**Task Roulette**

**Test Plan**

**COP 4331, Spring, 2014**

**Team Name:** MADNESS (aka. Team 14)

**Team Members:**

* Cody McMahon
* Jessica Carter
* Matt McGivney
* Steven Lo
* Gunnar Skotnicki

**Modification history:**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| v0.0 | 05/13/13 | S. Applegate | Template |
| v1.0 | 03/09/14 | C. McMahon | Formatting |
| v1.1 | 03/26/14 | G. Skotnicki | Added some Test Cases |
| v 1.2 | 03/27/14 | J. Carter | Introduction, Description of Individual Test Cases |
| v1.3 | 03/27/14 | M. Mcgivney | Stopping Criteria |
| v1.4 | 03/27/14 | S. Lo | Description of Testing Environment, Individual Test Cases |
| v1.5 | 03/28/14 | M. McGivney | Description of Testing Environment |
| v1.6 | 03/28/14 | J. Carter | Decription of Individual Test Cases |



**Contents of this Document**

1. **Introduction  
       Overall Objective for Software Test Activity  
       Reference Documents**
2. **Description of Test Environment**
3. **Overall Stopping Criteria**
4. **Description of Individual Test Cases**



**Introduction**

**Overall Objective for Software Test Activity**

* We expect the test efforts to show us any problems we are having such as support across multiple platforms, modifying the database, etc. This will help us iterate through the development process effectively, limiting time needed for maintenance after release.

**Reference Documents**

* Concept of Operations
* Project Plan
* SRS
* High Level



**Description of Test Environment**

We will be manually testing our Web application on both mobile devices and personal computers. On mobile devices we will be testing on both Android 4.4 and iOS 7.0.1 using their default browsers (Internet and Safari, respectively) as well as Google Chrome 30+. Personal computers will be using the following browsers: Mozilla Firefox 26+, Google Chrome 30+, Safari 6+, and Internet Explorer 9+. This will allow testing for access on all major modern browsers.

We’ll be testing between the DigitalOcean server and the web pages through the web browsers. For data storage (i.e., login information, assigned tasks for each account) we will login to the server directly and check the database against what the user has submitted and is what is visible on the page. Everyone will participate in testing in some way. Steven will be the primary QA tester and leading the effort.



**Stopping Criteria**

During each Sprint, testing will be integrated into the development process. Development/ Testing will not stop unless the bug is deemed critical. Each bug will be reported using our Google Drive Form and email system. Testing should continue until there is a fatal error or until the test concludes as planned. We are going to start a unit testing segment so the testing is not based on a set time, but instead based on each component’s completion time.

After each individual component testing phase is complete, we will move to full system tests. The system will be good enough to deliver when there are no fatal bugs. Specifically, cosmetic bugs are permissible for the purpose of delivering the project. However, they should still be fixed if time permits. Efficiency tests will be continuously run and if improvements can be made, we will work them into our sprints. Since we are using an agile project management model, there is always room for improvement to be worked into the development schedule.

**Description of Individual Test Cases**

*Template:*

|  |
| --- |
| ***Test Objective:*** |
| ***Test Description:*** |
| ***Test Conditions:*** |
| ***Expected Results:*** |

|  |
| --- |
| **Test Objective:** Data Consistency |
| **Test Description:** Pull up information for a user’s tasks from the database and check the consistency across various platforms. The platforms include:  Mobile: Internet (Android 4.4 )  Safari (iOS 7.0.1)  Google Chrome 30+ (Android and iOS  Personal Computers: Mozilla Firefox 26+  Google Chrome 30+  Safari 6+  Internet Explorer 9+.  The users will consist of all group members testing their own tasks. |
| **Test Conditions:** We will check the data across Windows, Mac, and Linux Desktop Operating Systems as well as Android and iOS mobile Operating Systems. |
| **Expected Results:** The data on the page will be the same across all platforms. |

|  |
| --- |
| **Test Objective:** Redirecting when nonexistent pages are accessed |
| **Test Description:** Attempt to go to pages that do not exist. The page names will vary since the final definite structure has not yet been determined. For example: [www.taskroulettle.biz/thisfiledoesnotexist](http://www.taskroulettle.biz/doesthiswork). |
| **Test Conditions:** We will attempt to access as both a signed in user and a non-signed in user. |
| **Expected Results:** We will be directed to a 404 page (a pretty login). |



|  |
| --- |
| **Test Objective:** Allowing Signed-in Users access to certain pages. |
| **Test Description:** Only allow users who are signed have access to the following pages:  [www.taskroulette.biz/login](http://www.taskroulette.biz/login)  [www.taskroulette.biz/signup](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/home](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/list](http://www.taskroulette.biz/tasks) |
| **Test Conditions:** When the user makes the request to one of these pages, if they have a cookie for being logged in, they will be able to have access to those pages. |
| **Expected Results:** The user receives a cookie and is able to access:  [www.taskroulette.biz/login](http://www.taskroulette.biz/login)  [www.taskroulette.biz/signup](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/home](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/list](http://www.taskroulette.biz/tasks) |



|  |
| --- |
| ***Test Objective:***Redirecting when not logged in |
| ***Test Description:*** Attempt to access valid pages while not logged in. The user will attempt to access  valid pages which include:  [www.taskroulette.biz/login](http://www.taskroulette.biz/login)  [www.taskroulette.biz/signup](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/home](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/list](http://www.taskroulette.biz/tasks)  A user who is not logged in will be directed to the login page: [www.taskroulette.biz/login](http://www.taskroulette.biz/login)  If the user does not have an account they will then go to: [www.taskroulette.biz/signup](http://www.taskroulette.biz/tasks) |
| ***Test Conditions:*** A user who is not logged in will try to access a valid page that is not either the home page or the login page.  When the user makes a request to each page, if they’re not logged there will be no cookie received so they will not be able to access the pages. |
| ***Expected Results:*** The user will be redirected to a login page. |

|  |
| --- |
| **Test Objective:** Logging out |
| **Test Description:** A user who is logged in will attempt to log out of their account. When the user presses the logout button on any valid pages they will be redirected to: [www.taskroulette.biz/login](http://www.taskroulette.biz/login)  Valid pages while logged in include:  [www.taskroulette.biz/signup](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/home](http://www.taskroulette.biz/tasks)  [www.taskroulette.biz/list](http://www.taskroulette.biz/tasks) |
| **Test Conditions:** A user will press the logout button while either on: |
| **Expected Results:** The user is logged out and redirected to the login page: [www.taskroulette.biz/login](http://www.taskroulette.biz/login) |

|  |
| --- |
| **Test Objective:** Manipulating a Task |
| **Test Description:** Once signed in, user will be able to add, modify, and delete a task on the page: [www.taskroulette.biz/list](http://www.taskroulette.biz/tasks)  There will be a “Get Task” button, a “View Tasks” button, and an “Add Task” button. When “Add Task” is clicked the user will input the Task, Category, Description, Time, and whether it is mandatory or not. “View Tasks” allows the user to view any tasks they have already added. “Get Task” will randomly generate a task for the user to complete. After a user completes a task, the task will be added back to the list of tasks automatically. |
| **Test Conditions:** A user will sign in and fill out the form to add a task. The user will then be able to either modify and/or delete the task. |
| **Expected Results:** The user will see the added task on the task page, and we will see the added task in the database. When the task is modified, the user will be able to see their changes on the task page and we will see it modified in the database. When the task is deleted, the user will no longer be able to see that task on the task page, and it will no longer be shown in the database. |