

# LineUp

## Team Aphrodite

*Nora Allison (efa91)*

*Bryan Djunaedi (dju90)*

*Stephen LaPlante (laplansk)*

*Evan Leon (ejl6)*

*Thomas Rothschilds (tgr4)*

*Simone Schaffer (simone09)*

*Nicholas Shahan (nshahan)*

*Evan Whitfield (evanw2)*

*UPDATED 5/31/14*

*Requirements:*

*Use Cases*

## CHANGELOG

5/31/14 – No changes

5/16/14 – No changes.

5/5/14 – updated with additional use cases to reflect three main types of users. Also updated to reflect changes in feature set and increased knowledge of product architecture.

## Use Cases

For any queue to properly function, it must be able to properly maintain a FIFO ordering during its execution. Based on this we have designed three use cases that are integral to proper queue functioning: creating a queue, enqueueing, and dequeuing. These basic functions also provide underlying functionality to more complex operations (eg. de/enqueueing at specific locations, creating already filled queues, group queuing). Without these basic features our product simply would not function.

The three primary use cases we have outlined also highlight the three different permission levels we plan to support. These are Admin, Employee, and User.

With these basic functions sharing and locating a queue can be performed outside of our system by simply sharing a URL. More advanced features include making a queue visible for search inside our system and finding a queue by search, recommendation features or by location.

### **An Admin user creates a new Queue.**

Preconditions: User is allowed to make queues (optional: login required)

Active Actor: Admin User

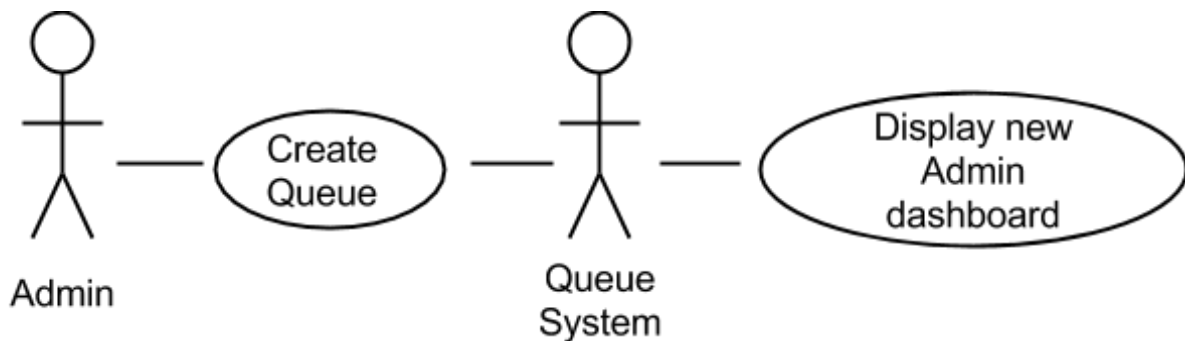
Other Actors: Queue System

Triggers: Admin chooses to create a new Queue

Postconditions: A new Queue now exists in the system.

Failure End Conditions: Admin does not provide the necessary information and/or cancels action

Extensions/Variations: none at this time



1. An admin user enters the data to create a new Queue (optional: login required).
2. The system presents the admin dashboard view.

### **An Employee (user with the permissions to dequeue) removes the next person in line from an existing Queue.**

Preconditions: Queue is not empty (optional: login required and active actor is a user with permission to dequeue)

