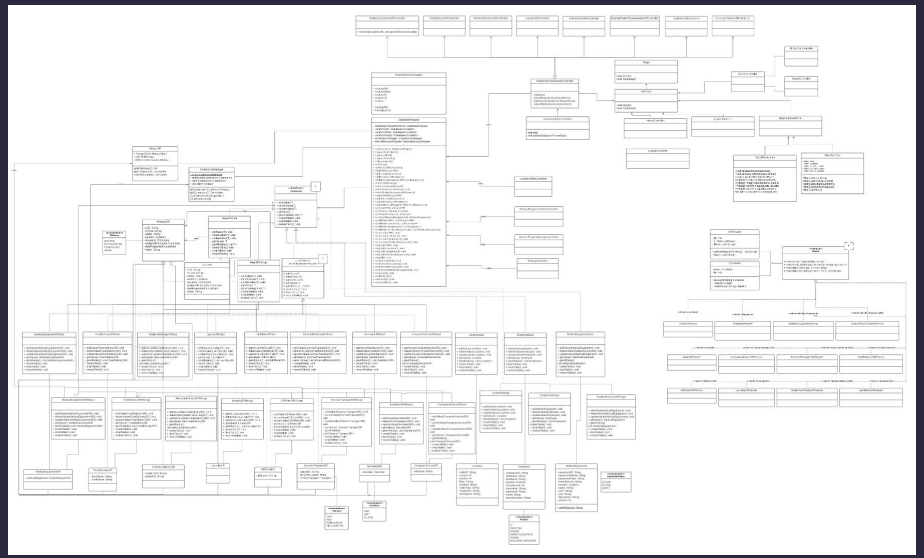
Diagrams

First Iteration Vs. Last Iteration UML Diagrams

First

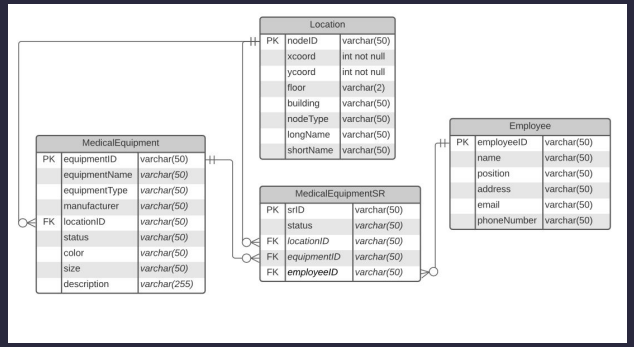


Last



