Dynamic Response to Urgent Maintenance Requests System Design

Wyatt Kormick

November 14, 2017

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the design of the business layer module of the "Dynamic Response to Urgent Maintenance Requests System" (DRUMRS). This module will be able to receive requests through the interface and turn them into work orders that will provide the necessary information and tracking that FACM Staffers will be able to use to fulfill maintenance tasks.

1.2 System Overview

DRUMRS handles the reporting, documentation, scheduling and progress of maintenance request for campus utilities. Maintenance is handled by the Facilities Management department. The system must handle 2 types of tasks: Repetitive tasks such as weekly cleaning/refilling and immediate tasks such as a leaking toilet. It's important that the maintenance staff is informed of an urgent task.

1.3 Design Objectives

1.4 References

The DRUMRS Requirements Document by Group 20.

1.5 Definitions, Acronyms, and Abbreviations

In addition to the definitions described in the DRUMRS Requirements Document, the definitions of all uses of uncommon or technical acronyms and abbreviations contained in this system will allow the reader to understand this design document.

Term	Definition
BLM	The Business Layer Module of the DRUMRS design described by this document.
BLM Interface	The interface between the business layer module and the outward facing sections of the DRUMR system.
DB	Database

2 Design Overview

2.1 Introduction

The design of the Business Layer Module is done in an object oriented fashion.

2.2 Environment Overview

The entire DRUMR system will be designed in layers, with the BLM being the layer between the front end and the low level systems. This layer will provide the main functionality of the system.

2.3 System Architecture

The BLM will receive requests through the BLM interface, turn the requests into work orders, and use the DB interface to store and retrieve the work orders. It will also use the BLM interface to allow work orders to be edited, scheduled, and assigned to staffers.

2.4 Constraints and Assumptions

3 Interfaces and Data Stores

3.1 System Interfaces

3.1.1 BLM Interface

Using the BLM Interface the front end of the DRUMR system can access the creation, scheduling, and updating of work orders and requests. It also prevents the front end from accessing the underlying functionality. To the front end, the BLM is a black box that just gives it what it needs.

4 Data Stores

4.0.1 DB Abstraction

The DB abstraction allows the BLM to plan for the future implementation of some form of data persistence. It doesn't matter what for that data persistence takes, it only needs to be compatible with the persist and retrieve functions.

5 Structural Design

5.1 Class Diagram

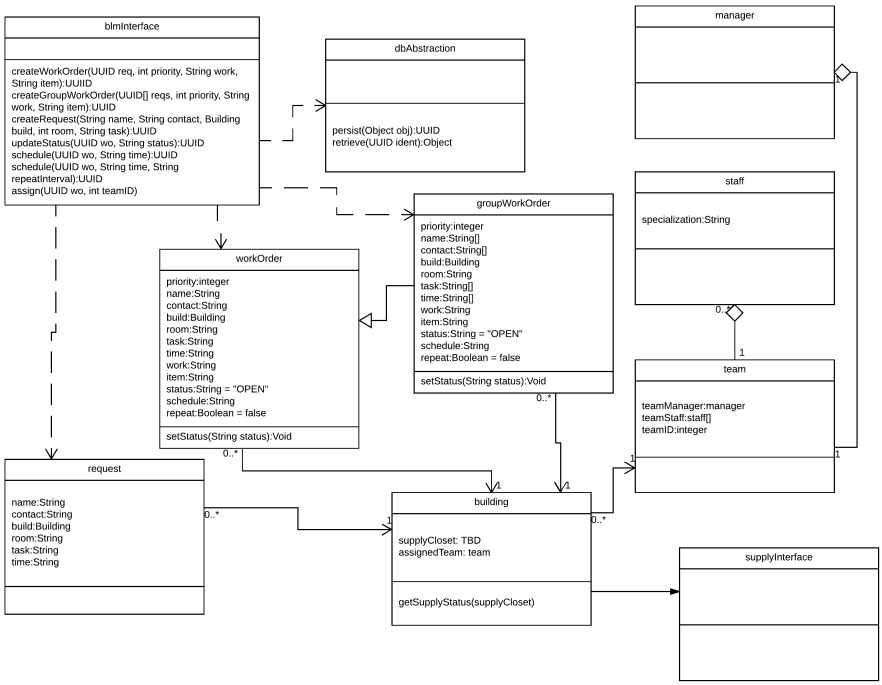

Figure 1: Class Diagram Draft

5.2 Class Descriptions

5.2.1 Class: blmInterface

- **Purpose:** Provide the outward facing functionality of the business layer module
- **Constraints:** None
- **Persistent:** No

Method Descriptions

1. **Method:** createWorkOrder(UUID req, int priority, String work, String item)
   Return Type: UUID
   Parameters:
   - req - the UUID of the desired request object to be turned into a work order
   - priority - the priority level of the work order
   - work - the specializations required to complete the work order

- item - the item(s) the task is to be performed on.

