Agile methods

~~The group decided to employ SCRUM for the development of the project. As a group 3 of the members had an understanding of what Agile software development was but no real knowledge of how to implement it. The process began by collecting ideas of how we could organise the project and entered them into a spreadsheet, as at this stage we had gained a little insight into how SCRUM worked from our lectures about agile practices.~~

~~The group gathered 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. At this point three of the team members had no knowledge of content management or version control systems, a shared Google drive was used to store all of the work. All of the code for the project was to be kept on the Raptor project drive. 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.~~

~~The process of creating the first sprint saw our team using planning poker for each story until we had estimated the points that we thought each story for the entire project was worth, then we decided on our first sprint goals according to importance and created our first sprint backlog. Our sprints would be 3 weeks each.~~

~~As can be seen in corpus index 1.2 we wrote all of our stories on post it notes which we then wrote the knowledge acquisition and non-functional requirements of each story on the back.~~

~~As our project progressed we adapted our methodology to use a hybrid of SCRUM and XP whereby we always had a sprint planned but if we finished tasks and found that we had time, or that there was a task that required urgent addition to the current sprint we added this job and included it in the workload for the sprint. This worked well for us as there were often times where we realised that a particular task required work on something else to be finished before it could be completed.~~

~~During our project we adopted the use of Taiga to keep track of sprint backlog tasks. The interface is useful when planning sprints and splitting stories into tasks. It made allocation of tasks to individuals easy as all we had to do was log in and pick ourselves something to work on, the team could see what other team members were working on and any tasks that were unassigned. This made the picking up of tasks very easy. The only downside to Taiga that we found was that it was hard to assign tasks if they were being done using pair programming, it meant the creation of two identical tasks and assigning both team members. We found that some tasks were unfinished at the end of some sprints, this might have been where we had underestimated the amount of effort required to complete the task and had assigned it a low number of story points, in these cases we ended up having to drag work across to the next sprint, Taiga allowed us to do this easily.~~

~~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 was a useful tool for version control and the decision was taken to create a repository to keep the project code and all project related documents in. This provided a backup 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.~~

Things we haven’t used:

~~Jenkins was explored as an option for automated building of the project files as it supports GIT. This option was discarded after research showed that Jenkins is more suited to the building of projects written in languages such as Java and would require the downloading of extra proprietary software onto the Universities raptor server, something that students do not have administrative privilege for. As it was not essential that we implemented Jenkins this was not pursued further.~~

~~Php unit testing was considered when performing the multitudes of tests and checks of our system. Further research into using PHPUnit revealed that, as with Jenkins, it required additional software to be installed on the Universities server which students are not permitted to do. In order to test the system, repeated entries into the database were made via the various forms created. Each entry made could not be a duplicate entry so writing unit tests would not have improved the testing process much as each test would have to be written with a unique entry, this was impractical and would not have saved much time.~~

File and Image upload & checks

~~Security is a big concern when allowing others to upload files to any server, checks must be performed to ensure that users are not uploading anything malicious that could compromise the system. As such many steps have been taken to protect against such attacks. When allowing owners of items to add them to the catalogue, a picture of the item is required, the item is added to the database and this picture is saved into a file in our project. The image is uploaded via a html form and the image is checked using php. The code used was adapted from the w3Schools.com website to suit our purpose. The code checks to ensure the file being uploaded is actually an image, it checks to see if there is already an image of the same name in the folder the images are saved in, that the image is not larger than our specified image size and that it is in either JPG, JPEG or PNG format. If the upload passes all of these checks it is then uploaded into the image folder while at the same time an SQL query is adding the item that the image is related to into our database. This SQL query will only run if the upload passes all of the checks. If the file fails any of the checks the user is informed via the site on a landing page that tells them there is a problem with the file, they are then given a link to click to return to the main page of the site. The uploading of agreements works exactly the same way and both of these checks have been written as php functions so they can be called and used anywhere there is a need to run the checks, at present we only have the uploading of new item images and new agreement text files but the check functions could be used again if we had an inventory page that allowed the upload of documents and images relating to the condition of items on their return to the owner- this was a feature of the system that was not implemented due to lack of time.~~