Active Actor: Employee

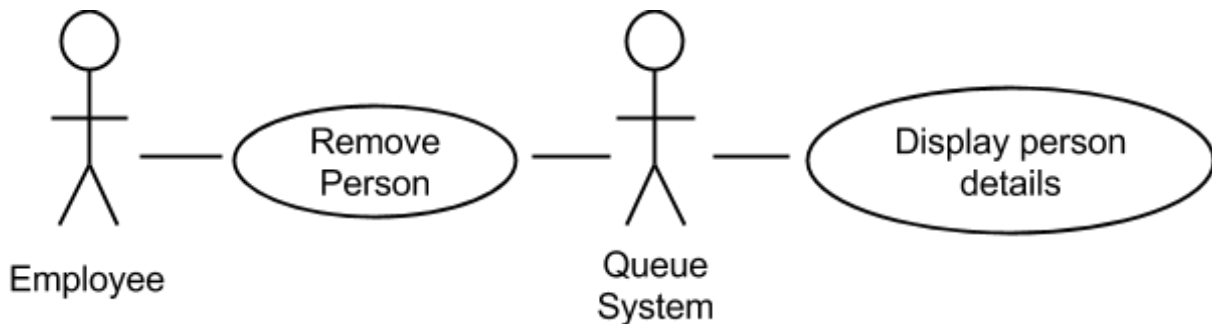
Other Actors: Queue System

Triggers: Employee chooses to dequeue the next person in the Queue

Postconditions: The person at the front of the Queue has been removed

Failure End Conditions: none at this time

Extensions/Variations: Employee wants to remove a person not at the front of the Queue (outlined in an additional use case below)



1. The Employee chooses to remove next person from the Queue (possible site login required).
2. The system displays the details of the next person in line.

**A User (person to wait in a Queue) adds herself to a Queue.**

Preconditions: Queue exists (optional: Queue is not full and User has not been blocked from the queue)

Active Actor: User

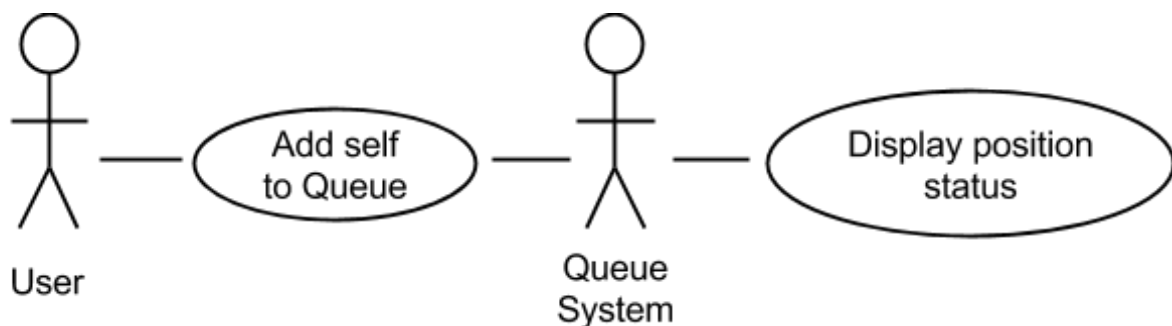
Other Actors: Queue System

Triggers: User chooses to add herself to a Queue

Postconditions: The User is now at the end of the Queue

Failure End Conditions: User does not provide the necessary information and/or cancels the action

Extensions/Variations: Employee adds person at specific location in queue (outlined in an additional use case below)



1. The user adds her name to an existing Queue.
2. The system displays a position status screen to track her place in the Queue.

**Additional Use Cases:** These use cases provide additional functionality that will be desired for a full featured product. These are beyond the basic functionality of a simple online queue.