First Iteration Vs. Last Iteration ERDs

First



Last

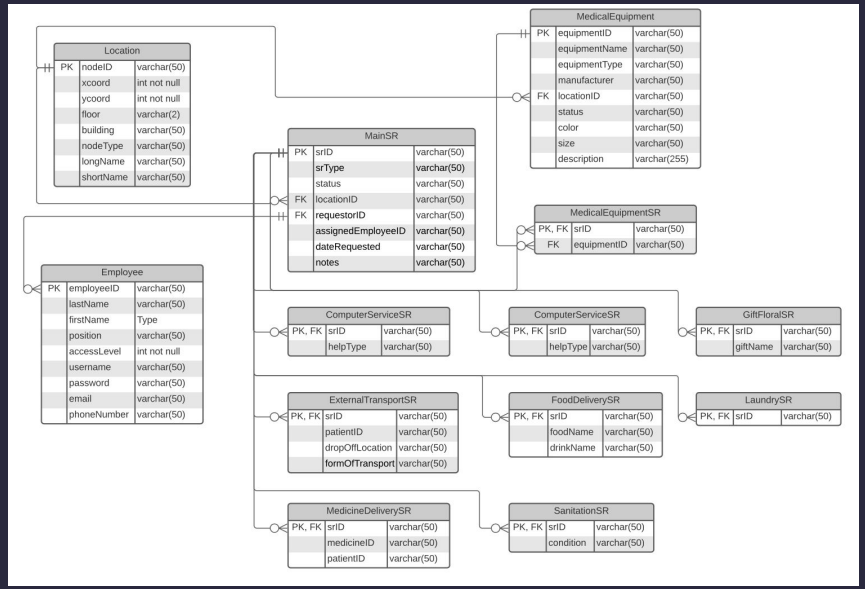
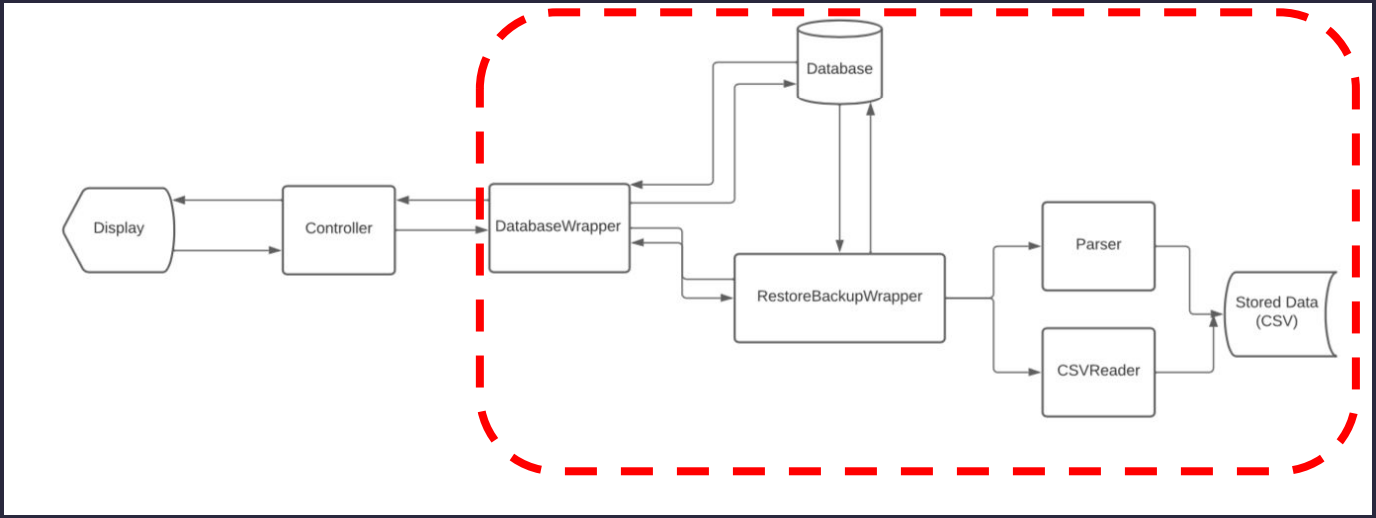
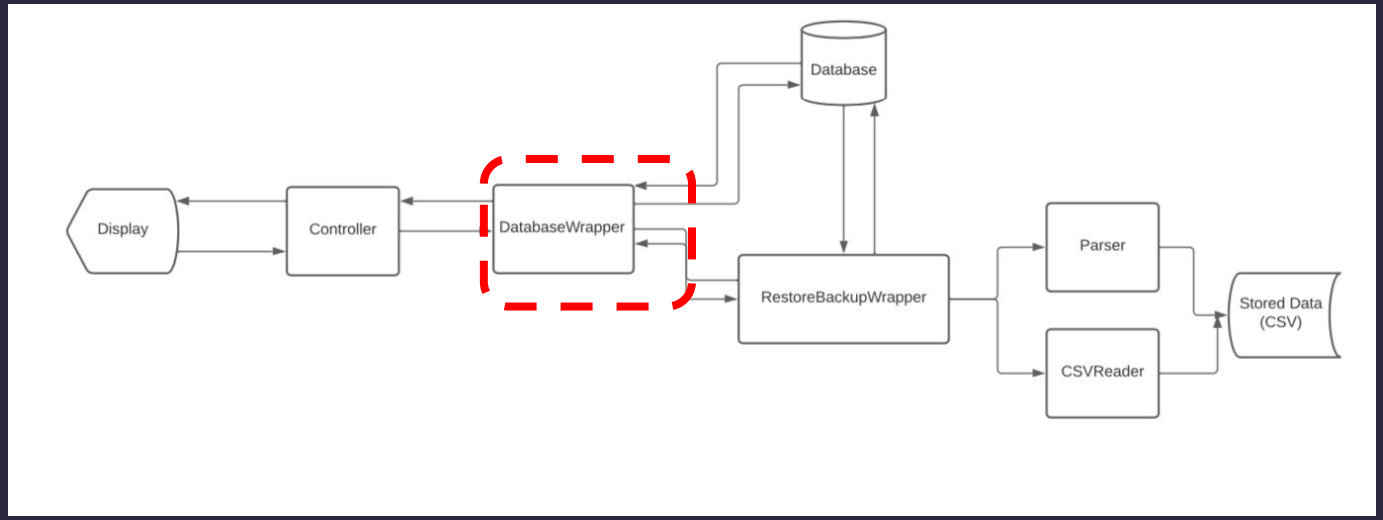


