Test Plan

# Purpose

*Obviously, the purpose is to test your software… But what does “testing” mean in your case? Ensuring data integrity is maintained across synchronized databases? Ensure that UI features function properly and with minimal delays in interactivity? Ensure the installer works? (What does “work” mean?) Are you testing all aspects of your software, only certain parts?*

We need to make sure data is updates happen correctly

UI needs to work smoothly without crashing

We need to make sure that clicking a button twenty times in row will not make the program crash. (Friend of mine submitted his first assignment at his job, and his boss crashed his program by doing just that)

All of the features must work…

# Assumptions

*Is there anything your software requires? Something it assumes is working, and that you aren’t responsible for? (Example: Our product relies on the Google Maps API for location data, and as such we don’t verify the accuracy of the information retrieved… we assume it’s correct).*

Are we making any assumptions?

# Testing procedure

*What steps are you going to take in order to verify <thing that you are testing>?*

* *Manual testing?*
* *Automated scripts or other processes?*
* *Unit tests?*
* *Something else?*

*Regardless of what you are going to do/use, how does that thing work? Explain it here.*

We can automate testing by writing programs that might emulate what a user would do. For instance clicking a button will call function foo() and pass in data x,y,z … So we could write some code that calls function foo() with data x,y,z in loop that runs like a thousand times.

Some testing will of course have to manual. Click the button, did it do? If yes, yay, if no, boo.

# Tools / Required Resources

*What is necessary to create or execute any of the tests? Are you using any outside resources? Anything to download or install? Do testers (which are often developers too) need to create an account on* [www.AwesomeTestingToolWeUse.com](http://www.AwesomeTestingToolWeUse.com)*?*

I do not know of any ‘tools’ that can be used to test our app other than testing code that we write. Anyone got any ideas?

# Verification

*How do you know the results of the tests? What sort of mechanism do you have to track this? Spreadsheets? Emailing of results? Are the tests quick, with simple screen output of Success/Failure as sufficient indicators?*

*Would someone other than the person who wrote the testing system be able to make any sense of it? Would a new team member be able to interpret results, or use the same tools with a minimal amount of fuss?*

Test results can be output into CSV files pretty simple with Python, won’t even need external libraries to do that. For instance a test writes 1,000 entries to a CSV, then writes the 1,000 entries to the database. Then retrieves the 1,000 entries and compares them to the CSV. If they match, success 1. Then it removes the 1,000 entries, then attempts to retrieve from the database, if they are all gone, success 2.

That same test can be run simultaneously from different locations to test for data collisions and inconsistencies across multiple updates from multiple sources. (MySQL is supposed to be able to handle that fairly well)