### **An Admin views and possibly changes the settings of a Queue**

Preconditions: Queue exists and the user has Admin permission

Active Actor: Admin

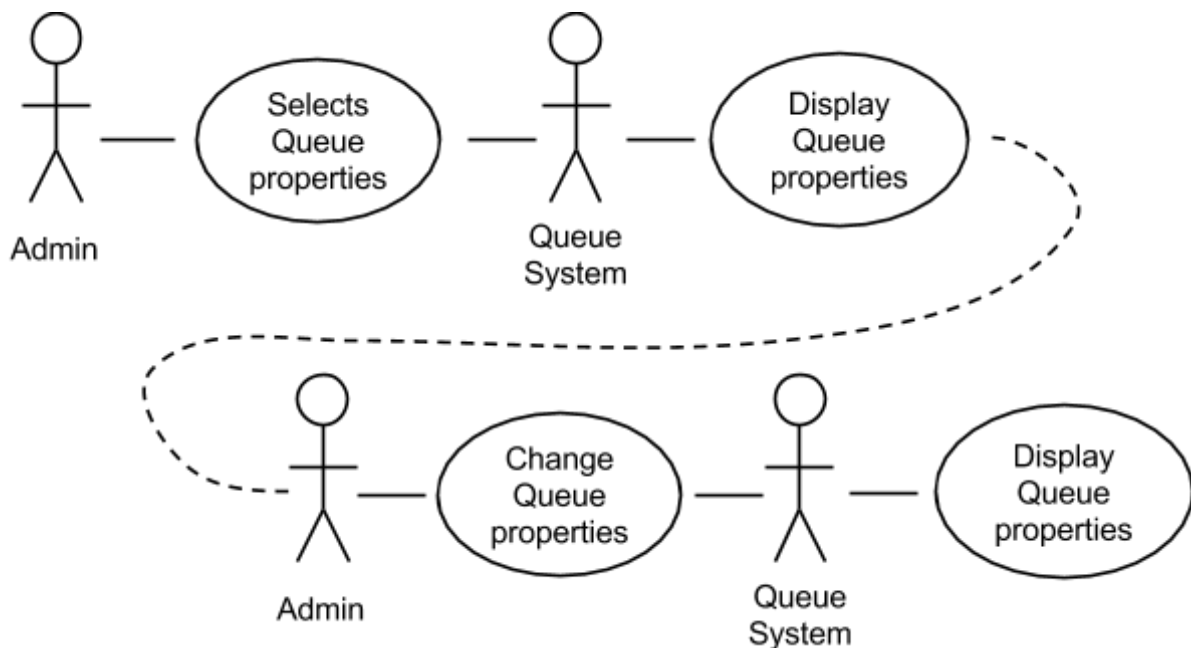
Other Actors: Queue System

Triggers: Admin selects the Queue properties option

Postconditions: Queue information is displayed

Failure End Conditions: none at this time

Extensions/Variations: none at this time



1. Admin selects queue from queue listing in admin panel
2. Admin selects display queue properties
3. The system displays the queue information

OPTION:

4. Admin optionally changes a setting
5. The system displays the new queue information

### **An Employee removes a specific User (not at the front) from a Queue**

Preconditions: Queue exists, Employee has access for this action, Desired User is in the Queue

Active Actor: Employee

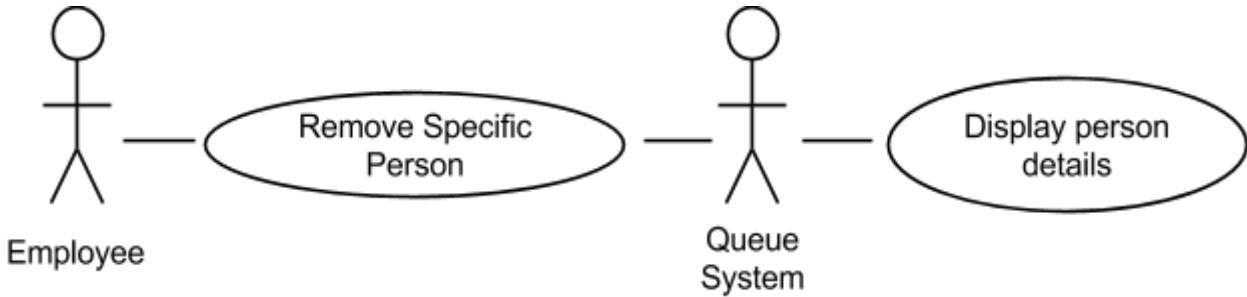
Other Actors: Queue System

Triggers: Employee selects to remove a specific User from the Queue

Postconditions: The specific User is no longer in the Queue, all others remain in FIFO order

Failure End Conditions: none at this time

Extensions/Variations: none at this time



1. The Employee locates and chooses a specific person to remove from the Queue (possible site login required).
2. The system provides the details of the removed person.

