|  |  |  |
| --- | --- | --- |
| **What did you do that made a meaningful contribution to the teamwork in producing the final solution** | **Evidence of that work (artefact)** | **How did it fit in the team work/project work?** |
| **RELEASE 1:** | | |
| Wrote and maintained the sprint 2 plan, devising the tasks for each user story and estimating the expected hours required for each task.  Also contributed heavily to the development of user stories both for our team (86) and the client team(85). | See the sprint 2 plan section in the document “doc\Release and Sprint Plan.docx”  GitHub issues tracking found here:  <https://github.com/tross78/IFB299/issues>  Googledoc Link to Our user stories: <https://docs.google.com/presentation/d/11iKWcVbtcOqUutTEVVaRXPkrSf9qpAnmL-eJ8mbwEIo/edit?usp=sharing>  Googledoc Link to Client Team’s user stories: <https://docs.google.com/presentation/d/1NN12EVXKxl3eQ9IHSNwWb-D1VbAj7mHJuq28nINv29I/edit?usp=sharing>  Googledocs provides a history of edits, which you can view in both documents to determine contributions made. | After dividing up the agreed upon user stories for the second sprint and writing out the individual tasks to be completed for each, I was able to then inform the group of the work to be done for sprint 2 and then assign user stories to each group member.  I also calculated time estimates, velocities and as we progressed tracked the actual hours spent making sure to post new issues to the GitHub Issues tracking section of our repository and inform the group via slack of what tasks were left undone. |
| Waitlist and course capacity functionality including checks and testing. | See lines 76 – 254 of ‘add()’ function in: “app\Controller\EnrolmentsController.ctp”  And lines 8 – 21 of ‘app\View\Enrolments\add.cpt’ | With that functionality: courses now fill up, have separate capacities for each gender/role, students go on a waiting list if the course they want is full, and the user is informed with error messages if a particular role is full or if they have been waitlisted, or already enrolled etc. |
| User permission enrollment checks for “Role”. | See lines 47 – 53 of ‘app\View\Enrolments\add.cpt’ | With this functionality implemented it ensured that only the users with the correct authorization could enroll in courses in their available roles as intended. This makes sure every user type has their separate permissions. |
| Co-authored the written content on the home, contact and donations pages. And set up the email address used for the site’s email notifications. | See images 01.png, 02.png, and 03.png. under “doc\andrew.grant-portfolio\”  Email address as also seen in image 03.png  (Had also contributed to written content for the about page and more of the donations page in sprint 1 but that work has since been improved upon and overwritten by other members of the team) | Fleshing out the initial website’s pages with information about the center, their services, practices, and their contact information. Also setting up a working communication channel in the forms of the email address that will be used more frequently in release 2 when we implement email notifications. |
| Organized development team communication, group meetings, Facebook group, meetings with the client teams and presented all the project demonstrations thus far. | Group meetings held each Wednesday in person from 9:00am-10:00am before workshop.  Organized two online group meetings over discord chat channel/server. (24/08 ~ 2hrs. 06.png) and (18/09 ~ 2.5hrs. 07.png) – (images here are examples of what was posted in the chat during those times. Unfortunately, this is the best I can do for evidence of these meetings)  Facebook group management and more screenshot evidence of organization of meetings. 04.png and 05.png  As for evidence of group meetings, as you are aware I have predominantly been the presenter for our group and have had the most interaction with the client team, presenting demonstrations of the sit and asking questions of you as the tutor. | Having someone in the team who is able to organize the group, keep track of our work, set goals, arrange meetings and maintain a general understanding of all parts of the project is essential for smooth, focused development, minimal confusion/complications and general interactions with the client. |
| **RELEASE 2:** | | |
| Wrote and maintained sprint 3 & 4 plans, devising the tasks for each user story and estimating the expected hours required for each task.  Then added user stories as well as other issues to issue tracking on GitHub repository. | See the release 2 section and sprint 3 & 4 plan sections in the document “doc\Release and Sprint Plan.docx”  GitHub issues tracking found here:  <https://github.com/tross78/IFB299/issues>  (Most are under ‘closed’ at this point) | With Release 2 I continued my maintenance of the sprint plans, user stories and issues, writing out the individual tasks to be completed for each user story.  I also calculated time estimates, velocities and as we progressed tracked the actual hours spent, making sure to post new issues to the GitHub Issues tracking section of our repository whenever they arose.  This was a significant contribution as it helped organize and prioritize our team’s remaining tasks and allowed each member to keep team members focused. |
| Added in Class/Timetable functionality for enrolments. Upon enrolling, students and teachers choose tasks and their timetables are populated. | Seen in images 08.png & 09.png  See also lines 64 – 85 of ‘app\View\Enrolments\add.cpt’ | This functionality added more substance to our website, enriching the overall user experience and adding more complexity or depth to the enrolments system.  This feature would have been expanded upon in release 3 to include additional features and customization options. |
| Overhauled other index and view pages visually in accordance with the design established by Tyson’s overhauled user pages. | Seen in images 09.png, 10.png & 11.png  See also: ‘app\View\Enrolments\index.cpt & view.cpt’  ‘app\View\Courses\index.cpt & view.cpt’  ‘app\View\Enrolments\index.cpt’  Finally the ‘view()’ function at line 136 in: “app\Controller\CoursesController.ctp” | After Tyson overhauled the user pages to make them more visually appealing and functional, we decided that the rest of the site could use a ‘face-lift’ as well, not just for the sake of consistency but for the benefit derived from the visual appeal of the new designs.  The overhauled designs of the view pages were not only more pleasing to the eye, but also functional; displaying more detailed information, user lists with working links etc. |
| Backend function checks to ensure courses and enrolments can’t have inappropriate data entered upon adding and editing enrolments and courses. | Too many examples to show with images, but images 12.png, 13.png & 14.png give the general idea. More error messages and boundary cases can be seen when using the website.  Lines 190-198 & 253-261 in “app\Controller\CoursesController.ctp”  Lines 220 – 234 & 350 – 354 in “app\Controller\EnrolmentsController.ctp” | Fully tested and developed checks to catch input errors will not only make sure our websites forms are filled out correctly ensuring no ‘bad data’ being store in the database, but also provides the users with feedback in the form of visible messages informing them of their mistakes. |
| Implemented proper server permissions into the database and website. | See 15.png for the option managers get when editing a user. That is how users can be changed to server permissions through the UI.  Line 36 in: ‘app\View\Users\edit.cpt’  Lines 269 – 272 in “app\Controller\EnrolmentsController.ctp” for automatic promotion to sever permission when they have completed a 10 day course as a student.  Apart from that, checks for the server permission are carried out across the entirety of our code in view pages, and controller pages in numerous locations. | Prior to this implementation, the sever permission was only set inside backend code in ‘EnrolmentsController.ctp’ and was unreliable and un-settable by a manager account through the user interface.  With this implementation, it was not only easier for us a developers to program for users with that permission now that it was properly stored in the database, but the UI integration means it was good for clients/end users. As expected Servers have different options that standard students and Manager accounts. |