|  |  |  |  |
| --- | --- | --- | --- |
| Project Plan | | | |
| Project Name | AT1 Wiki Prototype | | |
| Date | 9/8/2022 | | |
| Developer Name | Corin Little, P453208 | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| Project Requirements | Requirement gathering & analysis | Analysis the task to summarise the base requirements given by the client to complete the project. | Input: Provided project details (i.e. the scenario)  Output: List of requirements |
| Determine User Interactions | Requirement gathering & analysis | Analysis the task to figure out how the user will interact with the application. Should also note possible invalid inputs. | Input: Requirements list  Output: List the interactions & the resulting outputs |
| Basic GUI Design | System Design | Create a very basic design of the form (GUI) this helps give a better idea of what is required for the form to function as will show most/all the elements that the users can interact with. | Input: Requirements & interactions list  Output: Basic form design & refined interactions list |
| UML Diagram/ Methods | System Design | Using the previous tasks information figure out what methods & global variables you will need. Determine what event handlers will be needed to initiate the methods. | Input: Previous tasks  Output: UML diagram, methods & variables.  Comments on what methods will do. |
| Coding | Implementation | Start coding the form, starting from the display (ListView) to get a visual representation of records as soon as possible for easier error checking. | Input: GUI, methods, variables  Output: Working form |
| Error Handling & Testing | Integration & Testing | Begin testing making sure everything out correctly & minimising what errors can occur, handling the ones we can prevent. | Input: User interactions  Output: Error handling |
| Invalid Input Handling | Integration & Testing | Handling user input by prevent them from putting in invalid inputs, informing users of what should be input & why input was invalid. This can be done using the status strip. | Input: User interactions  Output: Invalid input error handling |
| Clean-up | Integration & Testing | Clean up the code, check that everything is properly commented, remove any code just use for testing. | Input: Application  Output: Finished app |
| Finished Development | Deployment of System | Submit application to client. |  |
| Fixes | Maintenance | Fix any errors discovered by client. |  |