iSanta Test Plan

Andrew Siegle and Sam Kim

# Manual Testing

Since the user interface of the iPhone is hard to simulate for automated testing, this document will outline the manual tests for the user interaction with the app to create an intuitive and responsive user interface.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Preconditions | Steps | Outcome |
| Basic user input | User begins a new target analysis | User taps on an input field. User is prompted for what to enter into the field. User clicks ‘done’ button on the keyboard | User is returned to scroll view and the data is saved in the model. |
| Take a picture | User begins a new target analysis. | User taps the ‘photo’ field. User is asked where to obtain the photo. User selects ‘take new photo.’ User is taken to the iPhone camera. User takes a picture of the target. | Photo is saved and a thumbnail is displayed in the photo line item. |
| Add an impact point | User is on the target analysis page and the photo is showing | User taps the screen once and a new green impact point is shown on the screen where the user taps. | The impact point is displayed and saved in the model. |
| Edit an impact point | User is on the target analysis screen and has inserted at least one impact point. | The user taps and holds to bring up the menu. The user selects edit mode. The user taps a point (which turns blue and is zoomed in on). | Edit arrow for fine tuning display on the screen. |
| Fine tune a point | Edit arrow are present on the screen. | User presses a direction. | The point moves a few pixels in the specified direction. |
| Export report to e-mail | User has finished entering data and clicks the Export button | User is taken to a report page where information is displayed for final review. User clicks the ‘action’ button. User is prompted for an e-mail address. | An e-mail containing all of the entered info as well as the target image is e-mailed to the specified e-mail address. |

# Automated Testing

The architecture of the system is Model-View-Controller, so the approach we took to testing it was to be able to easily test the models of the system. The system contains “engines” which are models connected to other models. Those too, can be easily tested. The approach for testing is to introduce common interfaces which can be mocked for easy testability.