User requirements:

* Users can share their location
* Users can specify which specific class they are taking attendance for
* Users can only take attendance within 30 mins from the beginning of the class
* Users can see the status of each seat in real time (available, considering, blocked)
* Users can only select in available status
* Any time a user click on an available seat, all the others in that room session need to be updated
* Users cannot change the attendance submission of other users
* Users are allowed to preview all the information before submission
* Users can change their attendance information above if theirs information are incorrect
* The selected seat need to be blocked from the others until submission

System requirement:

* The functionality of our system is to mark and record attendance in our database with the use of geolocation services
* System is required to save the current state of the seat
* The temporary cached database is synchronized with the real-time database
* Services: Geolocation Api, My SQL database
* Operational constraints: server crash, API overloading, seat updating error

Risk Analysis:

Real Time performance : Testing, Simulation

* Redis and/or MySQL, Frontend, Backend servers are crashed

=> possible solution: Terminate all the current online student, inform them to retake the attendance when server are available

Developing Wrong User Interface: Prototyping, scenarios, task scenarios

+creating a confusing UI

=> Solution: having a user/customer review it and give feedback to make necessary changes

* Database overloaded:

=>Solutions: Delete obsolete attendance information

Allocate more storages (caching and temporary) serving for the projects