

**St. John's University**

**CUS1166 Term Project  
Term Project Task 1**

**Rajiv Bharrat  
Sean White  
St. John's University Computer Science  
2/18/2022**

**Term Project (Task 1): Requirements Document**

## **Preface**

This document is intended to be read by those who would like to implement this system or learn exactly how this system works. This is the first version.

## **Introduction**

SR. Housing for Elders is an independent residence housing for elders with included benefits. These benefits include but are not limited to 24/7 security, 3 daily meals, on-site mechanic, on-site nurse, and more. Because it is a large housing facility for elders, the employees do daily checks on all residences that have not attended for any meals the day of to make sure residents are doing well. Also due to COVID-19, residents must also now be quarantined if they test positive for COVID, are new to the building, or are awaiting negative covid results. The purpose of this project is to simulate a patient management system that will help automate residents management for SR. Housing. The employees of SR. Housings will be the ones using the resident management system and will be required to log in with an employee id and password in order to manage the residents information, requests, and conditions. Based on user level (Administrator or Employee), the user will be able to add or remove residents that are moving into the facility, log events, update quarantine, hospital or missed meals lists, add maintenance requests, add or remove employees, change passwords, and update rent/payroll. When managing resident information the information will have properties for their name, age, date moved in, room number, phone number, and emergency contact number. For the quarantine and hospital lists, it will state their name, quarantine start and end date. For maintenance and maintenance requests, it will just have a description of the request, date filed, and room number.

## **Glossary**

- **GUI** - Graphical user interface.
- **MySQL** - Relational database management system.
- **Backend** - Code that receives requests from clients and contains logic to send appropriate data back to the client.

## **User Requirements**

- Users will be required to login to the system. (This will determine the authorization level of the user)
- User will select from a list of options to manage their inventory based on minimum authorization level
  - Add/Remove Resident (Administrator)
  - Add/Remove Employee (Administrator)
  - Resident Info (Employee)
  - Add/Remove Quarantine List(Employee)
  - Add/Remove Hospital List(Employee)
  - Add/Remove Missed Meal List(Employee)

- Generate Reports (Employee)
  - Whos in quarantine
  - Whos in the hospital
  - Who missed their meals
- Maintenance Request (Employee)
- Nursing Request (Employee)
- Change Password (Administrator)
- Logout (Employee)
- A Graphical User Interface (GUI) that represents an Resident Management System
- The GUI will open up to a screen prompting the user to login with their username and password.
  - Enter username:
  - Enter password:
  - Login
- User will enter their username and password, hit login and the GUI will display all the following options based on their authorization level
  - If its a regular employee, their options will look like:
    - Resident Directory
      - If the user selects this, they can type in a resident's name or room number for basic information about them such as their name, room number, phone number, emergency contacts, if quarantine true or false.
    - Quarantine List
      - If the user selects this, they can add or remove residents to the quarantine list by inputting the residents name, room number, date start and date end. If a patient tests negative before the date ends, they can be manually removed from the list.
    - Hospital List
      - If the user selects this, they can add or remove residents to the hospital list by inputting the residents name, room number, date resident left to hospital, and hospital name.
    - Maintenance Request
      - If the user selects this, they can submit a maintenance request form to the maintenance department by inputting the date filed, room number, and description of problem
    - Nursing Request
      - If the user selects this, they can submit a nursing request form to the clinic by inputting the residents name, room number, date filed, and description of problem
    - Generate Reports

- If the user selects this, it will generate the full quarantine, hospital or missed meal lists
- Logout
  - If user selects this, they will logout and be prompted back to the original login screen
- If it's an administrator, their options will look like the same above but also include:
  - Add / Remove Employee
    - If the user selects this, they can add or remove an employee
  - Add / Remove Resident
    - If the user selects this, they can add or remove a resident
  - Change Password
    - If the user selects this, they can change the password for a user account.

### **System Architecture**

- Login Function
  - The user will type their login credentials into the GUI.
  - The backend java code will take the credentials and check the MySQL database to verify the credentials are valid, and check if the user is an admin or regular employee. If the credentials are not valid the java code will push this to the GUI and the GUI will display to the user to enter valid credentials.
  - If the user is an admin, the backend java code will tell the GUI to display the admin menu, if the user is a regular employee the employee menu will be displayed.
- Add Functions
  - General flow of functions for adding a Resident, Employee, Quarantine List, Hospital List, Missed Meal List will all be the same.
    - User selects to add (Resident Resident, Employee, Quarantine List, Hospital List, Missed Meal).
    - User will type in the name and/or for the Resident, Employee, Quarantine List, Hospital List, Missed Meal List to be added.
    - The system will then access and update the MySQL database with the corresponding information.
    - The system will output a success message to the GUI
    - The GUI will inform the user their addition has been successful.

