**Classification of Components**

Mobile Device

Login

Ingredients

Recipes

My Drinks

Authenticate login info

Get all of the recipes

Get all of the drinks you can make

Get the users current ingredients

Page send credentials to the server

**I/O For Each Component:**

Login Component

User enters his/her login information

If authenticated then server responds with user info. If not then reject message is sent back

Database will populate recipe list on app on startup

User can input his/her own recipe

The database will populate what kind of drinks the user can make from his/her ingredients

My Drinks

If a user enters an ingredient then the database will be updated

Database will populate user’s ingredients on app startup

User can input his/her ingredients

Recipes

Ingredients

Database will be updated when a user recipe is added

|  |  |
| --- | --- |
| Module | Login Component |

Incremental Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No. | Description | Severity | How to Correct |
| 1 | Application should not allow users to create accounts with invalid emails | 2 | Add a regex statement to authenticate email correctly |
| 2 | Application should not allow passwords that are too weak in strength | 3 | Add a regex statement to authenticate password correctly |
| 3 | Application should not attempt to authenticate a username/password when field is left bland | 2 | Verify input before sending an authentication request to the server |
| 4 | Application should not allow users to create accounts with usernames already in use | 2 | Add a unique constraint to the server and a query to check for redundancy on the database |
| 5 | Application should not allow users to create accounts with emails already in use | 2 | Add a unique constraint to the server and a query to check for redundancy on the database |

Regression Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No | Description | Severity | How to Correct |
| 1. |  | 2 | Add an alert to show the |
|  |  |  |  |

|  |  |
| --- | --- |
| Module | Recipes Component |

Incremental Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No. | Description | Severity | How To Correct |
| 1. | Application should not allow drinks to be made without alcohol inputed | 2 | Add verification that the alcohol field has input before sending to server |
| 2. | Application should not allow drinks to be made without mixers inputed | 2 | Add verification that the alcohol field has input before sending to server |

Regression Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No. | Description | Severity | How to Correct |
| 1. | By enforcing a certain number of mixers and alcohol the database component can slow down with to many added components | 2 | Add a limit to the amount of alcohol and mixers you can add to a recipe |

|  |  |
| --- | --- |
| Module | Ingredients Components |

Incremental Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No. | Description | Severity | How to Correct |
| 1. | When inputing information when adding an ingredient, the keyboard will cover what you are typing | 3 | Add a keyboard avoiding view to the code to correct the situation |
| 2. | Application should not accept a proof above 200 | 2 | Add verification that the proof is under 200 |

Regression Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No. | Description | Severity | How to Correct |
| 1. | When the user would add a ton of ingredients to his/her inventory the my drinks component would take forever to load due to queries not finishing | 2 | Limit the number of ingredients a user can have in his/her inventory |

|  |  |
| --- | --- |
| Module | My Drinks |

Incremental Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No. | Description | Severity | How to Correct |
| 1. | When many drinks are available it can take a while to find the one you may be looking for | 3 | Added a search feature to the page for the available drinks and saved drinks |

Regression Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No | Description | Severity | How to Correct |
| 1. | By adding a search function and showing the results in a new page the flow of the pages in the app was thrown off and there was no way to return to the other pages | 2 | Instead return search results in the same view as the drinks are originally displayed |

|  |  |
| --- | --- |
| Module | Tomcat Web Server |

Incremental Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No | Description | Severity | How to Correct |
| 1. | Unauthorized handler was not getting called when appropriate | 2 | Fixed server filter order |
| 2. | Mapping a url to urls that do not need authentication was not working properly | 2 | Implement a pattern matcher for urls |
| 3. | Logining in with unauthorized creditals would send back stack trace instead of correct error message | 2 | Implemented custom exception handler |

Regression Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No. | Description | Severity | How to Correct |
| 1. | When implementing the custom exception handler the login page did not recognize it so it would crash | 1 | Added a function to recognize the exception being thrown by the web server |

|  |  |
| --- | --- |
| Module | Database Component |

Incremental Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No | Description | Severity | How to Correct |
| 1. | Character encoding for Japanese characters would not get stored right | 3 | Set up a handler to detect Japanese characters |
| 2. | Cascades were not set right and would leave orphan entities | 2 | Corrected script to ensure correct deleting |

Regression Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Defect No | Description | Severity | How to Correct |
| 1. | When adding the corrected cascade script the web server would crash because it would still call the old script | 1 | Updated web server to call correct cascade script upon deletion call |