CSCI 360 Software Architecture and Design
Spring 2008
Dr. Bowring

Team Project – SafeMessage System

SEE DESCRIPTION PAGE

Your team will produce a number of deliverables over the next few weeks that will serve as chapters in your final deliverable. The final project deliverable will be a compilation of these chapters with appropriate title pages, table of contents, introduction, conclusion, and team self-evaluation. The final document will be a single PDF document and submitted electronically via email. You are free to expand the contents of your submittal to suit the professional goals of your team and to reflect the quality of your design. Your deliverable will contain at a minimum an identifiable SRS, SAD and DDD.

All questions regarding this project must be submitted in writing via email, so that all teams may be electronically notified of the answers.

Schedule of Deliverables

19 March: Requirements [TeamName]_SRS.pdf

User-level Requirements including functional, non-functional, and data requirements
Use Case Model
Use Case Descriptions
Design Rationale
→ SRS illustrated with UML diagrams

31 March: Architecture [TeamName]_SAD.pdf

Conceptual Model
Propose two architectures and show decompositions and class models
Architectural Profiles and Scenarios (weighted) in prose as well as a Utility Tree for each
Choose one architecture and provide rationale
→ SAD illustrated with UML diagrams

14 April: Detailed Design and Final Project Deliverable [TeamName]_FINAL.pdf

Class Diagram(s)
Sequence Diagram(s)
State Chart(s)
→ DDD illustrated with UML diagrams
➔ Completed project including SRS, SAD, DDD

16,21,23 April : Presentation

20 minute presentations of your project (each member presents)
→ Files prior to presentation

SafeMessage System Description

A software development company intends to build a secure instant messaging product for use by bankers, brokers, lawyers, and other professionals with high communications confidentiality and integrity needs.

The *SafeMessage* system will allow individuals to communicate over the Internet with others using a peer-to-peer protocol at one of three security levels. Messages must be encrypted at every security level and the communication path between machines must be unspoofable. The three security levels are:

- *High*—At this level there must be an unspoofable communication path between users and message delivery must be non-repudiable.
- *Medium*—At this level there must be an unspoofable communication path between users, but no guarantee of delivery.
- *Low*—At this level there is neither an unspoofable path between users nor a delivery guarantee.

Thus, at the low security level, messages are sent securely from one machine to another with no guarantee of delivery, at the medium level they are sent from one user to another with no guarantee of delivery, and at the high level they are sent from one user to another with a guarantee of receipt.

Pairs of users must establish sessions with other another at a security level. All communication between the two users during the session then occurs at the established security level. Users may have concurrent sessions with many users at different security levels. Users may broadcast a message created in one session to other sessions provided the other sessions have a security level at least as high as the session in which the message is created.