Solution 2

Definitions

Constraints

Constraints are defined in file as follows:

```
#define QUEUE_MAX 200000
#define ZONES_MAX 1000
#define COMPANY_MAX 1000
#define STUDENT_MAX 10000
#define BATCH_MAX 5 // number of batches at once
#define BATCH_MIN 2
#define TIME_BATCH_MIN 1 // time to make the batches
#define TIME_BATCH_MAX 5
#define VACC_MAX 20 // vaccines per batch
#define VACC_MIN 10
#define STUDENT_WAIT_TIME 10 // max random time that a student wait before
entering gate
#define TIME_VACCINATE 1 // max time to vaccinate a student
#define ZONE_REGISTER_TIME 8
```

Why not busy waiting?

- It is slower, as cond variables put thread to sleep while waiting
- Along with being slower it has much higher CPU consumption
- Conditional variables have more practical applications
- Conditional variables have more flexibility with both broadcast and signal functionality
- More concise and readable code while easier to implement

Global Variables

```
int waitingStudents; // to calculate slots to put up
int numCompanies, numZones, numStudents; // total number
int vaccineUsed[COMPANY_MAX]; // tracks vaccine batches used for each
company
struct vaccineData vaccineQueue[QUEUE_MAX]; // companies put vaccine into
the vaccine queue (O(1))
int readPosVaccine, writePosVaccine; // for the queue
bool vaccinated[STUDENT_MAX] = {false}; // which student did get
vaccinated
double vaccineGiven[STUDENT_MAX]; // stores the pSuccess of vaccine for
student
struct zoneArgs *zoneData[ZONES_MAX]; // stores data about zone including
available slots
```

Mutex Locks

- vaccineQueueLock - Synchronizes the access to vaccineQueue, vaccineUsed among Zones, Companies
- waitingStudent - Synchronizes the access to waitingStudents
- studentLock - Synchronizes access to vaccinated, vaccineGiven among students
- zoneLocks[] - one for each zone, enables multiple students to go to different zones at the same time

Conditional Variables

- vaccineNotUsed (vaccineQueueLock) Companies wait on this while their medicine is being used
- noVaccineAvailable (slotQueueLock) Zones wait on this if no vaccine
- noSlotsAvailable (slotQueueLock) Students wait on this if no slot available
- notVaccinated (studentLock) Student wait on this when assigned but not vaccinated

Working

Companies

1. When it is created it gets an ID and pSuccess
2. `addToVaccineQueue()` makes vaccines and adds them to queue
3. `addToVaccineQueue()` uses the vaccine queue lock to add the vaccines to the queue
4. Then company waits on `vaccineNotUsed` CV and checks `vaccineUsed[companyId]` until all its vaccine batches are used
5. When all of its batches are used it again goes `step 1` to create more vaccines

Zones

1. When a zone is created it gets an ID
2. Initially it has 0 vaccines
3. Zone calls `getFromVaccineQueue`.
4. `getFromVaccineQueue` waits on `noVaccineAvailable` until there are some vaccines in `vaccinesQueue`
5. When it gets some vaccines it returns them to the Zone with the company and pSuccess
6. Now Zone will call `addToSlotsQueue` which will add slots for the zone in `slotsQueue`
7. `addToSlotsQueue` uses the `slotQueueLock`
8. After putting up slots the Zone will `sleep` for some time so that students can register
9. Then the Zone will take a note of registered students and zone will remove its slots to prepare for vaccination phase. If no students registered control goes to `step 6`
10. Zone will vaccinate the registered students in vaccination zone
11. Then it signals students waiting on `notVaccinated` so they can go for antibody test
12. The vaccination phase ends and if the leftover vaccines zero control goes to `step 3` or else it goes to `step 6`

Students

1. They enter collegeGate at random time and `sleep` for random time before getting ready for vaccination
2. They call `getRegistered` to get registered to a Zone for vaccination

3. If there are no zones with slots it waits on `noSlotsAvailable` CV. Note multiple students get registered to different zones simultaneously.
4. After getting registered student waits on `notVaccinated` CV until the Zone vaccinates them
5. Then they are tested for antibodies according to the `pSuccess` of the vaccination
6. If they have antibodies student thread `exit`
7. If they dont have antibodies they can get registered again in `Step 2` upto maximum 3 times in total.

Main

1. It takes input
2. Creates all the students, companies and zones threads and gives each of them ID in ascending order
3. Waits for student threads to exit and then cancels all other threads.