Functional specification document for [Project]

Team:

# Project scope

*{Provide a short description of the software and its purpose, including relevant benefits, objectives, and goals.}*

# Project features

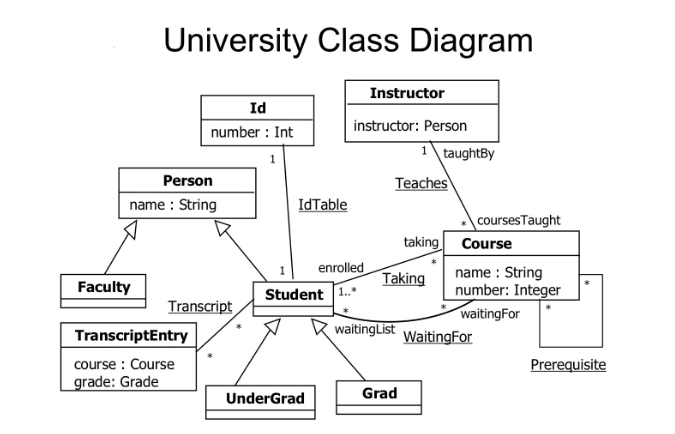
*{Summarize the major features the Project contains or the significant functions that it performs or lets the user perform. Details will be provided in other section, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the document. (Suggestion, link your project features with Test Case document.) A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram,* *is often effective.}*

Figure 1- Class Diagram example

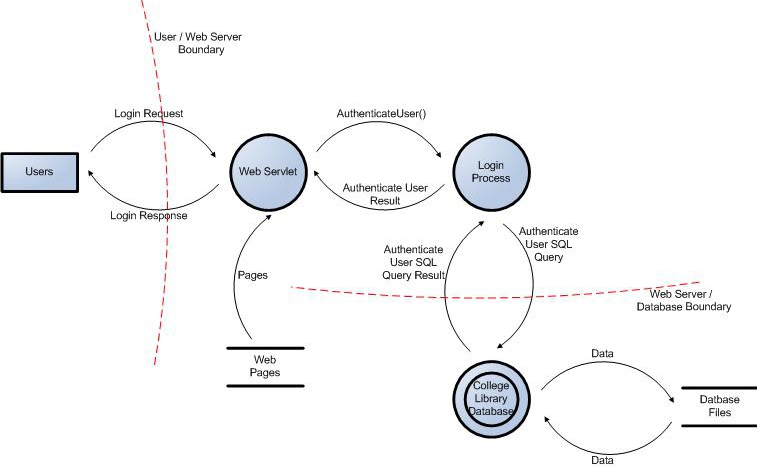


Figure 2 - Data Flow diagram example

# Operating Environment

*{Describe the environment in which the software will operate, software components or applications with which it must peacefully coexist.}*

# System Features

*{This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.}*

## 3.1. System Feature 1

*{Don’t really say “System Feature 1.” State the feature name in just a few words.}*

### 3.1.1 Description and Priority

*{Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority.}*

### 3.1.2 Stimulus/Response Sequences

*{List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.}*

### 3.1.3 Functional Requirements

*{Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.}*

*{Multiply section 3.1. for as many features as the application has.}*