CS7NS6 Distributed Systems

2023-2024

Exercise 3

Implementing a Globally-accessible Distributed Service

A large global café chain, with a large and highly mobile set of customers, wishes to introduce a 'friends and family' loyalty-card scheme to reward its customers with discounts on their in-store purchases based on (some function of) their transaction/purchase history. This will allow up to four people to share a loyalty-card account such that any of them can earn and obtain discounts, e.g., if Alice earns discounts in Store1, Bob could immediately avail of those discounts in Store2.

Implement a globally-accessible system whose primary purpose is to manage participating users' accounts including their relevant transaction histories and the discounts that they have accrued. Participating customers may use a mobile phone 'app' that acts as a contactless loyalty card through which they can identify themselves at a point of sale anywhere in the world so that their transactions can be recorded and/or discounts applied.

Your design should focus on providing appropriate levels of performance, scalability, availability and reliability, considering that your service may need to grow to be used by millions of users throughout the world.

Note that there is no 'right' design for such a system. In particular, you should begin by specifying a reasonable set of requirements for the service taking into account the level of dependability that might be expected by users of this service. Your design should then focus on how the system (and any individual subcomponents that you identify) might be designed to satisfy the requirements that you have specified. Be careful not to over engineer the system since providing higher levels of dependability is usually detrimental to the cost of operating/using a system.

The design will be assessed on the reasonableness of the requirements specified and the appropriateness of the corresponding design choices.

In designing the system you should consider whether and where paradigms/techniques such as transactions, process replication, checkpointing/message logging, caching, load balancing, replication, and/or partitioning as well as other appropriate techniques might be used. It is expected than any reasonable design will use some but not all of these techniques. Depending on the techniques being used you will need to consider, for example, the required degree of isolation, the degree of replication, and/or the level of replica consistency that are appropriate. Be explicit about any assumptions you make about the underlying failure model and the faults to be tolerated by the system.

You should also consider the expected usage and failure patterns of the system and how they might influence the design, for example, are users collocated or widely distributed? do more users read data or write data? is data shared between multiple users or not? are some data items more frequently used (read and/or updated) than others? are failures expected to be frequent or rare?

You may use any programming language, middleware, and/or development methodology that is agreed by all members of the group as you see fit. In demonstrating your solution you will be required to demonstrate its operation in a variety of failure scenarios, so you should build a deployment framework that allows such scenarios to be tested and eventually demonstrated.

Interim deadline

Please submit a short report outlining the technical architecture of your service and the technologies that you are planning to use to build it before Friday the 15th of March 2024. The report should be signed by all members of the group.

Final deadline

Each group is required to present and demonstrate its solution at a time to be advised, and to submit a report on the group's solution by 5.00pm on the 5th of April 2024 as a single PDF file by email to vinny.cahill@tcd.ie with the subject line "Group <number> CS7NS6 Exercise 2 Report". Further details will be notified on Blackboard.