Return value: The UUID identifying the stored, newly created work order
Pre-condition: The request exists in the system as the matching UUID.
Post-condition:
Attributes read/used:
Methods Called: dbAbstraction.persist()

Processing Logic:

2. **Method:** createGroupWorkOrder(UUID[] reqs, int priority, String work, String item)
Return Type: UUID
Parameters:

   - reqs - The list containing the multiple UUIDs of the desired request objects to be turned into one grouped work order
   - priority - the priority level of the work order
   - work - the specializations required to complete the work order
   - item - the item(s) the task is to be performed on.

   Return value: The UUID identifying the stored, newly created work order
   Pre-condition: The requests all exist in the system as the matching UUID.
   Post-condition:
   Attributes read/used:
   Methods Called: dbAbstraction.persist()

   Processing Logic:

3. **Method:** createRequest(String name, String contact, Building build, int room, String task)
Return Type: UUID
Parameters:

   - name - the name of the reporter of the request
   - contact - the contact information of the reporter (an email or phone number)
   - build - the building the request pertains to
   - room - the room in the building that the request pertains to
   - task - the job that the reporter wants done

Return value: The UUID identifying the stored, newly created work order
Pre-condition:
Post-condition:
Attributes read/used:
Methods Called: dbAbstraction.persist()

Processing Logic:

4. **Method:** updateStatus(UUID wo, String status)
   Return Type: UUID
   Parameters:
   - wo - the UUID identifying the workOrder to be updated
   - status - the status to change the workOrder's status to

   Return value: The UUID of the newly changed and stored work order
   Pre-condition: status must be one of "OPEN"—"SCHEDULED"—"COMPLETED"
   Post-condition:
   Attributes read/used:
   Methods Called: dbAbstraction.retrieve(), dbAbstraction.persist(), workOrder.setStatus(), updateReporters()

   Processing Logic:

5.2.2 Class: dbAbstraction

- **Purpose:** Provide the methods for the storage and retrieval of objects.
- **Constraints:** None
- **Persistent:** Yes

Method Descriptions

1. **Method:** persist(Object obj)
   Return Type: UUID
   Parameters:
   - obj - the Object to be stored persistently

   Return value: The identifying unique UUID generated for the object stored.
   Pre-condition: The inputted object has previously been instantiated.
   Post-condition: None
   Attributes read/used: None
   Methods Called: TBD

Processing Logic: TBD

2. **Method:** retrieve(UUID ident)
   Return Type: Object
   Parameters:

   - ident - the UUID for the desired object.

   Return value: The Object pertaining to the inputted UUID
   Pre-condition: The desired object has previously been stored using persist()
   Post-condition: None
   Attributes read/used: None
   Methods Called: TBD

   Processing Logic: TBD

5.2.3 Class: workOrder

- **Purpose:** To model the relevant aspects of individual work orders
- **Constraints:** None
- **Persistent:** No

Attribute Descriptions

1. **Attribute:** priority
   Type: integer
   Description: The priority level of the work order
   Constraints: Must be in the range of valid priority levels (1-4?)

2. **Attribute:** name
   Type: String
   Description: The name of the reporter
   Constraints: None

3. **Attribute:** contact
   Type: String
   Description: The contact information of the reporter (email or phone number)
   Constraints: None

4. **Attribute:** build
   Type: building
   Description: The building that the work order pertains to
   Constraints: build exists as a building in the system

5. **Attribute:** room
   Type: String
   Description: The room that the work order pertains to
   Constraints: None

6. **Attribute:** task
   Type: String
   Description: The problem that the reporter wants done
   Constraints: None

7. **Attribute:** time
   Type: String
   Description: The time stamp for the time that the request for the work order was created.
   Constraints:

8. **Attribute:** work
   Type: String
   Description: The specializations that the work order requires assigned staffers to have to complete it.
   Constraints:

9. **Attribute:** item
   Type: String
   Description: The item that the task is to be performed on.
   Constraints:

10. **Attribute:** status
    Type: String
    Description: The status of the work order
    Constraints: "OPEN"—"SCHEDULED"—"COMPLETED"

Method Descriptions

1. **Method:** setStatus(String status)
   Return Type:
   Parameters:

- status - the status to change the workOrder's status to

Return value:
Pre-condition: status must be one of "OPEN"—"SCHEDULED"—"COMPLETED"
Post-condition:
Attributes read/used: status
Methods Called:

Processing Logic:

5.2.4 Class: groupWorkOrder

- **Purpose:** To model the relevant aspects of grouped work orders. A subclass of workOrder where the re-listed attributes are instead lists.
- **Constraints:** None
- **Persistent:** No

Attribute Descriptions

1. **Attribute:** priority
   Type: integer
   Description: The priority level of the work order
   Constraints: Must be in the range of valid priority levels (1-4?)

2. **Attribute:** name
   Type: String[]
   Description: The names of the reporters
   Constraints: One for each request

3. **Attribute:** contact
   Type: String[]
   Description: The contact information of the reporters (email or phone number)
   Constraints: One for each request

4. **Attribute:** task
   Type: String[]
   Description: Descriptions of the problems that the reporters want solved.
   Constraints: One for each request

5. **Attribute:** time
   Type: String[]

Description: The time stamps for the time that the requests for the work order were created.
Constraints:

5.2.5 Class: request

- **Purpose:** To model the relevant aspects of individual requests
- **Constraints:** None
- **Persistent:** No

Attribute Descriptions

1. **Attribute:** name
   Type: String
   Description: The name of the reporter
   Constraints: None

2. **Attribute:** contact
   Type: String
   Description: The contact information of the reporter (email or phone number)
   Constraints: None

3. **Attribute:** build
   Type: building
   Description: The building that the request pertains to
   Constraints: build exists as a building in the system

4. **Attribute:** room
   Type: String
   Description: The room that the request pertains to
   Constraints: None

5. **Attribute:** task
   Type: String
   Description: The problem that the reporter wants done
   Constraints: None

6. **Attribute:** time
   Type: String
   Description: The time stamp for the time that the request was created.
   Constraints:

5.2.6 Class: building

- **Purpose:** To model the buildings that work orders will pertain to, staffer teams will be assigned to, and will have supply closet inventories. Most of this functionality will be added in a future version of the document.
- **Constraints:** None
- **Persistent:** Yes

Attribute Descriptions

1. **Attribute:** supplyCloset
   Type:
   Description:
   Constraints:

2. **Attribute:** team
   Type: team
   Description: The team that has been assigned to this building
   Constraints:

Method Descriptions

1. **Method:** getSupplyStatus(supplyCloset)
   Return Type:
   Parameters:

   •

   Return value:
   Pre-condition:
   Post-condition:
   Attributes read/used: supplyCloset
   Methods Called:

   Processing Logic:

5.2.7 Class: team

- **Purpose:** To contain the relevant information for a team of staffers, including Managers and Maintenance Staff Members
- **Constraints:** None
- **Persistent:** Yes

Attribute Descriptions

1. **Attribute:** Manager
   Type: teamManager
   Description: The Maintenance Staff Manager of the team
   Constraints: manager exists in the system as a manager object

2. **Attribute:** staff
   Type: staff[]
   Description: A list of maintenance staff members that are on the team
   Constraints: All of the staff members in staff exist as staff objects in the system

3. **Attribute:** teamID
   Type: integer
   Description: The unique identifier for the team.
   Constraints:

5.2.8 Class: manager

- **Purpose:** To contain the relevant information for individual staff managers.
- **Constraints:** None
- **Persistent:** Yes

5.2.9 Class: staff

- **Purpose:** To contain the relevant information for individual maintenance staff members
- **Constraints:** None
- **Persistent:** Yes

Attribute Descriptions

1. **Attribute:** specialization
   Type: String
   Description: The specializations of the staff member
   Constraints:

5.2.10 Class: supplyInterface

- **Purpose:** Provide the interface between building supply monitoring and supply chain functionality
- **Constraints:** None
- **Persistent:** No