~~SimpleSaml~~

~~To restrict the use of our booking system to University members only we employed the use of the login system provided by the University. This was relatively simple and required the addition of a small file to our project’s public\_html folder. The file is called .htaccess and contains the following code:~~

~~AuthType Mellon~~

~~MellonEnable auth~~

~~Once this file was saved in the public\_html folder it becomes hidden so that is not visible to anyone who may have accessed our folder with malicious intent. The file allows the use of SSO to authenticate users via the Universities own login system, and passes information about the user to our system. We were then able to access information about the user’s login info, email address and account type by examining the REMOTE\_USER variable of the person that had logged in. We discussed making a request to Information Services for extra information about the user – full name, campus etc but after asking Tim Bishop for assistance he suggested that we weigh up how important this was against the amount of work required to get this information, we discussed it further with our supervisor and decided that it might be less work to create a form for the user to fill in on initial login to request this extra information and store it in our own database.~~

~~Design and implementation of performing Checks for eligibility of loan:~~

~~This expands our explanation of an example page discussed in section …. About Ajax. The page which this relates to is named GetItemInfo.php(index…..) and calls various other scripts to perform the check.~~

~~In order to ensure that items are only borrowed by those who fall into the correct categories as stipulated by the restrictions set by the items owner, checks have to be performed when a request for a loan is made. At the same time, the availability of the item for the date requested must be checked. The system currently allows the user to select an item to borrow, input the date they wish to borrow it and the length of time to borrow. At this point the date checks are performed via Javascript in the onchange function of the booking duration selector, the eligibility checks are made when the user clicks the BOOK button, if this check fails the user is told they are not eligible and to check the items restrictions. The date check takes values from the date selector on the item page, this system was originally using a date picker calendar from the JQuery libraries, for some reason that we have never been able to discover this would not work for us, we tried various versions of these calendars but had no luck, eventually we had to devise a system of our own to input dates. The result is less aesthetically pleasing but is functional.~~

~~GetItemInfo.php also calls other pages in order to work, these being, CheckBookingDate.php(index …) and InsertLoan.php(index …) these pages are only called when they are needed by using JQuery.~~

~~We have made sure that the booking duration is unavailable to select until the booking date has been chosen as the date check runs from the onchange function of the booking duration selector and requires both values in order to perform the check. When a date is selected, the page will then use JavaScript to enable another select box where you can select how many days you would like to book it for. In our form we have ensured that a user cannot leave parts of the booking form un-selected by using the required attribute of the form element.~~

~~Once the date and booking duration have been selected it will run a page called CheckBookingDate.php, it uses ajax to send this data from the two select boxes, to run a script to check availability. The process of checking an available date requires that the values of the dates selected are added to the current date and then using the strtotime php function are turned into a date variable that can be used in SQL queries. It does this by getting the day that is chosen and again using date to string to add the booking length to the booking day -1, it uses a -1 to count the day that the user books the item to allow them to return it the same day, as some items can only be booked for one day. The SQL query checks to see if the date and return date chosen overlap with any other bookings for that item, once this check has been completed it will set information into a hidden field that the user cannot see on the page. If the value is set to a specific value it will enable a booking button allowing the user to book that item. If the item has already been booked out for the days requested, it will not enable the button and tell the user that it is not available for the days they have chosen.~~

~~Once the user has a booking date that is available, they can press the book button, this runs code to create a booking after doing some checks, The system checks the user’s information and compares this to the data held in our database about the item. Using SQL statements to return data relating to the item a selection of php variables is created and a loop performed to check that the user has agreed to the usage conditions, is not a banned user and if the restrictions of the item match the user type, if they do then the user is eligible and the availability for the selected date is checked, if not the loan is denied.~~

