

Design Meeting 2 with Microsoft
Week 3– 17th/05/2018

Imperial team presented documents of progress to MS team. A Gantt chart outlining the tasks and requirement along with due dates and task allocations within the Imperial Team.

Gantt chart designed by the team was accepted by MS with the goals being seen as realistic and achievable. Gantt chart discussion summarized

- Realistic targets and task durations
- Agreement with MS team that the C# Kernel should be completed by 5th/06/2018 –

prolog/shells script -> bash script to test on azure notebooks

this way we have a week prior to the hackbooth for testing and we can work with

Dino to get our kernel on Azure notebooks once its stable (putting docker file into a bash script)

- **TESTING** : MS team recommends at least a week for testing, recommends we test in the notebook environment early even before all functionalities are implemented.
 - Imperial team should have testing components ranging from the simple **hello world to complex ML.net sample with images to test the inline capabilities of Jupyter. Ensure we test both our C# Kernel and its interaction with Jupyter aspects such as images, text etc.**
 - Dino provided advice on testing on notebook environment by setting up a custom kernel on azure notebook as seen below:
 - “Here's a library that sets up a custom kernel:
<https://notebooks.azure.com/dino/libraries/CustomKernel> .It has a setup step configured via the settings button on the library that runs the setup.sh. That will make a new kernel in the user's .local/share/jupyter/kernels directory. It'll then copy over a .kernel.json which is in the library as well, and it grabs the icons from the system Python. Ultimately this just ends up being yet another Python kernel, but it runs against the system Python instead of one of the Anaconda Python's which are installed. You should be able to basically take the steps that are in your docker build and put them in there to build the C# kernel and point the kernel.json at that - it'll presumably be a little bit slow of a startup time, but it'll at least let you test things out.”
- MS recommends that delivery can be done by putting the kernel in a docker file format then into a bash script (make it a containerized environment by making a PR back once changes are made)
- MS recommends we work on the fork and drop a note on the repo to make clear of our intentions etc
- Note: PLOTLIY?!?! If we need to output in javascript and have it in the preview talk to Dino for rendering the **custom mime type**
- Demo -> Pick a basic C# sample, do a screen share during 31st/05 call (Ideally with azure notebooks -> once we can setup a custom kernel on Azure)

- A point came up that it would be worth looking into *ml.net* – a machine learning framework for .net users. A lot of C# users want to use ML when they develop and Jupyter is the right canvas for such a scenario so adding a kernel that can use the ML libraries would be GREAT!

Imperial team should make the kernel stable enough prior to delivering to MS team.

- Our kernel should be easy to set up
- MS recommends putting our final kernel (docker file) into a bash script
- Make it a containerized environment by making a PR back once changes are made

Supervisor Meeting 2 – 18th/05/2018

Main point across we need divide into subgroups and assign tasks to each group.

We divided into groups of 2-2-3 and assigned each of the core requirements of the project to each group.

Groups	Tasks
Mohamed and Mostafa	Intellisense
Andre and James	Syntax Highlighting
Mwana, Thanos and Omer	Error display