Diagram of Systems and Subsystems



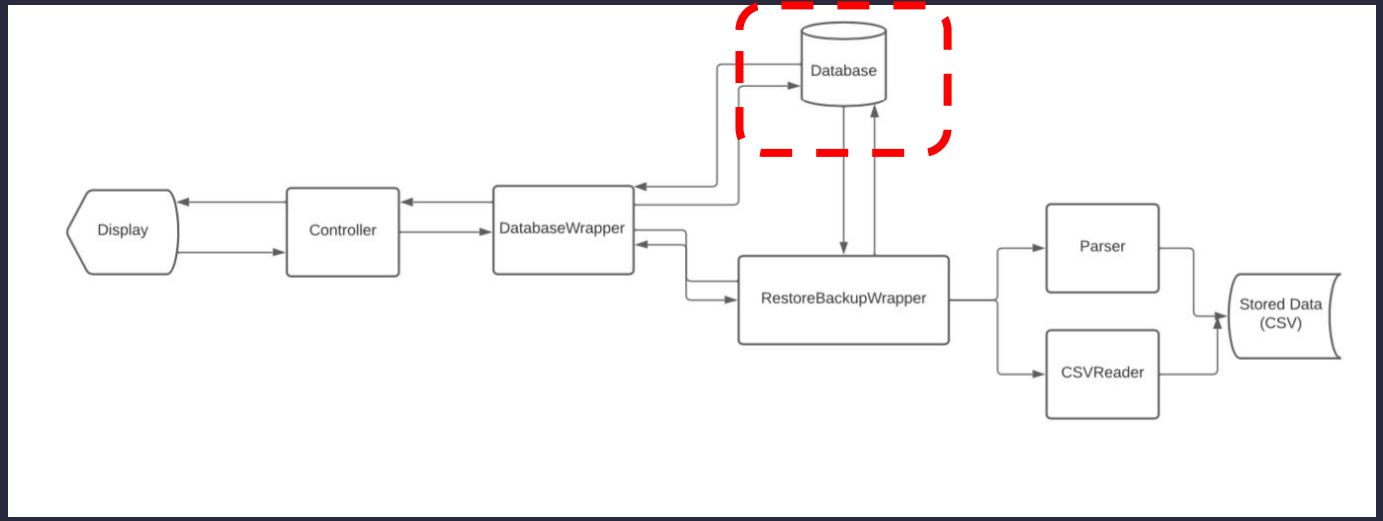
DatabaseWrapper



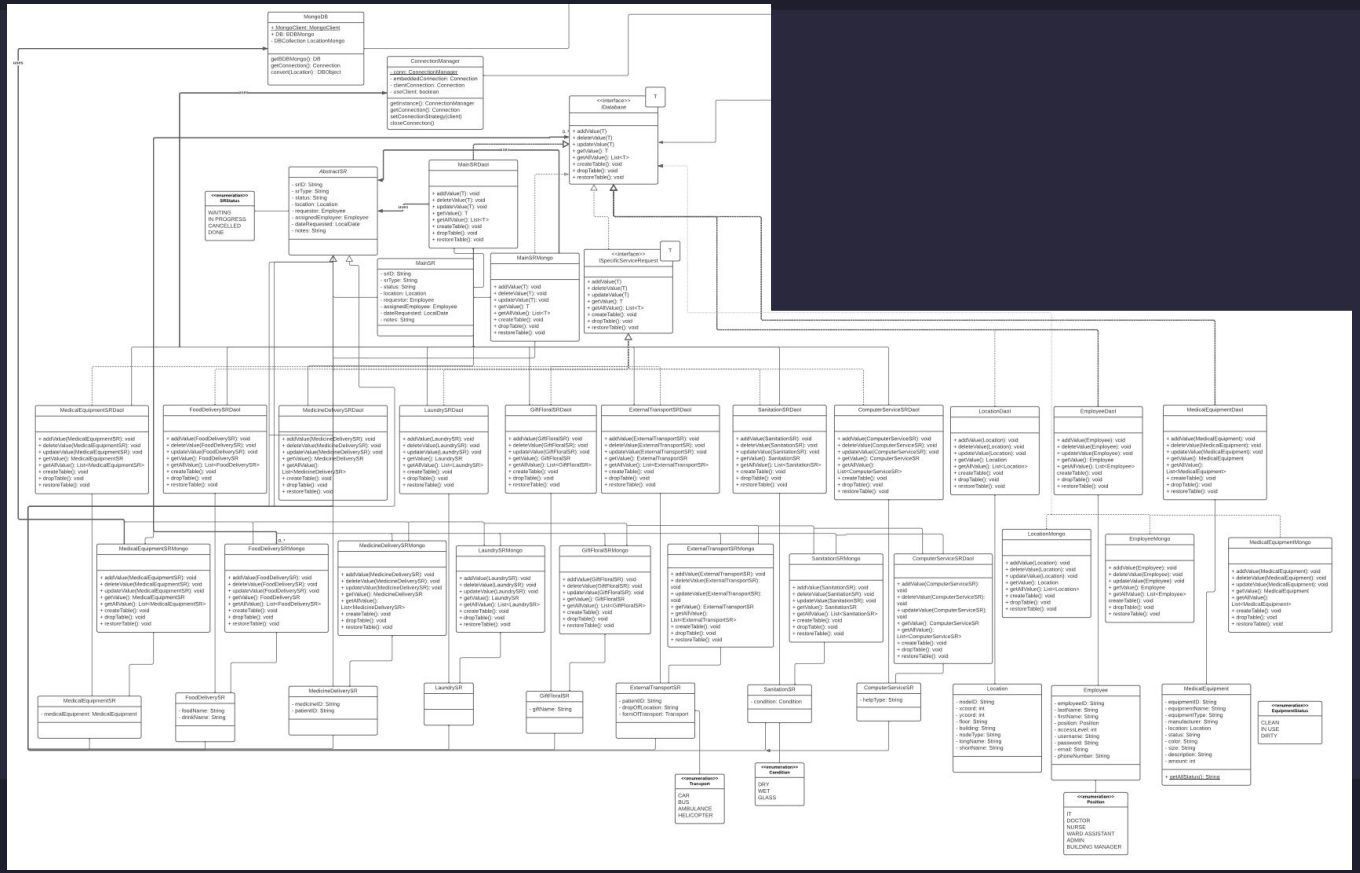
DatabaseWrapper

- Facade pattern
- Front-facing interface
- Masks complex database code and is what front-end uses to manipulate database
- Singleton pattern implementation

Databases Low-Level Code

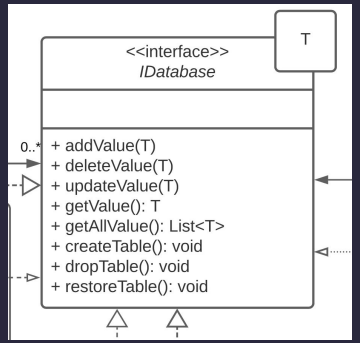


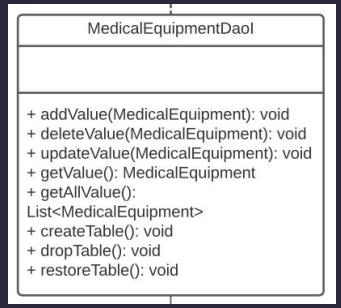
Databases Low-Level Code

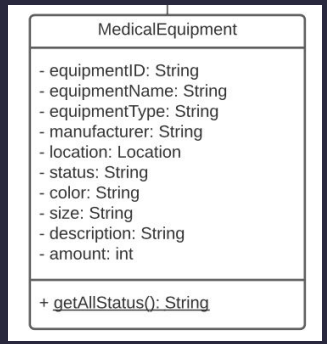


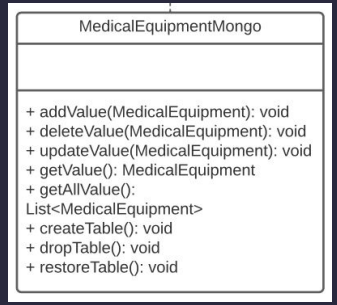
Databases Low-Level Code

- Used JDBC API for low-level access of databases
- Data access object (DAO) pattern using Generics
  - Data access object interface (e.g. **IDatabase**)
  - Data access object concrete class (e.g. **MedicalEquipmentDaoI**, **MedicalEquipmentMongo**)
  - Model object (e.g. **MedicalEquipment**)