### **An Employee adds a User to a specific location in Queue (VIP Enqueue)**

Preconditions: Queue exists and Employee has permission (optional: Queue is not full, User is not blocked from the queue)

Active Actor: Employee

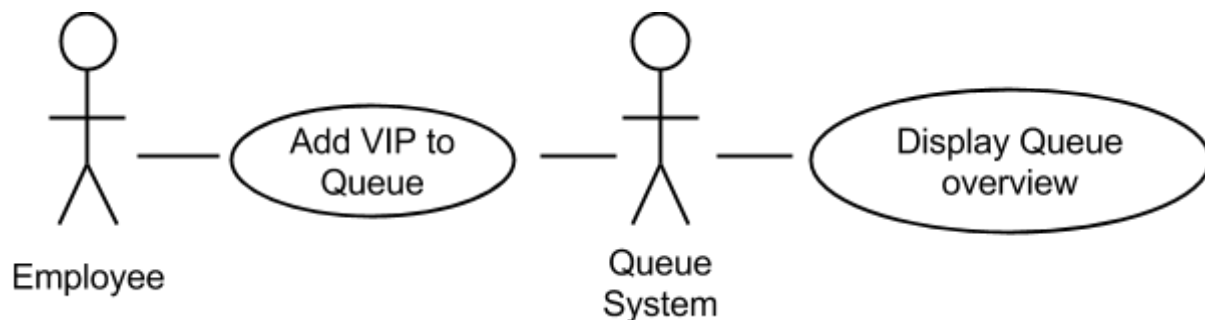
Other Actors: Queue System

Triggers: Employee selects to add a User to a specific position in the Queue

Postconditions: Queue order remains unchanged, except that the specific User has been inserted into the desired position.

Failure End Conditions: none at this time

Extensions/Variations: none at this time



1. The Employee adds VIP name to an existing Queue and selects target location in queue.
2. The system provides a position status screen to track her place in the Queue.

### **An Admin shares a Queue with Users**

Preconditions: Queue exists (optional: User is not blocked from the queue)

Active Actor: Admin

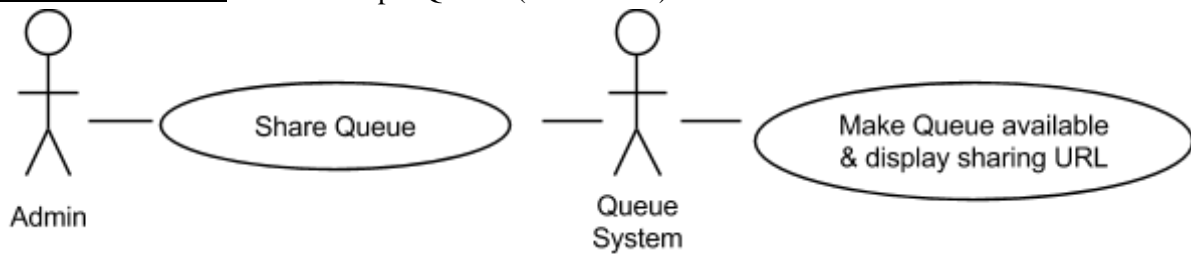
Other Actors: Queue System

Triggers: Admin selects that he wants to share this queue.

Postconditions: Queue remains unchanged, ready for Users to add themselves

Failure End Conditions: Insufficient information provided by Admin

Extensions/Variations: Share multiple Queues (batch share)



1. The Admin selects to share a Queue
2. The Admin enters needed information to publicize the Queue
3. The system displays the makes the Queue available for searching and provides a direct link for sharing.