~~Bookings are created via InsertLoan.php(index…) this page gathers the other required PHP files and runs them in a logical order, the sequence is as follows (all files index….):~~

~~InsertSQLLoan.php~~

~~This is the page that adds the first part of the loan this uses an SQL statement to do this~~

~~SelectCurrentLoan.php~~

~~This is a page to select the most current loan from the user, we then usethis information in order to run another PHP file that finishes adding the loan to the user. It gathers information about the first part of the loan that has just been created, the most important thing being the Loan ID as this is needed in the next PHP file.~~

~~InsertLoanContent.php~~

~~This is the page that adds to the rest of loan, this gets the information that is gathered from the previous file SelectCurrentLoan.php as it needs to gather the booking ID to add it the Loan Content.~~

~~GetBookingInfo.php~~

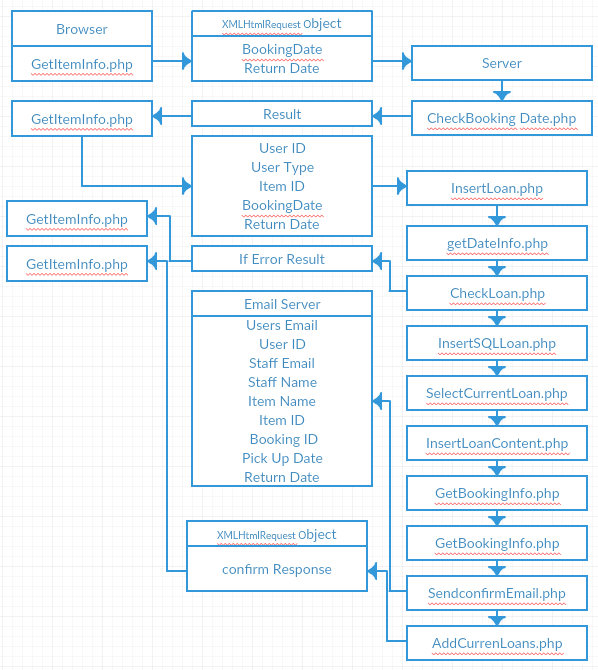
~~This is a simple page to get information about the booking to display it to the user later, as when the booking is a success we will give them a quick prompt to tell them this is information about their booking, it will also give SendConfirmEmail.php information that it can use to send an email.~~

~~SendConfirmEmail.php~~

~~This is the page that sends an email to the User as well as the owner of the item, this lets them know that their request was successfully processed and gives them confirmation about their booking request.~~

~~AddToCurrentLoans.php~~

~~The user should know how many items they have booked, this is a small PHP file that adds +1 to the current loans that the user currently has~~



~~Limitations of our system:~~

~~Although functional our system does have some unfinished features, this was often due to our underestimation of the time and effort required to complete a user story. We made some changes to our original plan which left some features un-used and some things added. There are some limitations of use that are also discussed in this section.~~

~~When deleting items from inventory the image stays in the image folder, if the system were to be implemented by the University it could cause issues with storage in the future.~~

~~Our initial plan was to only have a catalogue of items displayed to users based on their eligibility to borrow them, for example if there were items that were only available to third year and above students then these items would not appear in the catalogue displayed to a first year user. At the close of our project this was not implemented, as we had underestimated the amount of work required to implement other tasks we ran short of time, as an alternative we have a system whereby the eligibility is checked when the loan request is made. This is not our ideal situation as it would be something that might cause frustration form a user’s point of view, a user who has gone through all the stages to borrow an item only to find they are not eligible to borrow it might get annoyed, this is why we had planned to not show these items in the first instance. The bookings page displays the item restrictions on the page but unless the use checks it themselves and realises they are not able to loan the item there is nothing to stop them from going through the booking checks.~~

~~We have the ability to declare that an item must be loaned under supervision, this has not been fully implemented, the plan was to only allow 1 day loans for these items.~~

