Summary Use Cases

Admin/owner

**Promotion/ Demotion**

Level: Summary

Actors: Owner

Goal: To change the system access level of a Employee

Activities: The Owner will log into the system and have the option to promote and demote employees in the system. There will be a list of all employees with their role or access level and the ability to easily decrease/demote a manager to worker status or increase/promote an worker to manager status, after all changes are selected, they will be processed and sent to the database.

Quality: : Should work to a high standard insofar as it always works correctly, Owner functions will only be seen by a few people, the need to work properly (obviously) but do not need to be fancy.

Version: Oct 20, 2014, amm215

**Add New Employee**

Level: Summary

Actors: Owner

Goal: Add a employee to the system

Activities: The Owner will login into the system, the owner will then enter all the important information that the system requires about a Worker: first name, last name, email, wage, access level. As well, he will create a new unique identifier and password for the user, some kind of employee number or other identification, the system will check if this value is unique or not and notify the Owner. Once all the information is filled in, a new Employee account will be added to the database.

Quality: Should work to a high standard insofar as it always works correctly, not necessarily in a fancy manner.

Version: Oct 20, 2014, amm215

**Give away shift**

Level: Summary

Actors: Database, Employees

Goal: Send a request to all available employees in a particular timeslot

Activities: Person logs in, clicks one shift from any calender view and presses the give button. The database will find out which employees are available at the specified time and send each of them a message. After a worker accepts the shift, their manager has the option to approve the request.

Quality: This is one of the primary functions of the program, It should work correctly to the highest of quality.

Version: Oct 21, 2014 cbl013

**Check Messages**

Level: Summary

Actors: Employee

Goal: Display all recent messages correctly to the requester

Activities: After logging in, an employee can scroll through their messages in the home screen. The database passes these messages back after a request during login initialization.

Quality: Not highly important. Chronological display is the only requirement.

Version: October 21, 2014 cbl013

**Check Schedule**

Level: Summary

Actors: Employees

Goal: Communicate an employees current schedule to them

Activities: The employee will log into the system and on the main screen of the system, a basic view of their weekly schedule will be displayed. If they click on the basic display a bigger more detailed display of the weekly or monthly shifts will be displayed. The shifts of other users will be visible on this larger display.

Quality: As the main use case of the program, this will be of the highest quality

Version: Oct 21, 2014 wwf594

**Manager Assigns Shifts**

Level: Summary

Actors: Manager

Goal: Employee is assigned a shift

Activities: Manager will log into the system and create a shift, they will have a list of employees that they can assign it to.

Quality: High standard, part of the backbone of the program. Should work correctly, doesn’t necessarily have to be fancy as only the manager sees it.

Version: Oct. 21, kts192

**Check Pay/Wage**

Level: Summary

Actors: Employees

Goal: Employee logs in, checks there pay for some period. Also sees current wage.

Activities: Employee logs in, can check current wage, and pay for some period. Manager/Owner can adjust employees wage, indirectly adjusting the wage for upcoming shifts.

Quality: Low standard, not essential to program, and without capturing login/logout times, is not accurate. Only provides estimate.

Version: Oct. 21, kts192

**Add New Employee**

Scope: Administration function of the Shift Swap System

Level: User-goal

Primary Actor: Owner

Stakeholders and Interests:

* Owner: Wants a quick and easy way to create and add a employee profile to the system
* Employee: accurate and easy to access information
* Company: Up-to-date information on all employees that is accurate and correct.
* Managers: access to new employee information as soon as possible to add them to the working schedule

Preconditions: Owner is logged into the program

Success Guarantee (Postconditions): New employee is added to the database system, username is unique, all necessary information is filled in.

Main Success Senario:

1. Owner recieves or has access to new employees information (whether the new person is in the room or information has been recieved by email, etc)
2. Owner Starts a Add New Employee Transaction
3. Enters all of the new Employees information
4. Owner confirms the addition of a new Employee to the System
5. Information is sent through the System and added to the database for access by others

Extentions and Alternate Flow:

1. Owner Cancels the Add New User Transaction
2. Owner presses the cancel button at any point before confirmation
3. All partially filled out information is discarded
4. No changes are made to the system
5. Input username is not unique
6. Owner will be notified if the given username is already taken in the system
7. Owner will be prompted to change the given username
8. Owner will not be able to confirm new addition until a unique username is given
9. Connection to the database is refused/lost/dropped
10. System restarts the database connection
11. System attempts to reconnect to the database

2a. If successful, continue with transaction

2b. If not successful prompt to try connection again or cancel the transaction

1. Not all necessary information is present on confirmation
2. Owner misses filling in one of the required fields
3. Owner is promted to fill in the missing fields or cancel transaction

2a. Owner Cancels the transaction

2b. Owner fills in the missing information and the trasaction continues

Special Requirements:

* None

Technology and Data Variation List:

* None

Frequency of Occurence: Infrequently depending on size of staff and number of firings and hireings

Open Issues:

* Could (or should) this data be input by the new employee directly
* Is there any customization for different businesses

Version: Oct 20, 2014, amm215

**Manager Assigns Shifts**

Scope: Administration function of the Shift Swap System

Level: User-goal

Primary Actor: Manager

Stakeholders and Interests:

- Manager: Simple way to assign shifts

- Employee: Up-to-date access of shifts and notifications of new shifts via messages.

