Requirement Analysis of project

Tittle: Process Synchronization

Problem statement: Safe apartment inspection problem with tenants and real-estate agents synchronize so that:

• a tenant cannot view an apartment without an agent,
• an agent cannot open the apartment without a tenant,
• an agent leaves when no more tenants are in the apartment,
• an agent cannot leave until all tenants in the apartment leave,
• once an agent opens the apartment, she can show the apartment to at most ten tenants, and
• at most one agent can open the apartment at a time.

Our job is to write a program that (a) always satisfies the above constraints and (b) where under no conditions will a deadlock occur.

Requirements: Our solution must use binary semaphores, should not use busy waiting, and should be deadlock-free, abstract class, switch case, aptsim, mmap(), starvation free ,windows API , priority inversion, object factory, aging, nanosleep(), friend functions ,etc.

Pradyumn (211127) : Tenant class , mmap, friend functions ,object factory, aging, windows API, binary semaphores.

Nandini (211131) :Agent class, priority inversion, friend functions ,switch case ,object factory, windows API, starvation free.

Karan (211445) : Owner class, aptsim, friend functions, abstract class , nanosleep, windows API.