~~Originally the system was going to impose limits to the amount of items that could be loaned per user, we had a discussion with our supervisor who pointed out that there may be items that needed to be loaned together and if we limited the loans then it may mean that users were unable to borrow items that belonged together. As a result of this we decided not to limit loans. Instead we have implemented a ‘two tick’ system whereby a user requests a loan and it must be approved by the owner, this way an owner can see who has asked to borrow items and can approve as many as they see fit .~~

~~The system is only available to University of Kent members, this means that it could not be used in a situation where an item were to be loaned to someone at a different University. This could be overcome by the use of a guest University of Kent I.T account for the borrower but may be impractical if only granted for the use of this one system.~~

~~Our planning had also included a ‘shopping basket’ so that users could select multiple items to borrow and request them all at the same time, this was not implemented as we decided that requesting one loan at a time was simpler for both user and owner, an owner may wish to refuse a loan of one of the requested items but approve the others, this would make the system more complicated and our aim was for ease of use rather than complexity.~~

Previous systems

~~We looked into similar systems and particularly at the Universities own library system. The library system in use at the university allows the reserving of items, searches for particular items will return the location of the item in the library and the amount of available copies, users can also request a recall of an item that is out on loan if they wish to borrow it. Automatic email reminders are sent to those who have loaned items that are due for return and renewing of items can be performed online. This system works well for the lending of multiple instances of items such as is appropriate for a library, our own project would be used for loans of singular items personally owned by members of the university.~~

~~When thinking about our user interface we wanted something that combined the professional look of the University system along with some visual aspects in the display of items and search results in a similar format to shopping sites such as amazon.co.uk and ebay.co.uk. Most people are familiar with these sites so an interface with similar format would be user friendly.~~

Pair programming

Languages and technologies used and examples of their application inside the system:

~~MySQL database – All of our team members have previous experience of MySQL. We felt that utilizing a little of what we already knew would benefit us for the implementation as the database design would be the ‘backbone’ of the system.~~

~~PHP (Hypertext Preprocessor)~~

~~This was used throughout the website, as the pages needed to read from and insert into our MySQL database hosted on the University’s server, this needed server side code in order to process. An example of a page would be GetItemInfo.php (index…) this is a page which enables the user to view an item’s information and request a loan. This page has a few examples of PHP and the rest is run from an Ajax call, but this page Loads in all the information about an item, this would usually be presented in in a way such as GetItemInfo.php?id=509 this tells the page to call the information related to the item with the ID of 509 as recalled from the database. Other PHP on this page allows the user to select a date, this is done by using the PHP date function, it then uses a while statement, this is done to count days, and adding one day using date to string function in PHP, This was done so it did not need to be manually done by the user or a programmer adding new dates.~~

~~Html 5.0 (HyperText Markup Language)~~

~~Current HTML standard receiving constant updates, we have implemented this mostly for forms to collect information.~~

~~CSS 3 (Cascading Style Sheets)~~

~~CSS was used to style the webpages, this also allowed us to make responsive web design thanks to the power of CSS 3 and HMTL 5.0. Both of these together allowed the site to use a CSS command @media , this allows us to get the dimensions of the web page which was used to make the site look better on mobile devices with smaller sized monitors. Style is important on a webpage as it helps the user navigate from one page to another with elements of the page being placed in easy to find places, accessibility and ease of use was part of our goal for the system.~~

~~CSS also allowed the site to re-size the navigation bar for smaller screens as it turns in into a drop down menu rather than a list at the top of the screen. This takes up less space on a user’s device screen whist they navigate through the site.~~

~~JQuery (JavaScript library’s)~~

~~JQuery are a set of JavaScript libraries that has been implemented and used to make writing JavaScript easier, as they are exactly the same, and has been designed to simplify html. For this project we used version 3.1.0 from google library [1] [2]. We have used JQuery everywhere on the website, as it handles many things, these being:~~

