

UCL COMP0016 Team 12 - Lilly Neubauer, Dillon Lim, Oliver Vickers
Bi-Weekly Report: Period ending in 23rd March 2020

Overview: What we've done

Our final two weeks of development were disrupted by the Covid-19 pandemic which forced UCL to close and move to online teaching. Dillon went back to Singapore and was quarantined which made it harder for him to focus on work. We still had a lot of work to finish and tried our best to focus despite disruption. All in all it was quite a stressful few weeks.

Oliver focused on cleaning up the webhook. We decided not to implement the "make appointment" query because it was tricky to work out how to search a database for the "next available appointment", unless we made a database populated with every single doctor appointment slot showing whether it was full or not, which would have taken up a lot of space and been expensive. We therefore decided to focus the GP receptionist on checking users in for their appointments and allowing them to check whether their prescription was available for collection. We finished implementing these functionalities and testing them via the Watson Assistant interface.

Oliver made sure that each JSON response also had a "default" case in it to return a human readable string with an error message i.e. "you could not be checked in for your appointment". This made it easier for the Watson Assistant to return errors - we have not yet found a way for Watson to vary its response based on the result of a webhook, so, for example, we can't vary the response to return an error message if the returned webhook result is "false". This means that human readable error messages need to be returned by the webhook itself.

Lilly solved the app crashing issue and made progress solving some bugs. There was a problem that user prompts were not appearing. Even though when the user was too close to the avatar, and "the user is too close" would appear in the Unity console log, the SetActive statement for the UI text just didn't work and the text would not appear on the user's screen. We worked round this by accessing the text in a different way using GameObject.Find rather than having the text attached to the script using a public variable and the unity interface.

We wrote dialog and created tables for our Library receptionist, which can now answer queries like "I want to read something by Agatha Christie", "I am looking for the book War and Peace" or "I want to read something scary". It does this by looking through the new Books table in the database.

We had many issues with the UI. Since Lilly introduced a trigger image into our app, our avatar would only appear on the image when it was brought into the camera frame. This meant that the avatar, and its logic connecting it to the Watson backend, was instantiated as a prefab object in Unity only after the image was seen (making it a prefab and instantiating it in this way is necessary for the way that Tracked Images work in AR Foundation 3) It became clear that for some reason, the fact that it was a prefab made it impossible for it to interface with UI elements that were instantiated on the app startup (for example, when Lilly tried to get it to show text in a UI element in the main canvas, it simply wouldn't appear). She therefore had to introduce another UI canvas that was attached to the avatar. The messages that came from the avatar logic (such as showing what Watson Assistant had heard or what its return message was) were displayed on this canvas

instead. This worked, but it led to issues with having multiple canvases - they masked each other, so that when the avatar canvas appeared you could no longer use buttons on the other canvas. Lilly solved this by introducing a multi-player event system with multiple listeners (a recent new Unity feature) that allows multiple canvases to be active at the same time.

We user tested our app with as many users as possible, given the circumstances of social distancing and isolation. Lilly tested the app with five different users (friends and members of her family) in both quiet and noisy environments. This led to some improvements in the app - there were some questions that users asked that we had not considered, for example a user of the library asked to "borrow" a book rather than to "find" or "read" one, so we add this keyword into the "find" intent of the library assistant. Additionally, It became clear that there was a bug where the app would start saying it's "welcome message" to the user and then it would hear this as an input and start answering itself. This could lead to it just saying "hello" to itself over and over, although this was avoided by lowering the volume of the assistant output. Given a bit more time we could probably introduce a method so that the assistant is not listening whilst audio is playing.

We worked on the documentation for the project; our website; filming videos; tidying up our code and on our FHIR projects. Dillon created multiple pages for the website, so that the text was broken up rather than being displayed on one long, scrollable page, and added new text that Lilly had written.

Tasks completed:

- Majority of deployment manual / documentation completed
- Lilly wrote most of the text for the website, just a few things need finishing off
- Dillon created new format (multi-page) for website
- Oliver solved issues with webhook: Returning empty parameters on errors; Returning an error for certain members of the staff table
- Lilly wrote and implemented conversation structure for library assistant
- Lilly wrote and implemented conversation structure for GP assistant
- Finished and submitted our FHIRworks projects
- Lilly finished and submitted IEP Creative Writing Blog Post
- Lilly finished and submitted Poster
- We wrote our legal issues essays and agreed with Jon (IBM) to release our project with an Apache 2.0 licence
- Lilly conducted user testing
- Lilly filmed our app for our video
- Oliver solved issues with our azure membership (we had run out of credit)
- Lilly created maps of use cases for each receptionist to show what can be said by a user and how our application handles each case
- Lilly fixed the user prompts that were not appearing
- Lilly added another canvas to fix the fact that the avatar text input and output panel wasn't working
- Lilly and Dillon finished the use case lists
- Oliver finished entity relationship diagram and wrote text explaining his section of the design

Are we on track?

Our deadline is this Friday, so we don't have a lot of room left to not be on track! Overall we have done reasonably well. Our app does work, and we have implemented all our must-have and most of our should-have functionality. We definitely have some reliability issues and with more time to work on this project we could do a lot more, but this is to be expected with such an ambitious and evolving project.

Known Bugs

We still have some bugs in our programme which we are aware of, and we will probably not be able to fix by the deadline.

- The "I'd like to speak to a human button" is not working. A user can speak to a human by asking the chatbot directly, but in the case that this fails there would be a critical user error with the user unable to summon a human.
- If nothing is returned from the Watson Assistant (i.e. an error occurs in the Watson Assistant, which sometimes happens if it gets confused about user input and there is a recognition clash between two different intents), the Watson logic will freeze and will not recognise any new intents. Lilly tried to fix this by adding safety net statements to check for null returns, but this doesn't seem to have solved the issue
- Sometimes Watson Assistant carries over context variables from one query to the next when it shouldn't do so. For example, if you ask if for a particular book, it will save this in a context variable. If you then follow this by asking for a book it doesn't recognise, it will override this non-recognised entity with the previous context variable, which can lead to invalid responses. Lilly researched why these was happening and tried to find out how to clear context variables in Watson, but couldn't find information on how to do this.
- Image tracking doesn't work if the trigger image goes out of the camera frame. Currently, when the trigger image goes out of the camera frame, the tracking of the avatar becomes buggy and it tends to float around the space. It's unclear whether this is due to the TrackedImageManager component of Unity AR Foundation 3 that underlies this AR feature - this was only released earlier this year so it's possibly that it's tracking capabilities are not perfect. It's possible that Lilly hasn't implemented this properly and there might be some way to resolve this using Anchors, but she hasn't been able to find any online information to confirm this or suggest alternative approaches.
- We have not tested our UI with different sized devices to check for responsive design, only on Android mobile devices. We did not have an android tablet available in the last few weeks of the project and due to Covid-19 were not able to get one from the department. Before this time our UI was not in a sufficient state of completion where we would have been able to test it properly on a tablet. For this reason we don't know if there would be issues building to larger devices.

Plan for the final week

- Finish our website
- Film our technical and client videos
- Finish writing the deployment manual / documentation
- Finish writing our user manual
- Package our code to upload to Moodle and handover to client