~~Agile methods~~

~~As a group we began this process at a disadvantage to other groups, as 3 of our members had the understanding of what Agile software development was but no real knowledge of how to implement it. We started the process by collecting ideas of how we should go about the project and entered them into a spreadsheet, as at this stage we had gained some insight into how SCRUM worked. From this early and very rudimentary beginnings of a project we added system until we got to where we were comfortable the process we used could easily replicated in our working lives after university.~~

~~We gathered our early drafts of how we wanted the website to work, what should go in it and how it should look and kept them in a centrally accessible point. This started off at being on Google Drive which we all shared, but had no real way of discovering who had done what, when or why. This gave us a very simple repository for our growing body of work, but no real version control as such because work could easily be lost or written over without the ability to roll back or even spot a collision of work.~~

~~Within a few weeks of the process beginning we began to understand how we could apportion difficulty to the tasks we had discovered from our early meetings. We prioritised the jobs into tasks and used “planning poker” to attribute difficulty of the tasks. We could then start to see some of these tasks were epic, allowing us to split these tasks into more manageable sub tasks. These metrics were then fed into our spreadsheet and we could plan our first SPRINT. To plan this first sprint we had a meeting to discuss what we could manage to complete in a period of 3 weeks (this was the period of all our sprints) to find out our initial velocity. This first sprint was very much a learning curve for us as a newly formed group and we began by discovering the processes and information we would need for later stages of the project.~~

~~Using a spreadsheet to hold our information was difficult for all our group to understand and made it hard to visualise the project and its progress. With some research we came across a tool that helped us to visualise our project, track progress and plan our sprints on. This tool is an online project management platform called Taiga~~

Version control

~~It became apparent that a better solution to our Google drive was required when an incident occurred whereby a page of the code was accidently lost and no backup was present. The Agile module had shown us that GITLab or GitHub where useful tools for version control. We made the decision to create a repository on the University of Kent’s GITLab facility to keep the project code and all project related documents in. We choose GITLab mainly because we found it much easier to work than GitHub, although GitHub may have been a better choice to show future employers of our successful work as GITLab requires a current University ID to be able to gain access to it. GITLab provided a safe repository which could be used to track changes made and by whom so that if such an event occurred again a last working copy of the code could be reinstated.~~

~~As a result of this project all of our team now appreciates the importance of having a content management and version control system, this has been an extremely valuable lesson learned. In order to use GITLab we required the installation of TortoiseGIT on the University library PC’s as this was where we were working from. We already had access to our own GITLab repositories via the University and had set up one for the project files, in order to make use of the repository we needed TortoiseGIT. Both our supervisor and our team members had sent emails to the Drill Hall library IT support team to request this software installation and eventually this was done for us.~~

~~This approach however did have a flaw which required all the members of the group to commit to the same system of work, and require them to respect this system once decided upon. There were genuine accidents that occurred, but if the system was adhered to correctly they could have been recovered. Regular commits with meaningful messages for the other group members to follow and all working from same repository would allow conflicts to be quickly spotted and fixed.~~

Database

~~Done ☺~~

User Requirements

~~In order to ascertain the requirements for this system we set aside time to discover what we wanted to ask and who we needed ask to produce the workings of a script (index 2.1) for the coming interviews. We decided on the wording for an email (index 2.2) that we would send out to the possible members of staff that might want to engage in this project~~

~~We wanted to find out what kind of equipment they had available to lend, what kinds of restrictions, if any, they wished to impose on those items and how they envisaged a system like ours should work. So as an aide memoire we made a simple Questionnaire (index 2.3) for us to follow when we got results from our emails to potential lenders. At this stage we also thought it a good idea to go with a draft of the user requirements (index 2.4) the site would have. This would give the lenders the core values of the site and allow them to add any specifics to the user requirements they needed to give them the confidence to lend out equipment.~~

~~At this stage we treated this data (index 2.5) as if it was the only data we would use to demonstrate Bookit’s use. We didn’t get an enormous response to our emails, but what information we did gather guided us to produce the following set of lender requirements:~~

* ~~The lender can also borrow other lenders assets.~~
* ~~The lender can see that their personal agreement has been registered by the user.~~
* ~~The lender can see where their assets are at any time.~~
* ~~Add assets to the lenders account.~~
* ~~Edit/Remove assets if they change or become unavailable.~~
* ~~Ban users who disrespect the rules.~~
* ~~Produce separate terms and conditions for specific lenders.~~
* ~~Two tick system that allowed them to decide if they would allow people to borrow items rather than a blanket allowing of lending.~~

~~We also decided upon some borrower requirements:~~

* ~~University of Kent student or staff member should be allowed to borrow assets.~~
* ~~Need to be able to cancel a booking they don’t need.~~
* ~~Need to know where to pick up and drop off asset.~~
* ~~Need confirmation from the lender that the item was returned after use.~~

~~Common requirements:~~

* ~~The service needs to available on multiple platforms~~

Introduction

~~Having a service that can be accessed online is becoming a popular and even an expected method of browsing and requesting loans of equipment. Indeed our own University library has an online service for the sourcing and renewing of loans. Having such a facility means that equipment available is advertised to borrowers on a platform accessible to all rather than only a select few knowing of the existence of such items. A catalogue of items can be browsed and searched by those looking for equipment to use for projects or research purposes, the availability of the items can be checked and requests made for loans. It also means that owners of equipment can confirm or deny requests and track where their items are by way of an inventory page.~~