### **An unknown person creates a new account**

Preconditions: Desired username does is not already in use

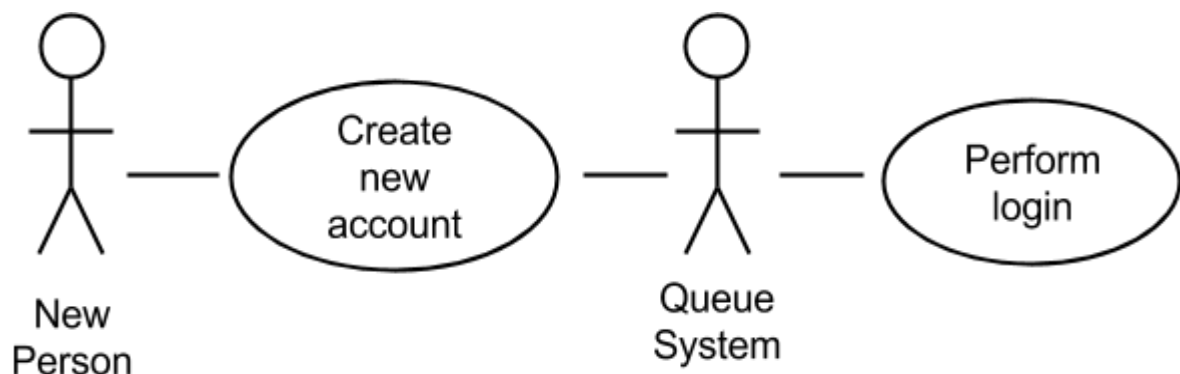
Active Actor: Person

Other Actors: Queue System Triggers: Person selects create new account

Postconditions: New account has been created and the log in use case has been performed automatically

Failure End Conditions: Person does not provide the required information

Extensions/Variations: none at this time



1. A new person chooses to create an account
2. The person enters the required information
3. Queue System automatically performs the login use case on behalf of the Admin/User.

### **An Admin grants Employee account access to a Queue**

Preconditions: Desired Employee does not have access to a specific queue

Active Actor: Admin

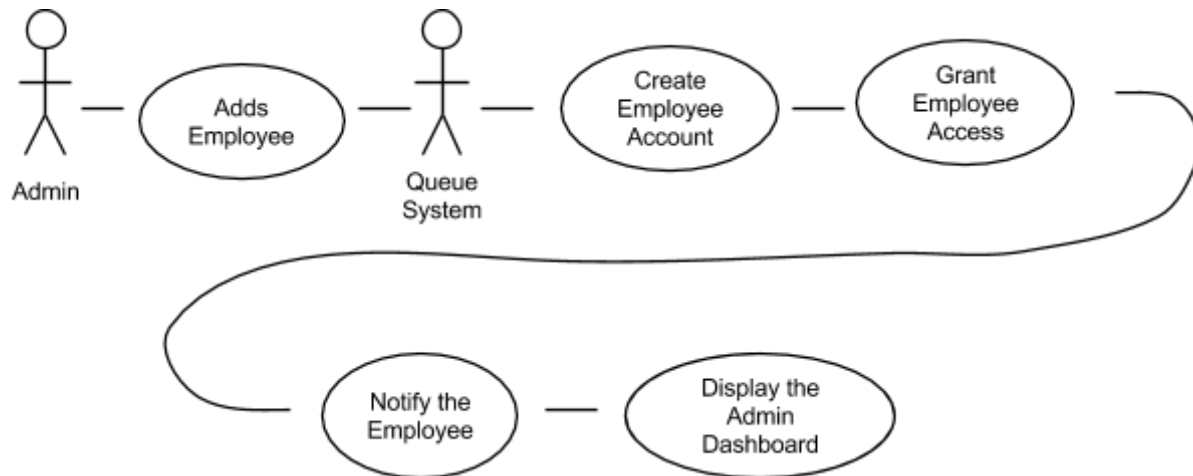
Other Actors: Queue System

Triggers: Admin chooses to add new Employee account to a specific Queue

Postconditions: Employee now has access to add, remove Users from the Queue

Failure End Conditions: Admin enters incorrect information for the Employee

Extensions/Variations: Admin creates the Employee account at the same time, Admin adds grants Employee access to multiple Queues at the same time.