~~Navigation~~

~~JQuery was used on the navigation rather than using the normal <a href=””> this was used because of how the site works. Instead of going to a new page the page that we want to the user to go to is loaded into an element that is on the page, this means that effectively the entire website is on the same page, of course this has its own benefits, as this allows us to hide pass variables from the Address bar on the page, if the user cannot modify the address they can’t access something they should not be accessing, for example if we choose to hide catalogue items from certain users then they cannot modify the address bar to find items that they are not supposed to, of course this can also be done using PHP, but if the user does not have to option to do it then it stops them from even attempting to do that. [image 1] of course the user can still see the id of the item when looking in the developer tools when looking at the networking traffic [image 2] of course if the user is able to access a page for an item they are not supposed to book, they won’t be able to do it anyway because when booking the item it also checks the users Information before they are able to book it.~~

~~Loading Pages~~

~~Much like the navigation we load every page onto one page, this allows the page to feel like it is loading much faster. This works together with the navigation in order to display the pages and for pretty much the same reasons, this works by finding the element that the user has cocked on, and then loading into another element a page they corresponds to the element clicked. Below is an example of how the code works.~~

~~$(document).ready(function() {~~

~~$(".addi").click(function() {~~

~~$(".holder").show();~~

~~$(".holder").load("ajax/Pages/Inventory/add\_inventoryImage.php");~~

~~}); });~~

~~Calling Ajax~~

~~We also use JQuery for calling AJAX and JQuery’s version of Ajax is much easier to use, it allowed us to gather the information from textboxes and selection boxes on the page and then set them in variables to be sent off to other pages, this allowed information to be transferred to other pages without having to be redirected.~~

~~Disabling/hiding and displaying elements~~

~~Of course using this allows the app to disable buttons and other items that are not needed or should not be used by the user at a certain time, there are a few examples of this, and one being on the administrative pages, and on these pages the website disables the textboxes and hides buttons that are not being used. When the edit button is pressed by the user it will enable the textboxes that are relevant to the one that is pressed, it will also display the save and cancel buttons as they would only be needed once they original edit button has been pressed, this is used as the website does not need to show elements and objects to the user that they don’t currently needed to edit.~~

~~Changing and creating ID’s and classes on HTML elements~~

~~This also works in conjunction with hiding and displaying elements. This allows the website to know what element needs what attributes, as if each element had the same attribute it can cause problems scenically when using ID’s as an ID can only be used once on the page. The website has been made to set the ID Attribute to an element only when it needs it, this allows the website to only collect the correct information.~~

~~AJAX (asynchronous JavaScript and XML)~~

~~Ajax is not a programming language, but rather a combination of JavaScript and HTML, it uses browser built-in XMLHTMLREQUEST to request data from the webserver, this allows the site to enter information into the database as well as gather information from the database, although AJAX is mostly used for XML we can also use it to call a PHP file, the page can see the gathered information in variables and send it as normal to a PHP file.~~

~~This works in 6 steps: [3]~~

~~1: an event occurs on the webpage, such as a click event, or a change event, usually on the webpage it would most likely be a click event, once this happens the information is gathered into;~~

~~2: An XMLHTMLREQUEST object, this would be created by JavaScript this is the information that the website creates in order for it to send the results to a PHP file, once this information has been gathered it will;~~

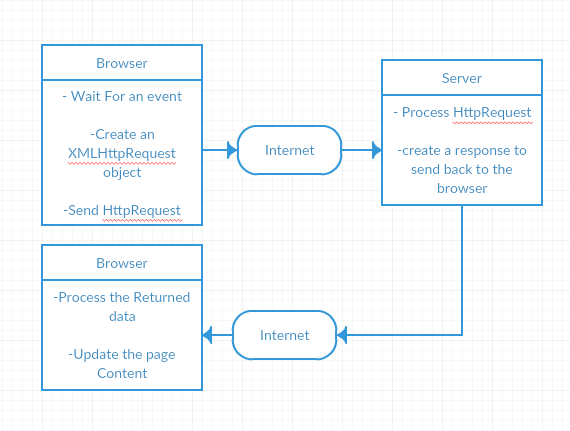
~~3: Send the XMLHTMLREQUEST to the web server, the data that has been sent will then;~~

