1. Gantt Chart

The Gantt Chart should be updated to include more detail for all milestones for the remainder of the project.  It should also record actual vs projected finish dates for the Planning & Design stage.  The Gantt chart needs to be accurate - faking it will result in lower marks.

1. Logbook

The Logbook needs to be continually updated during the project.  Since it is online, the teacher will be keeping track of it during this stage.  Faking it will result in deduction of marks.

1. Client Communication

Client communication needs to continue throughout this stage, especially with respect to prototypes.  Failure to take into consideration the client’s perspective may result in a lower quality product.

1. IPO Charts

The IPO Charts should be updated to reflect any changes from the Defining & Understanding stage.  In addition, it should now reflect all subroutines in the project.  Note that the Process column does not need to be highly detailed, but merely describe what each subroutine does.  The Input and Output columns need to be accurate.

1. DFDs

The DFDs need to be refined to the lowest level in the system.  All subroutines in the project should be represented by a process in a DFD.  In addition, all changes from the previous stage need to be reflected.  The DFDs need to remain consistent with all documentation.

1. Structure Charts

The Structure Charts need to be refined to the lowest level in the system.  All subroutines in the project should be represented in a Structure Chart.  In addition, all changes from the previous stage need to be reflected.  The Structure Charts need to remain consistent with all documentation.

1. System Flowcharts

The System Flowcharts need to be updated to reflect any changes made to the system.  They should remain consistent with all other documentation.

1. Data Dictionaries

The Data Dictionaries are vital in this stage of the project as they ensure that all components of the project will interact with each other correctly.  All data that is passed between subroutines and is stored in the system must be in the Data Dictionary.  Any constants must be also be represented.  Variables local to a subroutine do not need to be defined.

1. Prototype Screenshots

Once the Storyboards and Screen Designs have been completed in the previous stage, a Prototype can be created for user feedback.  Prototypes can range from accurate images of all screens in the project, to fake user interfaces in Powerpoint or HTML, to actual user interfaces with limited functionality.  Prototypes are invaluable in this stage as they will ensure that, once the clients have signed off on them, they are the model to which the user interfaces are crafted in the next stage.  Accuracy is key here.  This is not to say that they will be exactly the same in the final product, but they must give an accurate user interface experience. Getting client feedback from the prototypes is vital, as they can refine requirements that the developer may miss.

1. Algorithms

Algorithms are vital to the understanding of the logic behind the project.  By getting the algorithms right, the implementation of the project becomes very easy.  For full details about algorithms, you should ensure that you are very familiar with the course specifications document.

You do not need algorithms for your entire project.  However, you need to include enough subroutines to satisfy the requirements of the course.  Your algorithms must show selection, repetition, file/database reading & writing, the use of external modules, and anything ‘complex’.  All algorithms must be ‘correct’ and easily understandable.

1. Specifications & External Modules

You should clearly specify the software and hardware specifications of your system.  They should clearly and precisely detail what operating systems your project will run on, what browsers, what software needs to be pre-installed and/or what hardware it will run on.  Your system should aim, in general, to run on as wide a variety of systems as possible.  The final system will be measured against what is stated here. You should also clearly list the external modules and sections of code your system will rely on.  References to the modules and code need to be included.  The language(s) the system will be run on should be detailed here as well.

1. File & Record Definitions

The precise format of all files and records need to be clearly and precisely defined.  For example, how high score tables are stored, user details on a database, etc…  It should be easy for anyone reading the format to write code to read from or write to any files or databases in your project.  If this changes in the future it will need to be updated.  You may consider using EBNF at this stage for any files.

1. Test Data

Your project needs to be designed so that it can be tested in a modular way.  As such, test data for all non-trivial algorithms needs to be designed at this stage.  By creating test data now, you will ensure that all code written will operate exactly as originally specified. Your project will be evaluated as to how well it passes the test data created in this stage.  If there is insufficient test data, then your project may be evaluated poorly.

1. Reflection

Reflection is a vital part of the learning process.  Innumerable studies and articles have been written on the benefits of reflection in all parts of life, especially education.  Reflection is integral to the Cultures of Thinking ethos at many schools, which is why it is consistently an assessable component in PoUs.  The most successful students will be able to reflect on their learning processes well.  The key to being able to reflect well is to ask yourself the right questions (and of course to then answer them well).  Some question you could ask yourself include:

* What have I learnt?
* How has my thinking changed from the beginning of this process?
* What problems have I encountered?
* How have I overcome issues?
* What issues have I not been able to overcome?  Why?
* What have I succeed at?
* What am I happy/unhappy with?
* What would I change if I had the time?
* What would I do differently next time?
* How do I feel about this?

There are of course many other questions that can be asked.  A successful reflection can be 500 words (or even more) in length.  Writing a couple of paragraphs is never a successful reflection.  Also consider that it is the quality of the answers that matter.  Keep in mind, that while you can reflect on other people, especially in group assessments, the reflection is all about YOU.

1. Website Screenshots

The Website is the repository of all the information relating to the project.  In the future, it will contain links to download (or access) the project, installation instructions, user documentation, screenshots, description of how it works, and possibly development information (such as the project log).  It does not need to be professional, but should be clear and pleasing to the eye.  It is suggested that use is made of a web template, so that only content needs to be added.  In the future, a successful website would probably be multiple pages, but at this stage can be one page in length.  Screenshots (as well as the link to this page) should be submitted.  The website can be hosted on Hippo or on an external site such as GitHub or Google Code.