**Use-Case #1 Adding, Removing, and Modifying Employee Information**

Characteristics

|  |  |
| --- | --- |
| Intent | The user will be able to enter, remove, and update employee information. |
| Author | Sam DeNucci |
| Last Update | 10/20/19 |
| Status | Conceptual |
| Preconditions | Data structure containing employee information is running. |
| Trigger | User navigates to “employee” info page, and clicks either the “add” employee button, “remove” employee button, or the “modify” employee button. |
| Success Condition | The new information is either added, removed, or modified. |
| Failed Condition | If the user cancels, or exits the program, the information will not be saved, and the program will revert back to the last state of the database / document. |
| Operations Concept | This will use either an embedded database or a document that the program can read from and write to. If using a database, the GUI will use add, remove, and modify SQL statements to change information. If using a document, the GUI will change what is written in the document. |
| Overview | The user will be able to use a GUI to add, remove, and modify employee information. The GUI will display a list of employees and information. Next to each employee’s info will be a remove and modify button. If the remove button is clicked, the employee’s information is removed. If the modify button is clicked, the user will be able to change the information assigned to that employee. An add button will be at the bottom of the GUI. If the user clicks the add button, an empty employee is added to the GUI. The user can then enter the employees information, and confirm the information. The GUI will also have a save button that commits any added employees, removed employees, or modified employee information to the data structure. |

Main Success Scenario

|  |  |
| --- | --- |
| Step | Action |
| 1 | User clicks on employee information tab / button on the main screen |
| 2 | User clicks the add employee button, the remove employee button, or the modify button |
| 3 | User is able to enter new employee information, modify employee information, or delete an employee’s information |
| 4 | User clicks the save button and the new information is saved to the database / document |

**Use Case # 2 Enter Day Data**

Characteristics

|  |  |
| --- | --- |
| Intent | Allow the user to input shift data for a set of days (weekend, and weekday) |
| Author | Sam DeNucci |
| Last Update | 10/20/19 |
| Status | Conceptual |
| Preconditions | Data structure containing day information is running. |
| Trigger | User navigates to “day” information page and clicks on modify days button on either the weekend set, or weekdays set |
| Success Condition | User is able to change and save day information |
| Failed Condition | If the user cancels, or quits the program, information will not be saved |
| Operations Concept | This will use either an embedded database or a document that the program can read from and write to. If using a database, the GUI will use modify SQL statements to change information. If using a document, the GUI will change what is written in the document. |
| Overview | This will allow the user to manage the weekdays and weekends information. This means the user will be able to modify how many shifts are needed on these days, and which types of shifts are needed. |

Main Success Scenario

|  |  |
| --- | --- |
| 1 | User navigates to the days information page |
| 2 | User clicks modify on either the weekends, or on the weekdays |
| 3 | User enters new information |
| 4 | User clicks save and information is saved to the database / document |

**Use Case # 3 Create a Month Calendar**

Characteristics

|  |  |
| --- | --- |
| Intent | Generate a monthly calendar using calendar theory. Have each day be able to contain the people working, the shifts they are working, and the shifts needed. |
| Author | Sam DeNucci |
| Last Update | 10/20/19 |
| Status | Conceptual |
| Preconditions | The program is running |
| Trigger | The user clicks on the create calendar button, enters the month, and the year |
| Success Condition | An empty calendar is created with the proper days and dates |
| Failed Condition | If the user exits the program, the schedule will not be saved with new information. |
| Operations Concept | Using calendar theory, generate a calendar object. The calendar object will contain day objects. Each day object will contain an empty array of employees working, shifts filled, and shifts needed. |
| Overview | The user will click the generate schedule button. They will input the month and year of the calendar to generate. The program will then generate an empty calendar for the given month. In the GUI, this will appear as the calendar month name and year on the top, and a grid below with employee names on the vertical axis, and the day on the horizontal axis. Each day will have a set number of shifts to fill based on the day information (see use case 2) |

Main Success Scenario

|  |  |
| --- | --- |
| 1 | User clicks the generate calendar button |
| 2 | User enters month and year |
| 3 | Monthly Calendar is generated and displayed |

**Use Case # 4 Create a Schedule**

Characteristics

|  |  |
| --- | --- |
| Intent | Generates a schedule when the user presses the generate schedule button |
| Author | Sam DeNucci |
| Last Update | 10/20/19 |
| Status | Conceptual |
| Preconditions | All employee information is saved to the employee information page, all day information is saved to the day information page, and a monthly calendar has been created. |
| Trigger | User clicks auto generate schedule button on the calendar page |
| Success Condition | All shifts on each day are filled based on the day information and employee information. |
| Failed Condition | The scheduler does not have enough employees to fill all shifts. The schedule will generate as much as possible, but be incomplete |
| Operations Concept | The scheduler will run through each day, find which shifts need to be filled, and find an employee to fill it based on the employee information provided. This will fill the arrays in the days of the calendar |
| Overview | The user will click the auto generate schedule button. The schedule will go through each day and assign the necessary shifts to employees. This will be based on the information provided in the employee information page and the day information page. It will then display the shift where the x = employee name, y = day / date |

Main Success Scenario

|  |  |
| --- | --- |
| 1 | User clicks the auto-generate schedule button |
| 2 | Scheduler runs through each day and assigns employees to shifts |
| 3 | Calendar displays shifts |

**Use Case #5 Manually Fill Schedule**

Characteristics

|  |  |
| --- | --- |
| Intent | Allows user to fill shifts that weren’t filled based on the information provided to the auto-scheduler |
| Author | Sam DeNucci |
| Last Update | 10/20/19 |
| Status | Conceptual |
| Preconditions | The monthly calendar has been created |
| Trigger | User clicks on a coordinate (where x = employee name, y = day / date) |
| Success Condition | The user is able to input the shift they want worked that day, and the information is added to that days information |
| Failed Condition | User is unable to enter information for the given date |
| Operations Concept | The user clicks on a coordinate, and either types or selects a shift from a drop down. The arrays in the calendar object are then updated |
| Overview | In the event that the provided employee information is not enough to fill the necessary number of shifts, the user will have to manually assign shifts to users. The user will be able to click a coordinate, and then assign the desired shift. |

Main Success Scenario

|  |  |
| --- | --- |
| 1 | User clicks on a coordinate |
| 2 | User enters / picks a shift for that employee |
| 3 | Schedule adds information to the day’s arrays |