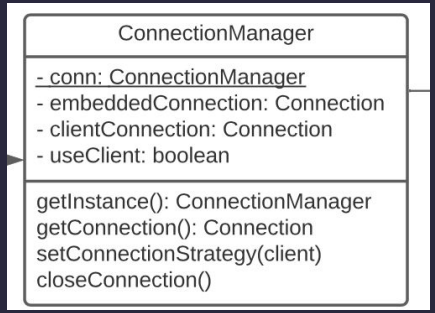


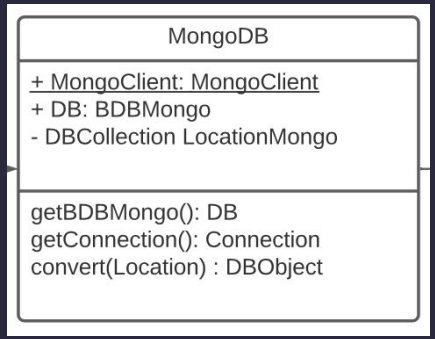




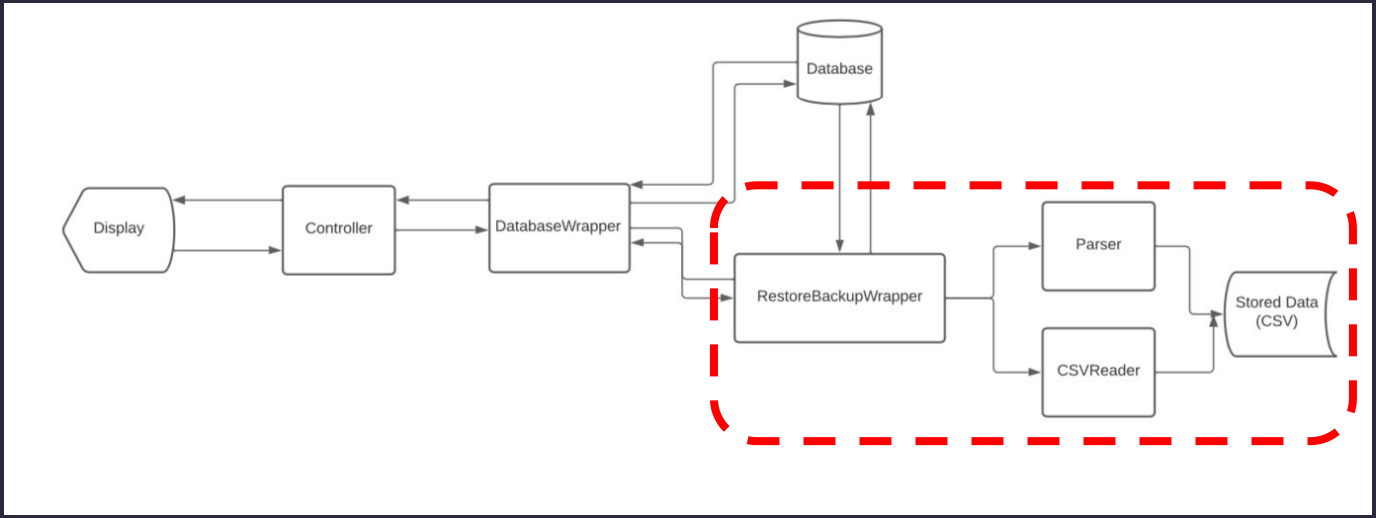
Databases Low-Level Code

- Classes to manage database connections
- **MongoDB** class
  - For cloud-based MongoDB database
- **ConnectionManager** class
  - For embedded and client-server Apache Derby databases
  - Switches between both connections

