

# Basic Project Plan

By default, each project should roughly follow this basic structure, although the student is welcome to suggest changes to this structure that may be appropriate for their specific project.

Below, deliverables are listed in the default order in which initial drafts of them should be completed and turned in to the supervisor. However, each deliverable document may continue to be revised throughout the rest of the semester, before inclusion in the final report. Suggestions as to length and time spent on each deliverable are given, although these may vary from project to project.

## Generic List of Deliverables:

1. **Project Proposal** (1-2 pages. 3 days) - In the first week of the semester, the student should turn in a general, high-level description or summary of what they wish to accomplish in this project. This can be informally written text. As the student learns, the goals of the project may shift. Over the course of the semester, the student should adapt the original proposal document to become an evolving **project summary**, although it is often helpful to keep the original proposal around for reference to review how things have changed.
2. **Project Web Site** (3 days) - Also in the first week, the student should construct a web page and subdirectory on which all deliverables relating to the project should be posted as soon as they are ready. This will allow the student and supervisor to easily show people what the student is doing. Also, by gathering all materials onto the web page, the project can easily be archived by the supervisor so that it will remain after the student leaves, for future reference. You may use Blogs also.  
Each student is required to have multiple reliable backups of their work as proof of work done even after submission. These backups are to be kept for up 2 months after the end of the semester in case of any queries. No excuses like I lost my copies or the dog ate my homework or the hard disk crashed shall be accepted.
3. **Project Plan** (1 page., 3 days) - Like this "basic project plan" you are now reading, the project plan should list all deliverables the student plans to

produce. However, it should be tailored to the project's particular needs, and specific deliverables associated with specific steps in the project's development should be added. *Very important:* The project plan should include a complete schedule or calendar for the semester, pairing each deliverable with a specific due date. The plan should explicitly account for any planned vacation time or time reserved for working on projects for other courses. You can use a project planner to prepare a formal Gantt chart for their project plan, if you wish, although it is not an essential requirement.

4. **Background & Related Work** (2-3 pages, 1 week) - Here, the student should dig and do research to find out about a variety of existing projects/products that are most closely related to what they are doing, and to learn any general background needed before they can start work on the project. Often, suggestions from the supervisor about places to look for sources may be helpful. The best place is google.com for details and yahoo.com for introduction to new subjects. The student should compare and contrast what they are trying to do in their own project against what has been done in related projects. Are they adding new features, or providing a simpler or more efficient implementation, or using a different technology? The project should provide some kind of benefit to society or solve some kind of problem which exists in reality.
5. **Technologies to be Used** (1-2 pages., 1 week) - Here, the student should research the different alternative options available for underlying technologies to be used in the project's implementation. The student should discuss the pros and cons of the different available options, select their preferred technologies, and give the reasons for this choice. This is also a good time for the student to set up their development environment, testing tools, etc., and ensure that they are working properly.
6. **Functional Requirements / Specifications** (2-4 pages., 3 days) - After learning about related work and the capabilities of their tool set, the student is in a good position to write detailed functional specifications for what their system will actually *do*. This should expand on the general goals stated in the project proposal/summary, but go into considerably more technical detail. However, it should remain focused on a "black box" view of the functionality of the system as seen by the outside world, and not yet delve into details of the internal workings of the system.

7. **System Architecture** (2-4 pages, 1 week) - This is a high-level description of the planned structure of the system, showing its major components and the interactions between them. A block diagram is often given, with accompanying textual discussion of the components and their interactions. It is often a good idea to show several possible, alternative architectures, and discuss why a particular one was selected.
8. **Interface Specification** (2-4 pages, 1 week) - This document gives complete technical details about the system's internal and external interfaces. Actually, the external interface may have been already completely described in the functional specifications earlier, in which case this document only needs specify the internal interfaces, that is, between the internal components of the system. This should include complete descriptions of all communication protocols, data formats, APIs, public classes & methods, *etc.* for each component.
9. **Component Designs** (2-4 pages., 2 weeks) - The size of this deliverable may vary considerably depending on how many components there are. It should describe the overall approach taken in the implementation of each component (algorithms & technologies used, *etc.*).
10. **Component Implementation** (1-2 pages, 2 weeks) - Describe the outcome of the component implementation process - how many lines of code were needed, what problems were encountered during implementation, *etc.*
11. **Component Test Results** (1-2 pages, 1 week) - Describe the testing and debugging process for the individual components, problems encountered, any reimplementations needed, final results of testing and/or performance measurement. Do the components satisfy their interface specifications?
12. **System Integration Report** (1-2 pages, 1 week) - Describe any additional problems that were encountered during the process of putting together all of the components and testing them within the framework of a complete system. Does the system as a whole satisfy its external interface specifications?
13. **Final Report** (10-20 pages, 1 week) - The final report should consolidate & summarize all of the above documents, in their final form after whatever needed revisions have been applied.
14. **Project Archive** ( 3 Days ) - It is required that student turn in a complete

archive of all deliverables, web documents, design documents, source code, working executables, etc. associated with the project in a complete archive - a ZIP or TGZ file, a CD-R, a readable, on-line directory structure, etc.

Note that the total time listed above is 92 days, and this does not include time to be spent on vacations, work for other classes, etc. So it is a tight schedule for a 15-week semester, especially since unforeseen problems often occur during the implementation phase. The student may wish to eliminate some deliverables from their project plan to speed up the process. The final project is due 168 hours (2 weeks) before the last day of classes.

**Warning : Anything submitted after this time can only be considered for grading in the next semester.**