# 

# 

# 

Test and Validation Plans for Beta Prototype

#### Aaron Kwan, Andrew Lewis, Jonah Simon, Nick Strazis, Patrick Bowling

# 

# 

# 

# 

# 

# 

# Description of the Beta Prototype

### Server

The old server we used in the alpha was very primitive and caused issues with repeated use. Our Beta server is entirely new. It is written python on the Django Web Framework and the Django REST API extension. Additionally, the new server stores the configurations in a MySQL database. The new server is hosted on pythonanywhere.com on the free plan. It currently takes an integer from the client and stores in the MySQL database. Also, the server has an endpoint for grasshopper to grab the current integer/configuration for the wall.

### Android App

The Android App has been improved from last semester. It now supports the new server and a feelings based approach. The user can select happy, sad, angry, scared, or a combination. The app then asks the user to confirm and sends the new configuration to the new server. The new version of the Android App actually reduces the overall functionality of the user to interact with the wall since we got rid of the location, width, and height controls. However, what the user loses in functionality, they gain in convenience as the new version is much faster and has a much lower complexity in actual use cases. Overall we believe that this version of the app will be much faster and easier to use while still keeping in mind the needs of the user.

### Kinect

The Kinect has been designed so that it will track the user’s motions and convert them into coordinates that move the wall. The kinect takes a user’s body and creates a digital skeletal structure of their body. Once the skeleton structure is made, it takes anchor points at joints and iconic features and then maps gestures based on how the anchor move. The kinect then sends the user’s gestures to the server as an XML file filled with coordinates. The next step for the Kinect is to be able to communicate emotions from the user to the wall, as an option instead of the existing coordinate system. This is in the early stages of planning, so we will know more in terms of how this capability will manifest in the Kinect project after planned meetings with the rest of the group.

### Alexa

Our prototype for receiving voice commands uses Amazon’s Alexa technology, one of the best pieces of software for voice recognition in the industry. This was split into two parts: creating an Alexa skill to recognize certain keywords and spoken messages, and writing an AWS Lambda function to process the input and produce an appropriate output. A user can say a command or an emotion such as “I am happy” and the Alexa skill will receive the phrase, parse out “happy” and send that to the Lambda function. The function will then send an integer corresponding to that emotion to the server which then passes it to Grasshopper.

### Affectiva (emotion recognition)

The Affectiva application is currently hosted by our server and runs in the browser, using the user’s webcam as the input device. Our application uses Affectiva’s free SDK to analyze the facial patterns of the user. The Affectiva software uses many small facial markers such as “tightness of lips”, and “width of smile”, to decide the emotion that is displayed on the user’s face. After the Affectiva has determined the emotion of the user, our application sends the data to our server as a preset integer.

# Scenarios for the Tests

With our testing, we want to simulate how this will be used in the real world. To do this we crafted the following scenarios. The main user scenario would be to change the wall when their feelings change. For that we have done user tests where users change the wall by giving an emotion to Alexa, Affectiva, or the Android app through their respective input methods. This will also be the test scenario we utilize for future devices we add. The next scenario covers the window wall and the Kinect. This wall does not support feelings and instead uses the coordinate system with Kinect. For this scenario the user can test by interacting with Kinect and seeing the window change.

# Data Collection Plans

Since our project’s goal is to design a way for users to conveniently pass information to manipulate a wall, we decided to have the data collection focus on user feedback of our models. We believe that tracking how users felt about our products would be the most efficient way to make improvements to our system. The plan is to bring in users to test the models and then ask how their experience went. Based on what they want improved for a better experience, we will set up a priority list on what to improve.

# Analysis, Modeling, and Implementation/Simulation Plans

For analyzing, the information will be done by creating a priority chart based on user suggestions on what to improve. As for modeling and implementation/simulation plans, the team has decided that these will not be necessary. The team believes that going straight into live testing will be the best way to improve the system since modeling and simulations will not give us the proper feedback we will need to effectively improve the system.