## Test Plan Identifier:

* Test Plan 1

## Introduction:

* I will do my own testing on the different aspects of the project. A user must be able to find other users and invite them to campaigns. Once in a campaign, the owner of the game must be able to give and take items from each player’s inventory, and each player must be able to rearrange and remove items from their own inventory. After testing myself, I will have users use the project for a period of time. I will use the feedback from the users to improve my project.
* I will be spending eight hours testing the project myself, then I will begin user testing while I continue my own testing. I plan to do user testing for four to six weeks on Windows machines.

## References:

* Project Plan: [final-proposal.docx](file:///C:\Users\taylo\files\college\fall2020\csci498\final-proposal.docx)
* Meteor main page: <https://www.meteor.com/>
* Reference source code repo

## Test Items:

* JavaScript v10.19.0
* Meteor 1.11
* Virtual Box 6.1
* Ubuntu 20.04
* Test Users
* <questionnaire.docx>

## Features to be Tested:

* Inventory Database: The inventory database will be organized in a way such that each item will have a name, location, and weight.
* User Sign-In: When the user signs in, their list of campaigns should be displayed if the login was valid. If it was invalid, a prompt will appear showing which field(s) is/are incorrect. There will also be a create account option. If the username entered while creating an account has already been taken, a prompt will appear letting the user know the username has been taken.
* Friend System: The user will be able to search for other users by username. They can friend other users. If the searched user is not already friends with the user, the option to friend the searched user. If they are already friends, the option displayed will be to unfriend them.
* Campaign System: Users who own a campaign can invite other users to the campaigns they own by searching for the desired user by username and clicking the “add player to campaign” option, then selecting which campaign to add them to. From the campaign screen, the owner of the campaign can kick users from their campaigns.

## Features Not to Be Tested:

* All features will be tested.

## Approach:

* I will use sample user data to ensure the system can perform each task it should have the ability to perform. Additionally, I could have my system used in campaigns I participate in. Then, I can get user feedback and make improvements.
* This test plan is a master test plan.
* Doing two approaches, manual testing, usability testing. I will perform manual testing myself. I will systematically attempt to break my project. When I find issues, I will fix them and implement regression testing. For usability test, I will ask some friends to use the product after the manual testing and give me feedback on what they do and do not like about the user interface and general usability. I will make changes based on their feedback and perform a second round of manual testing.

## Item Pass/Fail Criteria:

* For the inventory system, if the database can store and access the correct information, then it passes the tests. A person’s inventory should always display all of their items and only their items.
* For user sign in, valid users should be able to access their accounts. When a user signs in, the sign in process should be completed within 30 seconds.
* For account creation, only unused usernames are valid. Creating a new account should take no more than 30 seconds.
* When searching for friends, all results should be able to have the query as a substring and results should be fetched within 30 seconds.
* For the campaign system, campaign creation should take no longer than 30 seconds, and adding another user to a campaign should also take no longer than 30 seconds.

## Suspension Criteria and Resumption Requirements:

* To suspend testing, I must find a bug harmful enough to warrant an immediate fix. Anything that removes users from campaigns or friends lists when they should not be, anything that deletes accounts, and anything that removes items from the inventory without the user’s command are examples of such bugs.
* Once the bug is fixed, I must go back and test anything that portion of the project interacts with.

## Test Deliverables:

* Test Plan: This Document
* Test Cases
* Test Scripts
* Defect/Enhancement Logs
* Test Reports

## Test Environment:

* I will be testing in Linux using custom user data. The software I am using to create the software is Meteor, and the network I will be using is CSU Wireless.

## Estimate:

* Testing should not take any money.
* Testing should take around 12 hours of custom testing and a few weeks to a few months of user testing.

## Schedule:

* Mar 9 – Beta complete
* Mar 10 – Testing Begins
* Mar 13 – User Testing Begins
* Mar 20 – User Testing Ends
* Mar 30 – Changes made based on user testing feedback

## Staffing and Training Needs:

* None

## Responsibilities:

* I am responsible to ensure user data is not leaked and is not used in malicious ways.
* Test users are responsible for not being malicious with any exploits they may find.

## Risks:

* User data could be leaked.
* Use encapsulation and other hack prevention methods to mitigate the possibility of a vulnerability.
* Text fields could be exploited via SQL Injection attacks.

## Assumptions and Dependencies:

* This project is dependent on Meteor.

## Approvals:

* This test plan must be approved of by Dr. Hayes
* Provide space for signatures and dates. (If the document is to be printed.)