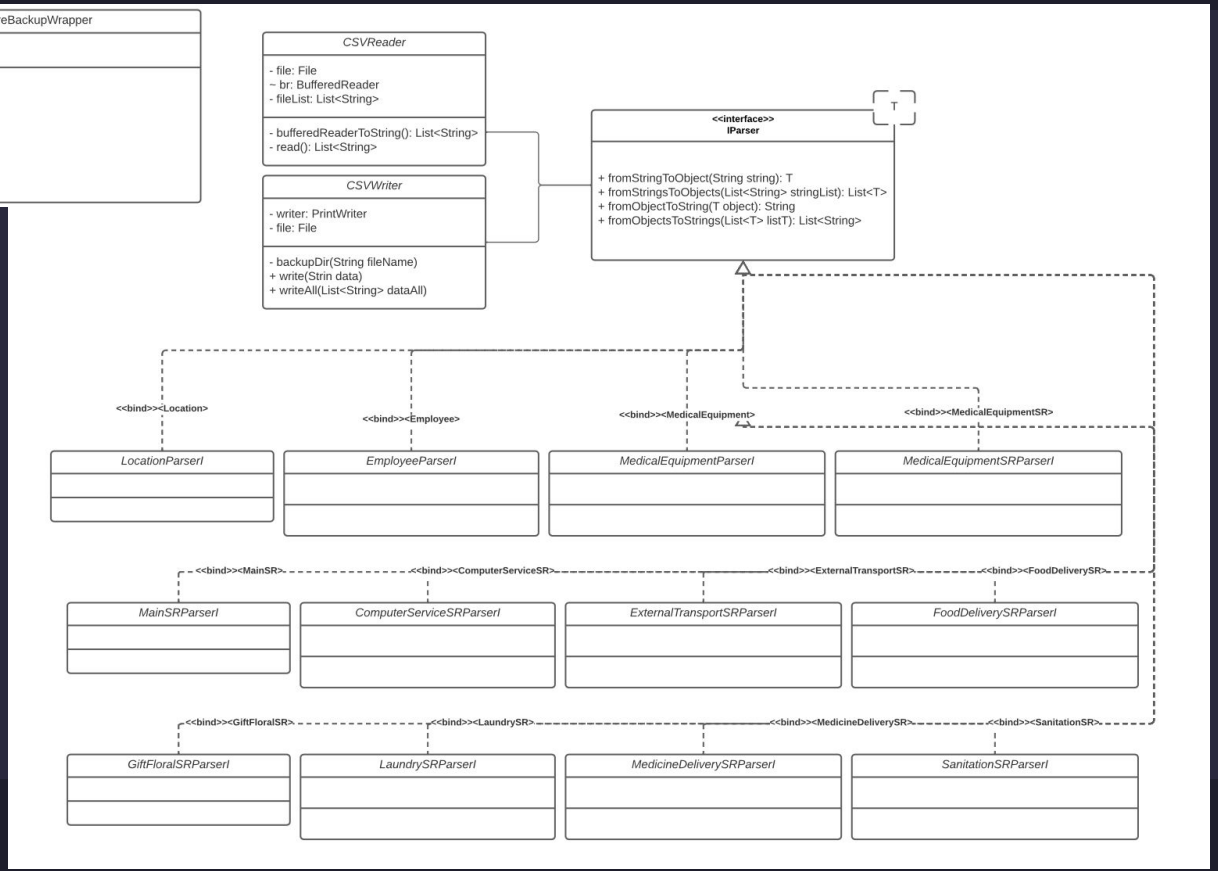




Restore/Backup Subsystem

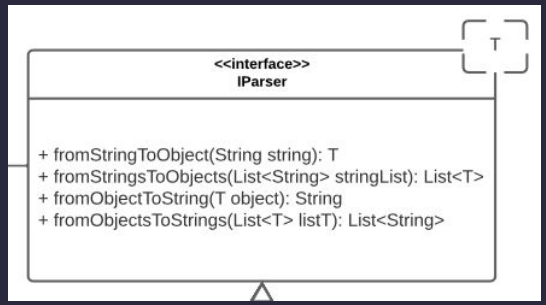


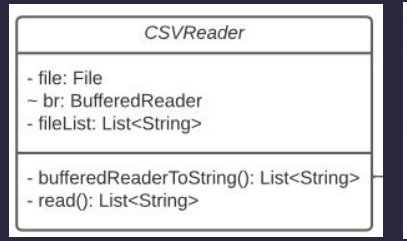
Restore/Backup Subsystem

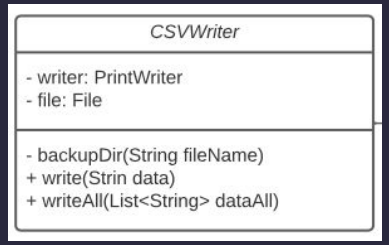


Restore/Backup Subsystem

- Purpose of subsystem was to read from (restore) and write to (backup) CSV files.
- **RestoreBackupWrapper** class
  - Interface between rest of back-end system and the subsystem (Facade pattern)
- **IParser** generic interface class
  - Converted from List of Strings to Collection of Objects
  - Converted from Collection of Object to List of Strings
- Implemented low-level **CSVReader** and **CSVWriter** classes







Demo