

Project 1: Some Guidelines for Your Design Report

This handout aims at getting you a step closer towards your implementation of project 1. You must demonstrate that you understand the project's design and starter code, including the purpose, responsibilities and specific implementation details of each package. At the end of the design document, you should be able to easily commence coding your project components. Your design report should include the following two sections:

1. Project Design:

In this section, you should relate this project to Lecture 3 (Architectural models of Distributed Systems). *Briefly* answer the four questions posed in the class: What are the communicating entities? What is your communication paradigm? What are the roles and responsibilities of each entity (and, accordingly, the dictated architecture)? How would you place such entities over a heterogeneous distributed infrastructure with computers and networks of varied computational, storage and communicational powers?

2. Project Implementation

Below are some questions you may want to address in your document. Please spend some time pondering over the above-mentioned questions. They will help you attain a good understanding of the project and deeper insights into the required implementation. Once finished with the design document, you should be able to start the implementation phase with no vagueness or confusion.

- **RMI package:** What is the purpose of Stub.java and Skeleton.java? What methods should you declare and implement in order to achieve the intended functionalities? Which Java class do you use for representing a Stub? How do you use this class?
- **Common Package:** What is a path? What is the purpose of each utility function? Where would you make use of each utility function?
- **Naming Package:** What is the logic of each method declared in each interface? What are the other classes needed in this package (if any)? How to create the required Skeletons? How to represent the directory tree and the associate meta-data (i.e., which Java Collection to use)? How to deal with invalid paths?
- **Storage Package:** What is the logic of each method declared in each interface? How to create the required Skeletons and Stubs? What happens pre- and post-registration? Which Java class do you use for reading and modifying the contents of the temporary directory?