**Software Architecture Document**

**BookKeeper Project**

**Architectural Representation**

This SAD summarizes the architecture from multiple views. These include:

* Logical View
* Deployment View
* Data

In addition, this SAD references the Supplementary Specification where you will find architecturally-significant requirements recorded. It also summarizes the key architectural decisions in a format called a technical memo – a short description of motivation, which may help when you need to modify the architecture.

NOTE: This is a very early iteration of the SAD. Many aspects of this document are not yet complete. Most of the features and diagrams of this document will be completed in the next iteration.

**Architectural Factors**

These are recorded primarily in the Supplementary Specification, and one should to that when any question concerning architectural factors arise.

**Technical Memo**

**Issue: Reliability – Recovery from Remote Service Failure**

Factors:

* Robust recovery from remote database failure
* Robust recovery from remote server failure

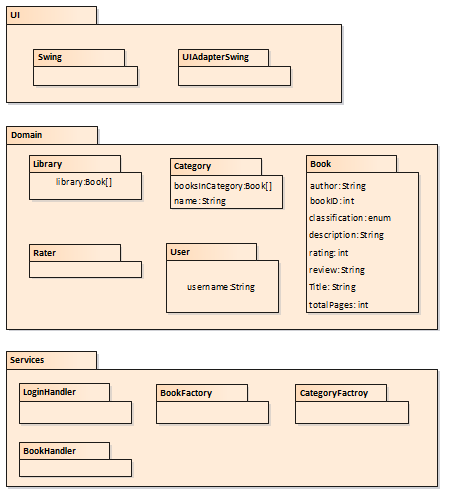
*Solution*

The solution to this problem is to mainly employ a strong central database and server that will be able to hold and process a large amount of requests in a short amount of time.

*Motivation*

The motivation for this is pretty simple. People will want to be able to access the application at all time. Being unable to will lead to customers being unsatisfied with our software, and they will quit using us in favor of others.

**Logical View**

****

***Discussion and Motivation***

A classic layered architecture will be used. No application layer of sessions objects was inserted between the UI and Domain layers, as the system operations are simple, without much workflow coordination.

**Deployment View**

***Discussion and Motivation***

The application’s database, which contains all user’s books and categories, are deployed to single computer as performance and reliability goals should not be too hard to maintain. All features and operations occur on the central server, and the client is used only to communicate with a user.

**Data View**

***Discussion and Motivation***

The use cases we have detailing the creation of categories and books give the greatest amount of detail in how our major data flow works.

**Use-Case View**

Like what is said for the data view, any use case that deals in the creation of something is by far the most impacting to the architecture of our application, as the majority of actions performed by users on our application will be to create objects such as categories, books, reviews and ratings, etc.