~~4: Be processed by the server, in the case of BookIT it will use the variables that we have given the PHP file and handle a request with it. For example, it will check all the dates of an item that has been booked in the database, and then check the date that we have sent to it, then it will send a response back.~~

~~5: The response is sent back to the webpage, using Ajax means that the site does not need to navigate between pages, it allows the page to send information to another pages whilst allowing the user to stay where they are.~~

~~6: The response is read by the JavaScript, here the JavaScript decides what to do with it. We have programmed it to display or utilise the information that has been returned to it. For our date example it can disable and enable a button on the webpage, this being the button that allows the user to book the item or not.~~

~~This particular process is explained in further detail later in our report as it is an important process in the booking system.~~

~~~~

~~Ajax was used throughout the project as this allowed the pages to move data from one page to another without having to load multiple pages. We utilised Ajax all over the website, this was an extremely useful and powerful tool, especially on the page where we needed to check if the user could book the item. If we had done this with normal PHP they would then be redirected to another page or just receive an error when they tried to book an item that has already been booked for the selected days. As we could send this information and get a result back without moving from the page, it made it easier and faster for the user to navigate and see if they could book things or not.~~

~~Ajax is a good way to cut loading times, the site only loads particular data when it's needed, using Ajax means that we don't use a large amount of data, as the site does not need to load an entire page of code that does many things, as the information is only called to the page when it's needed and not it being loaded to the page without it being needed.~~

~~Ajax was a challenge as the team previously had no knowledge of this methodology, one member of the team (James) took it upon himself to learn about Ajax and then spent time with each member helping them to learn and implement it. It was a steep learning curve to adapt what we already knew into this new way of building our pages for some of us as we had become comfortable with the way PHP worked but it was definitely worth the effort of learning. Our system has benefitted from the sleek way it loads data and our team has benefitted from learning a better way of building a site. Ajax has also benefitted our project in a security aspect; a user cannot ‘bookmark’ a page and return to it at a later time without logging in via SSO, a php page would display a specific address that can be stored, Ajax only ever displays the ‘front’ page of the site, the user is never informed that a page address has changed.~~

~~User feedback~~

~~When demonstrating our library system at the poster fair we got some good feedback. People generally thought it was a good idea and provided us with helpful insights about the way it looked and the ease of use.~~

~~We had a user who mentioned that the booking button was a little confusing, it had two words on the one button – “Available – Book” In response to this we have amended the button to only display one word. The same user also questioned our implementation of the date selector and asked why we hadn’t used a calendar type picker as this would have allowed users to see the days of the week. It was explained that we had started off trying to use a JavaScript calendar but for some reason it would not work the way we wanted it to, any page changes rendered the calendar useless, we have still yet to work out why this was happening, as a result of this issue we had to scrap the calendar and implement our own date picker.~~

~~Another user enquired as to whether we could allow infinite booking periods, he liked the system as he himself had items that he lends to people and has often lost track of them once they had left him. He also asked if it was possible to transfer ownership of an item, this would be possible by the deletion of the item by an original owner and, if the new owner wished, the creation of a new listing by the new owner.~~

~~The ‘Shed’ at Canterbury were enthusiastic about the system but would have liked to see it allow for multiple listings of the same item, for example if they had 100 Raspberry pi’s they wanted to lend could they just list once with a quantity of 100. This was not something we had anticipated when setting the user requirements, at present items listed are singular and uniquely referenced in the database so this would be a challenge at this stage to implement, however this could be done by the use of a batch ID system if a model number were supplied but would require some changes to our database and possibly the inclusion of a separate search area on the site.~~