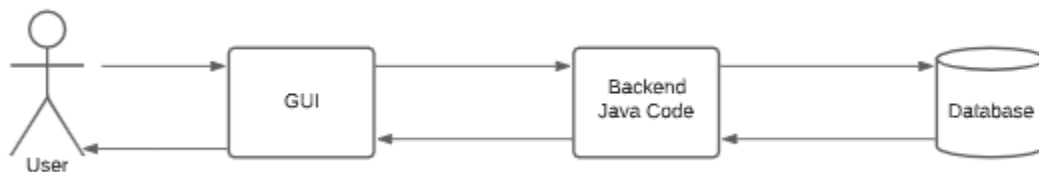
- Delete Functions
  - General flow of functions for deleting a Resident, Employee, Quarantine List, Hospital List, Missed Meal List will all be the same.
    - User selects to remove (Resident Resident, Employee, Quarantine List, Hospital List, Missed Meal).
    - User will type in the name and/or information for the Resident, Employee, Quarantine List, Hospital List, Missed Meal List to be removed.
    - The system will then access and update the MySQL database with the corresponding information.
    - The system will output a success message to the GUI
    - The GUI will inform the user their addition has been successful.
- Generate Reports Functions
  - General flow of functions for generating reports for either quarantine, hospital, or meals will be the same.
    - User selects generate reports option.
    - User selects which type of report they want to generate (quarantine, hospital, meals) and the date for the report they want.
    - Backend java code accesses the MySQL database to get all entries in the corresponding MySQL table for the specified date.
    - Backend java code will push this information into the GUI.
    - GUI will display the report in a table format.
- Resident Info Function
  - User selects resident info option.,
  - User will input the id number of the resident they would like to get info for.
  - The backend java code will access the MySQL database and get the corresponding residents information.
  - The backend java code will push this information to the GUI, which will display the resident info to the user.
- Request Functions
  - General flow of functions for either a maintenance or nursing request will be the same
    - User selects either nursing request or maintenance request option.
    - User inputs the corresponding information for the request.
    - The backend java code pushes this information and adds it to either the nursing or maintenance request table in the MySQL database.
    - The backend java code will push a success message to the GUI.

- The GUI will display that the user's request has been added.
- Change Password Function
  - The administrator selects the change password option.
  - The administrator enters the username of the employee they would like to change the password for.
  - The backend java code verifies the user exists, if they do not the java code pushes this to the GUI and the GUI will tell the administrator to enter a valid username.
  - If they do exist, the administrator is prompted to enter a new password.
  - The backend java code updates the appropriate entry in the employee MySQL table.
  - The backend java code pushes a success message to the GUI.
  - The GUI informs the administrator that they have successfully updated the password.

### **System Requirements Specification**

- The language the program will be written in should be Java and JavaFX.
- All sections and prompts should load quickly.
- The code should be adaptable to any later evolutions.
- Users should enter information that matches the database
- Users should only have access to sections their employee level permits.
- The code should be able to access MySQL database.

### **System Model**



### **System Evolution**

The current use for this system will be in a relatively small facility for seniors. Should the facility greatly expand in numbers or should this system be adopted by a much larger facility some changes could need to be made to the code and database to improve efficiency. Changes could also be made to the GUI, system, and database if any additional services or features are added to the care center that would need to be accessed in the system.

### **Appendices**

- Hardware Requirements
  - Windows or MacOS required
  - Java Installed and Running
  - MySQL Database Installed on Computer
- Database Requirements
  - System will grab information from the database when requested:
    - Resident info (name, age, room number, moved in date, etc)
    - Employee info (name, age, position, password)
    - Update, add, or remove quarantine lists (name, age, quarantine true or false)
    - Update, add, or remove hospital lists (name, age, hospital name, date left for hospital)
    - Update, add, or remove missed meals list (name, room number)
    - Update payroll info (name, hours worked, amount owed, paid true or false)
    - Update rent info (name, room number, rent amount, paid true or false)
    - Change password (username, password)

### **Index**

A	J	S
Administrator 2,3,4		
B	K	T
C	L	U
		User 2,3,4,5
D	M	V
	MySQL, 4	

E	N Nurse, 2 Nursing, 3	W
F	O	X
G Generate Report 2,3 GUI 3	P Password, 2,3,4	Y
I	R Report, 2,3	Z