Preconditions:

- Manager is logged into the system, there exist employees to take some shifts.

Success Guarantee (Postconditions):

- Shift has been assigned to an employee (the shift is in the database)

Extentions and Alternate Flow:

- Manager cancels the shift creation.

- Presses Cancel button

- partial info is discarded

- no changes made

- Shift overlaps with another shift

- give warning, return to shift creation

- Connection to the database is refused/dropped

- System attempts to reconnect

- If it fails to connect, prompt “failure” message, transaction cancelled.

- Missing employee or shift time

- Prompt for missing information

Special Requirements: none

Technology and Data Variation List: none

Frequency of Occurrence: daily/weekly as shifts are needed

Open Issues:

Versions: Oct. 21, kts192

**Done User Login**

**done done DONE Check schedule\***

**Done Check messages**

**Done User wants to trade particular shift**

**(fully done done)Give away shift \***

**Done) Manager assigns shifts\***

**Done done done Manager approves shift change\***

**Done Check pay or wage**

Fully Dressed Use Cases

**Add New Employee**

Scope: Administration function of the Shift Swap System

Level: User-goal

Primary Actor: Owner

Stakeholders and Interests:

* Owner: Wants a quick and easy way to create and add a employee profile to the system
* Employee: accurate and easy to access information
* Company: Up-to-date information on all employees that is accurate and correct.
* Managers: access to new employee information as soon as possible to add them to the working schedule

Preconditions: Owner is logged into the program

Success Guarantee (Postconditions): New employee is added to the database system, username is unique, all necessary information is filled in.

Main Success Senario:

1. Owner recieves or has access to new employees information (whether the new person is in the room or information has been recieved by email, etc)
2. Owner Starts a Add New Employee Transaction
3. Enters all of the new Employees information
4. Owner confirms the addition of a new Employee to the System
5. Information is sent through the System and added to the database for access by others

Extentions and Alternate Flow:

1. Owner Cancels the Add New User Transaction
2. Owner presses the cancel button at any point before confirmation
3. All partially filled out information is discarded
4. No changes are made to the system
5. Input username is not unique
6. Owner will be notified if the given username is already taken in the system
7. Owner will be prompted to change the given username
8. Owner will not be able to confirm new addition until a unique username is given
9. Connection to the database is refused/lost/dropped
10. System restarts the database connection
11. System attempts to reconnect to the database

2a. If successful, continue with transaction

2b. If not successful prompt to try connection again or cancel the transaction

1. Not all necessary information is present on confirmation
2. Owner misses filling in one of the required fields
3. Owner is promted to fill in the missing fields or cancel transaction

2a. Owner Cancels the transaction

2b. Owner fills in the missing information and the trasaction continues

Special Requirements:

* None

Technology and Data Variation List:

* None

Frequency of Occurence: Infrequently depending on size of staff and number of firings and hireings

Open Issues:

* Could (or should) this data be input by the new employee directly
* Is there any customization for different businesses

Version: Oct 20, 2014, amm215

**Give Away Shift**

Scope: Employee function for self-shift-management

Level: User goal

Primary Actor: Employees

Stockholders and Interests:

* Employees : May help out coworkers by giving them a break when they need it
* Shift Giver: Want to have a “day off” or lighten their schedule where they would not normally, and wants to have the greatest chance of succeeding in their effort.
* Manager : wants users to handle their own affairs, while retaining oversight

Preconditions: Giver must have the shift they wish to give away, and the receiver(s) must not.

Success guarantee:

* The giver will no longer work on the shift they gave away (schedule changed)
* The receiver will now work the shift they accepted
* The manager will be asked to approve the transaction
* All parties will be notified that the trade was complete

Main Success Scenario:

1. Employee logs in
2. Employee clicks on a day they really want time off
3. Employee sees a dialogue containing information about that day, and clicks the “give” button
4. Another employee who is logged in after this has happened sees that they are able to take someone else’s shift, if they so choose, in their messages box
5. They click the message and then the “accept” button
6. The manager receives notification of the request
7. The manager opens the message and clicks “approve”
8. The transfer is made in the database
9. All parties are notified of the new state of affairs

Extension and Alternate Flow:

* Database Unavailable (trade impossible)
* Manager Unresponsive (as someone else?)
* User doesn’t work on the day they want to give away (no “give” button)
* Manager Disapproves (not a failure state)
* No workers accept the request (request Timeout)

Glossary

Controller – Brains of the program, all requests that are meant for the database are sent through the controller first, and similarly all information coming from the database are first sent through the controller. The controller parses requests and the information that comes back.

Owners – The user with the most administrator type role in the system, this user has access to all the information in the system as well as the ability to add and remove other users and change access level and other settings of the system.

Workers – basic users of the system, the regualar employees of the company, these users have limited access to change information, but can see their schedule and requests trades among other things

Managers – Mid level users in the system, can see limited personal information on all other users as well as be able to add and drop shifts for any and all other users.

Employee – general term for workers and managers and owners

Requests – any time the program tries to access the database information in some way, a request is generated and sent to the controller for parsing

Access Level – There are 3 basic levels of access in the system 1 for workers, 2 for managers, and 3 for owners. Each level of access provides different looks at the system and the information within