~~It had become apparent from discussions with lecturers from University of Kent (and later at the Poster Fair, other Universities) that it can be troublesome to remember what assets the lecturer actually has at one time and who has them. It can also be useful to have a layer of bureaucracy between a lender and user, so there is a metaphorical paper trail the lender can use to find their assets. Another advantage of this system is the lender can point the potential borrower at the Bookit system rather than have to field calls and let the system decide if they are eligible to borrow the asset.~~

Aims

~~We aimed to supply online application available to University of Kent undergraduate, postgraduate and staff across multiple platforms. The application can be used by staff to advertise assets that they would like to lend to specific audiences within the University of Kent domain. Initially the domain is set for the School of Computing across the Medway and Canterbury campuses, but could be expanded to other faculties in the future.~~

~~The borrowers and lenders both have to agree to the site rules before being allowed to advertise loans or make them. This allows the users of the facility to be sure they understand that the equipment will be respected or access to the facility will be withdrawn. The assets will able to be set up so that only selective groups can borrow them depending on the rules set by the lender, allowing the lender to set their own conditions that the borrower has to accept if they want to borrow the asset. The lenders should be in no doubt of their responsibilities under the agreements and should be allowed to view them and digest them before agreeing to them on the booking screen.~~

~~The application will also supply administration of the site to be able to block users if the break the conditions of the site, and unblock them after a period of suspension has elapsed. It will give administrators the ability to add new lenders, edit and remove (warnings issued for this) them. The administrators will also have the ability to delete and edit users as would be expected for any online application with user accounts.~~

~~Bookings will work on a two tick process this works by the eligible borrower requesting the asset from the lender (first tick) at this point the lender can confirm (second tick) or reject the request. At these stages the Bookit system will keep the borrower and lender up to date by automated emails. On confirmation the lender will be emailed with details of where to pick up and return after the term of the loan. When the loan period is over and the asset is returned the lender will complete the loan by confirming receipt of the asset back which will trigger a thank you for participating email to the borrower from Bookit.~~

~~The online application will look endeavour to look and feel as much like a University of Kent website should incorporating the username and password used for all other University of Kent services allowing users of this service a smooth acceptance in the scheme without having to create a new account.~~

~~We will follow and expand on our knowledge of agile software development techniques to increase the quality of our work and incorporate version control to foster a safe environment for the production of systems of work~~

App

~~As part of the multi-platform aims there was a decision to create an app for mobile phones this needed us to choose a direction of which device we should concentrate on. As a group we are more comfortable and have greater access to Windows based machines which made the choice of Andriod a virtual automatic choice. Andriod offers a free Software Developers Kit installed on the University of Kent machines called Andriod Studio, and the University library had reading material for one of our members to use and digest. We originally started with the idea of having a standalone app to perform the same tasks as the website, but quickly made the decision that time and skills would not be possible to deliver this aim. Instead we made the website as portable as possible and created a mirroring app for Andriod phone users.~~

Conclusions

The website is a familiar look and feel to University of Kent’s output that utilises the common log in system that all students and staff have already. We believed this feature to be useful as it didn’t require the user to remember another set of username and password, and could reuse an already checked set of attributes for the website to use. We did have to ask for some extra information from the user on first logging into the system this was because we could not easily access name, year or campus. This meant we had to add some extra administration to user accounts as name, campus user resides at and year level can change over the period of the accounts are active.

The website succeeded in its main aim and that was to create bookings for assets belonging lenders. That being said we did change the way the bookings appeared in the database from a shopping cart method where multiple assets could be linked to a loan. To a single asset being placed in a loan content that relates to a unique loan this allowed us to easily institute a 2 tick confirmation method of loan control, rather than a scenario where part of a loan could be denied and part of a loan being confirmed.

Our system in complexity doesn’t compare with the drill hall library system which has a far larger amount of attributes connected with each type of asset belong to it, but our system has a greater emphasis on understanding what each of the items looks like and the control connected to the owner of the asset. The drill hall library has a blanket coverall set of rules which not only cover books, but behaviour within its buildings. This differs from our terms and conditions, as we can set terms of use for the website and then the lenders can set if they want extra terms and conditions to use their assets.

This isn’t a new idea, but what made it novel is we tried make our website look University of Kent library system and feel like an online shop this allows the users easily navigate around the assets they might want to borrow without the hassle of signing up for another facility

We would have benefitted from a clearer defined scope of the project allowing us to begin the building stage earlier. We concentrated too much time on gathering information at the beginning of the exercise about who wanted to lend what we should have used our own initiative and made up assets. We needed to know how lenders wanted to lend rather than what they wanted to lend, for example linking types of assets to year groups and attaching agreements to different assets.

In the future we have had idea of how we could expand this to other faculties which has partially been implemented by adding the Group table to the database. If this happened we might need to expand on the types of asset we want to store, because at present it is limited to Pi’s, books, Lego and EEG headsets.

It had also been discussed with the “shed” people from Canterbury how they would like the system to be able to lend out batches of items, for example a block of 40 Pi’s. We believe this could be done with expansion to the AssetType table to include model numbers for specific items. This way we could query all the items with that model number then add all of their AssetUID’s to a block booking.

Pair Programming