**Assumptions**

**Unit Tests**

**Note: Those classes not included only had getters**

* Test getters of system classes
* Test User
  + Call addPoints()
    - Expected: The user’s points should be updated by the amount specified.
  + Call login()
    - Expected: The user should be logged in. All of the user’s information will be viewable and they will be able to access and make changes to their information.
  + Call getUserInfo()
    - Expected: Return userId, password, email, address, points and name.
* Test Administrator
  + Call deleteReview()
    - Expected: Review with specified attributes should no longer be visible. For example: Should be able to delete reviews by a certain user on a specified date and time.
  + Call banUser()
    - Expected: Administrators should be able to ban users so that they can no longer post photos and comments.
* Test Moderators
  + Call reviewImage()
    - Expected: Moderators should be able to determine whether photos are appropriate and will be viewable.
  + Call deleteImage()
    - Expected: If images do not meet site requirements they should be able to be deleted. The user will be notified.
* Test Customer
  + Call updateInfo()
    - Expected: The appropriate fields should be updated.
* Test Gaming
  + Call trackLocation()
    - Expected: The user’s location is returned.
  + Call setPoints()
    - Expected: The user’s points should be updated by the specified amount.
* Test Review
  + Call addReview()
    - Expected: A new review should be created and will be viewable by moderators.
  + Call EditReview()
    - Expected: Details of a review should be changed.
* Test Cart
  + Call addItem()
    - Expected: A new item should be added and will be visible in the customer’s cart.
  + Call checkout()
    - Expected: A customer should be forwarded to checkout where they can enter their payment and billing information.
  + Call deleteItem()
    - Expected: The specified item should be removed from the cart and will no longer be visible.
* Test Payment
  + Call validateInfo()
    - Expected: Customer billing information should be validated after the user inputs the information.
  + Call submitPayment()
    - Expected: The user’s billing information should be accepted and processed.
  + Call printInfo()
    - Expected: The user’s billing, payment and order info should be available to print.
* Test Order
  + Call createOrder()
    - Expected: A new order should be created using information submitted by the user during checkout.
  + Call submitOrder()
    - Expected: Order information should be submitted so that it can be stored for future reference.

**Integration Test**

* Test User interaction with Social Media
  + Is the user’s social media information appropriately stored?
* Test User interaction with Gaming
  + Does getPoints() update points in the User object
* Test User interaction with Review
  + Is a User able to create a review? (createReview() will be called)
* Test User interaction with Cart
  + Is the User able to add items to the cart? (addItem() will be called)
  + Is the User able to proceed to checkout? (checkout())
* Test Cart interaction with Ticket
  + Can a ticket be added to the cart?
* Test Cart interaction with Donation
  + Can a donation be added to the cart?
* Test Cart interaction with Order
  + When user submits information through checkout is the order created? (createOrder())

**System Test Procedure**

* *Assume a web-based GUI supporting rental system app process.*
* *All info will be persisted in a DB*
* Questions to Answer in Test:
  + Can all of the classes interact with each other in the Rental System?
  + Can a User object act as a customer object when purchasing Tickets or Donations?
  + Can a User object access a Review object? Moderator object?
  + Can a User object access a Social Media object?
  + Can a user object access past Orders?

**Regression Test**

* Base on prioritization method (retest high priority)
* Resource limitations so test based on priority
* Highest priority regression testing
  + User application should be tested first to make sure that if anything else in the system is debugged properly. No other functionality is broken.
  + User login should be tested next to test that User information is viewable and accessible.
  + Cart and Orders should be tested so that Users can immediately make purchases.
  + The Park directory should also be completely functional.