I. Operating System Requirements

A] The "Elementary School Library Application -2019", developed by "Trinity Internet & Database", shall be a windows form based application written in Visual Studio 2017 the application shall run on a Windows System (Windows 2005/2008/2012/2017 Server, and/or Windows 7/8/10). Stakeholders will receive all source code upon successful project acceptance and payment in full; all rights to share, or sell, this software solution is the sole property of "Trinity Internet & Database"

II. Database Requirements

A] The "Elementary School Library Application -2019" shall use a Microsoft SQL Server 20016 R2 Database that resides on a local area network 2012/2016/2017 server (at least 16 GB of RAM); purchase of the server, and all necessary support software, shall be the responsibility of the stakeholder. The database shall be called "Library". Backup and database management will be the responsibility of the stakeholder. "Trinity Internet & Database" will create, and install, the initial "Library" database; classes and indexes will be designed to optimize performance.

III. Data Size Limitations

A] The physical limitations shall be disk related, all speed contingencies will be based on upper limits of 1,000,000 Users, 10,000,000 Inventory Items, 10,000 Vendors, 100,000 Purchase Orders, and 100,000,000 Check-Out/Sales Transactions on the local database server.

IV. User Sub-System Performance Requirements

#1 The " Elementary School Library Application -2019" will provide the addition of four types of users (Guests, Students, Teachers, and Administrators).

- The user will find options to "Add-Administrator", "Add-Teacher", "Add-New-Student", & "Add Guest" on the main system menu-strip control.
- Once the user has launched any of the four Add form ("Add-Administrator", "Add-Teacher", "Add-New-Student", & "Add Guest"), they must either Save or Cancel to exit the form.
- Guests and Students will be able to add a new user account from the web site or at one of the local computers in the library.
- Only those users at the administrator security level will be able to add a teacher or add an administrator.
- Once data enters the data on one of the Add forms and chooses to save the record, the average database insertion time will be no more than .5 seconds.

- The Add forms will contain the data items agreed upon when the stakeholders signed the Class Diagrams.
- The Add forms will contain that functionality agreed upon when the stakeholders signed the Architecture Diagrams.

---------- Validation Recommendations I Would Make To Programming Team ----------

Create A Demo Database, using program code, which contains 1,000,000 dummy user accounts.

For validation, I would recommend a batch script which

- ⇨ (1) reads 1,000 new user records (including some normal users, teachers, administrators) from a file into memory,
- ⇨ (2) starts a clock,
- ⇨ (3) uses a loop to add these 1,000 internal memory records to the database using the same add function (as the windows form)
- ⇨ (4) stops the clock,
- ⇨ (5) uses a loop to read each of the 100 records from the database and validate that all fields are correct, and then deletes those sample records.

The TotalTime/Average will provide us with an Average Time to add a record to 1,000,000 users. I will make sure that there is a safety buffer to account to fragmented files, etc.

When using an agile approach, this type of validation process can be executed in the daily verification builds in an automatic process. The time requirement (.5 sec or less) is easily going to be verified and the stakeholder will not be able to say it is "too slow".

It might be that the data fed into the Add-User function comes from a "Windows Form", a "Web Page", or "Records" from some file. The "Windows Form", & "Web Page" would have to be manually tested, but I would not include this in the daily build.

---------- Comment ----------

By having the user

- ⇨ sign off on the architecture diagrams,
- ⇨ sign off on the class diagrams, and
- ⇨ sign off on the prototype

we get a "safety net" by which to renegotiate pricing if the stakeholder opts to include additional User information.