

Jaunt Specific Test Plan

We will be testing all models, controllers, and views of the system. The models that will be tested will be the Reservation, Timeoff, Settings, and Users models. Controllers to be tested will be Ajax, App, and Pages, as well as Reservation, Timeoff, Settings, and Users. In addition, we will test all coded views, templates, layouts, and any other files in the source folder. The entire group will assist in writing the tests, though individuals will focus their efforts on testing one file, instead of on particular types of tests. For example, one group member might focus on writing tests for the UsersController, while another member might focus on writing tests for the ReservationsController.

We will prioritize our test cases in the following order: - Valid and expected inputs - Common scenarios - Malformed or malicious input - Edge cases - Issues from early development or early feedback - Recurring but inconsistent bugs - All other test cases Unit tests will be written for methods that trigger different functionality, especially different behavior in the views. In addition, we will write unit tests for all methods in the controllers. The name of the test will indicate what is being tested, but comments will be added in case the test name is not clear. For example, the test named testLoginWithMissingEmail() will test a user trying to login with a missing email.

Finally, we will be performing other non-unit tests. A brief description of each follows below.

Security Tests

We will write unit tests to verify that the system is protected against SQL injection and malformed input in all areas of the project (logging in, signing up, adding a reservation, adding a time-off request, viewing users, adding new users, editing profile, changing settings, and viewing user metrics). We will write unit tests to verify that a user cannot login with incorrect login information or without being properly approved. In addition, the system should enforce some minimal password choice security and be able to handle incorrect URL redirects.

One of the most extensive security tests will involve ensuring the users cannot access functions that are not available to them. The required functions for each user type, quoted from an email from Lucas, are listed below:

Online Reservation System

- Client
 - Make reservations
 - Change email address
 - Cancel previous trip requests (not within 24 hours of scheduled appointment)
 - Request an appointment
- Hourly Staff
 - Review all pending reservations
 - Confirm reservations
 - Deny requested trips
- Admin

- Review all pending reservations
- Confirm reservations
- Deny requested trips
- Review statistics on individual staff members including the number of reservations opened and the number of emails generated with trip information to clients
- Driver
 - N/A

Staff time-off request system

- Client
 - N/A
- Hourly Staff
 - Submit requests for days off
 - View calendar of days on which timeoff requests are granted from that day up until June 30 of the next year (end of JAUNT fiscal year)
- Schedulers
 - Submit requests for days off
 - View calendar of days on which timeoff requests are granted from that day up until June 30 of the next year (end of JAUNT fiscal year)
 - Edit and grant timeoff requests for all hourly staff
 - Print out formatted acceptance or denial form in one-click to distribute to driver or staff mailboxes
 - Admins
 - Submit requests for days off
 - View calendar of days on which timeoff requests are granted from that day up until June 30 of the next year (end of JAUNT fiscal year)
 - Edit and grant timeoff requests for all hourly staff + schedulers
 - Print out formatted acceptance or denial form in one-click to distribute to driver or staff mailboxes

Usability

We will ask for feedback on the following processes: creating an account (both during sign up and when an administrator creates another user), logging in, submitting a reservation, submitting a time-off request, reviewing (approving, editing, and denying) a reservation, reviewing (approving and denying) a time-off request, viewing reservations, and viewing time-off requests. Feedback will be gathered from the customer and the test users the customer has asked to extensively test for bugs and problems.

To evaluate the subjective usability of the system, we will ask the customer and other users specific feedback questions that fall into different categories: - What did you like about the interface in general? - What did you dislike about the interface in general? - Creating an account (sign up) - Did you get any error messages? - Did you understand the whole “have you ridden jaunt before” thing? - Were any error messages clear? - Did the captcha make sense? - Creating an account (for staff

member) - Were all requirements clear? - Logging in - Did you get any feedback messages? - Were any feedback messages clear? - Submitting a reservation - Were all requirements clear? - Submitting a time off request - Were all requirements clear? - Did selecting the options on the calendar make sense? - Reviewing (approve, edit, deny) a reservation - Was the process clear? - Reviewing (approve, deny) a time off request - Was the approval process clear? - Was the information displayed clear and useful? - Viewing reservations - Is the information displayed clear and easy to understand? - Viewing time off requests - Is the information displayed clear and easy to understand?

Installation

The customer will be hosting the system on a Windows machine so we will create installation instructions that will work with this operating system. Since our development was not done in Windows, we will create a blank Amazon EC2 instance that runs windows and use this to verify the installation instructions that have been written.

(<https://github.com/uva-slp/jaunt/blob/master/docs/installationInstructions.md>) We will verify that all listed commands work as intended, and do not produce any error messages. The acceptance criteria relies on what capabilities the built-in users are able to accomplish. Built-in users should be able to login, create new users, submit a time-off request, and submit a reservation (contingent upon the user having the appropriate permissions). In addition, all unit tests must pass.

Requirements

We will consult the requirements document and make sure that everything has been completed. Next we will consult with the customer to check if everything has met their needs. For this, we will check if all the different user types have proper functionality within the system as well as check and see if the required functions are working properly.

Compatibility

We will execute the requirements test on the most recent version of the major web browsers (Firefox, Chrome, IE, and Safari). If -- and only if -- the requirements test passes on all of those browsers, then the compability test will be considered to have passed.