1. Admin chooses to create a new Employee in Admin Dashboard
2. Admin enters the required information, including which Queue to grant Employee permission on
3. The Queue System creates a new Employee account
4. The Queue System notifies the Employee of the access
5. The Queue System displays the Admin Dashboard to the Admin and denotes success.

### **An Admin/Employee/User logs in to Queue management system**

Preconditions: Admin/Employee/User account already exists

Active Actor: Unidentified User

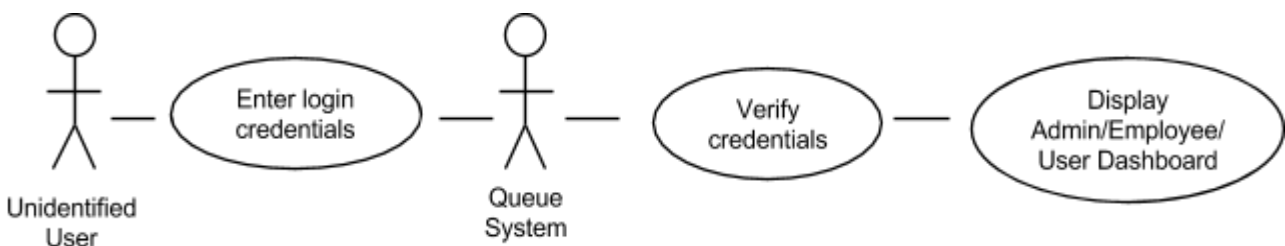
Other Actors: Queue System

Triggers: Clicking login button

Postconditions: Admin/Employee/User now identified and can access their Admin/Employee/User dashboard

Failure End Conditions: Admin/Employee/User enters incorrect credentials

Extensions/Variations: none at this time



1. Person submits their login credentials
2. The Queue System verifies the credentials
3. The Queue System displays the correct Dashboard to the now identified User

### **A User locates a Queue**

Preconditions: User logged in, queue already exists

Active Actor: User

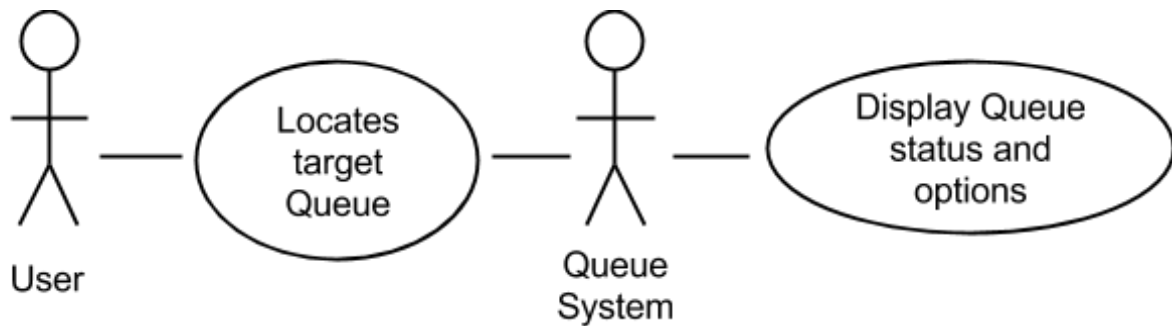
Other Actors: Queue System

Triggers: User selects the queue listing

Postconditions: User is on the queue information section of the panel, with the target queue displayed

Failure End Conditions: none at this time

Extensions/Variations: none at this time



1. User selects the Queue listings
2. User finds the target Queue among the possible Queues
3. The Queue System displays the Queue information and options to the User