# **Testing and Evaluation**

## **Testing**

Unity provides a collection of testing tools that allow us to do unit, integration and assertion testing. However, we mostly used integration testing for our PoC as it is commonly done in Unity because unit testing is too low level and there are plugins/assets that do unit testing automatically for us. Integration testing allows us to test multiple objects interaction in the same scene and requires little to no coding and everything is done in a separated test scene independently, meaning in each test the scene will be cleaned up before they take place. This testing is done for user interaction with the AR element which in our case, user opening and closing the chest

Vuforia is a very powerful AR development platform meaning it comes with the cutting edge and reliable image recognition algorithm. We tested different types of images (different colour and pattern) as the “Image Target” and the app never fails to recognise the image regardless of the surface. This was the most authentic form of testing because it gave us a true idea of how the app works on a real device.

### **Boundary testing (This part might be only applicable for the API?)**

#### Input tests

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Test | Expected Outcome | Actual Outcome |
| Add Entry | | | |
| 1 | Press Sign In when nothing has been entered | Alert will pop up to tell the user to add some detail. The post will not be saved | The post was saved. This needs to be fixed |
| 2 | Press submit when nothing has been entered (after fix) | Alert will pop up to tell the user to add some detail. The post will not be saved | Correct |
| 3 | Press submit when the time selected is in the future | Alert will pop up telling the user that you can’t select a time in the future and the post will not be saved | Correct |
| 4 | Press submit when a meal that has already been entered that day is selected | Alert will pop up telling the user this can’t be done and the post will not be saved | Correct |
| 5 | Press submit when only thoughts have been added | The post will be saved in Logs and appear as a Thought post | Correct |
| 6 | Press submit when only a purge has been selected | The post will be saved in Logs and appear as a Purge post | Correct |
| 7 | Press submit where meal has been selected, but no food and drink has been entered (or vice versa) | Alert will pop up telling the user to add whichever of the two has not been added | Correct |
| 8 | Press submit when location and/or people have been selected, but food and meal haven’t | Alert will pop up telling the user to add more details | Correct |
| 9 | Press submit when there are line breaks in a text area (e.g., food and drink or thoughts) | Line breaks will be ignored in Logs | Correct |
| Thoughts | | | |
| 10 | Press submit when nothing has been entered | Alert will pop up telling the user to add more detail | Correct |
| Purge | | | |
| 11 | Press submit when nothing has been entered | The post will be saved in Logs and appear as a Purge entry | Correct |

#### Logs tests

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Test | Expected Outcome | Actual Outcome |
| 1 | Swipe on an entry and press the delete button | The entry will be permanently deleted | Correct |
| 2 | Delete an entry *x* that comes chronologically before another entry *y* when the user added *y* before *x* | The entry will be permanently deleted | Correct |
| 3 | Delete an entry by pressing the button on the detailed logs view | The entry will be permanently deleted | Correct |

#### API tests

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Test | Expected Outcome | Actual Outcome |
| 1 | Call create-reward API with optional params | Reward is successfully added with default parameters | Correct |
| 2 | Call sign-in with wrong credentials | Authorisation Failed | Correct |
| 3 | Call closest-advertisement with coordinates | Geofenced advertisement lists | Correct |
| 4 | GET specific reward linked to the AD | Reward with details | Correct |
| 5 | GET random reward specific to the AD when there was no more reward available | Failed to obtain more reward | Internal Server Error(500) |

Additionally, we used a few example applicationsReference for testing and to see how some features work in practice. These examples also helped us find a way to refactor some of our code.

## **Design Pattern**

In the API sector, we employed MVC (Model-View-Controller) design pattern. Django provided database ORM, which are represented as models,templates of responses as views, and finally controller which is represented as ‘views’ in Django.

## **Achievements and Evaluation**

Throughout the whole app development project, we’ve had 4 meetings with Luis Lancos and Mike Smith - our Client. 3 of those meetings were Skype meeting with app demos where we showed the current stage of our app along with explaining the designs and techniques we used and are planning to use and report the team’s contribution.

### **Meeting #1:**

This was our first and only meeting that took place in person, it was at the start of the year, we were introduced to Luis who would be taking over Mike Smith (our original client) and discussed the project goals.

### **Meeting #2:**

We had a Skype meeting with Luis and his team, we showed our first prototype which has a strong focus on the AR game because we were just interested in the capabilities of Vuforia as an AR development platform. At this stage we were still figuring out how to integrate the API into our code in either the Xcode build and the Unity build. Luis and his team were quite impressed with the capability of Vuforia, one of his colleague - Lee - seems to be very happy that we use Unity to work on our PoC. This prototype allowed user to pick any point in front of the camera to be the area which the AR elements will appear on which was not what he had in mind so we were instructed to make it so that all AR elements will only appear on a pre-programmed image target (the advertisement)

### **Meeting #3:**

In this meeting, we demonstrated Prototype 2 which has the ability to interact with the API to allow user login and voucher database. Based on the previous meeting, we decided to include only a simple interaction where we open a chest on the advertisement. Prototype 2 also has a simple UI instead of just going straight to camera view like Prototype 1. Our client was satisfied with our progress thus far and was setting up an “in office” meeting for us to give a live demo but unfortunately Luis had some issues on his side so we never met directly again.

### **Meeting #4:**

In this meeting, we showed our client our final build. Zac was unable to implement the visual map like Pokemon Go as our method (which collects coordinates data from Google Maps API and render it into a 3D block map) was using up too much resources, slow and will eventually crash the app and we couldn’t figure out a better way to implement this, perhaps more research needed to be done. However, we managed to successfully implement all necessary functionality and made the final design to our app, Luis was satisfied with all aspect of the app.

## **Future works**

Currently we only very simple geolocation as well as AR features in the app, however if we’d had more time, we would have liked to implement a visual map and extra AR game for the app, such extra features will certainly attract a wider audience and create more drive to play use this app and therefore exposure to advertisements. Additionally, we would integrate a system of analytics in order to get an even clearer picture of the way the user base interacts with the app. Because of the implementation of analytics, it allows us to monitor the usage of the app and can give us insights about what features could perhaps be better implemented and lead to future changes.

The app has potential for even more growth with the addition of cloud recognition. This would open up a wealth of possibilities, for example, it would then be possible to create a different game for each advertisement panel, or it will allow the advertiser to upload their own version of the AR interaction (something similar to Blippar).