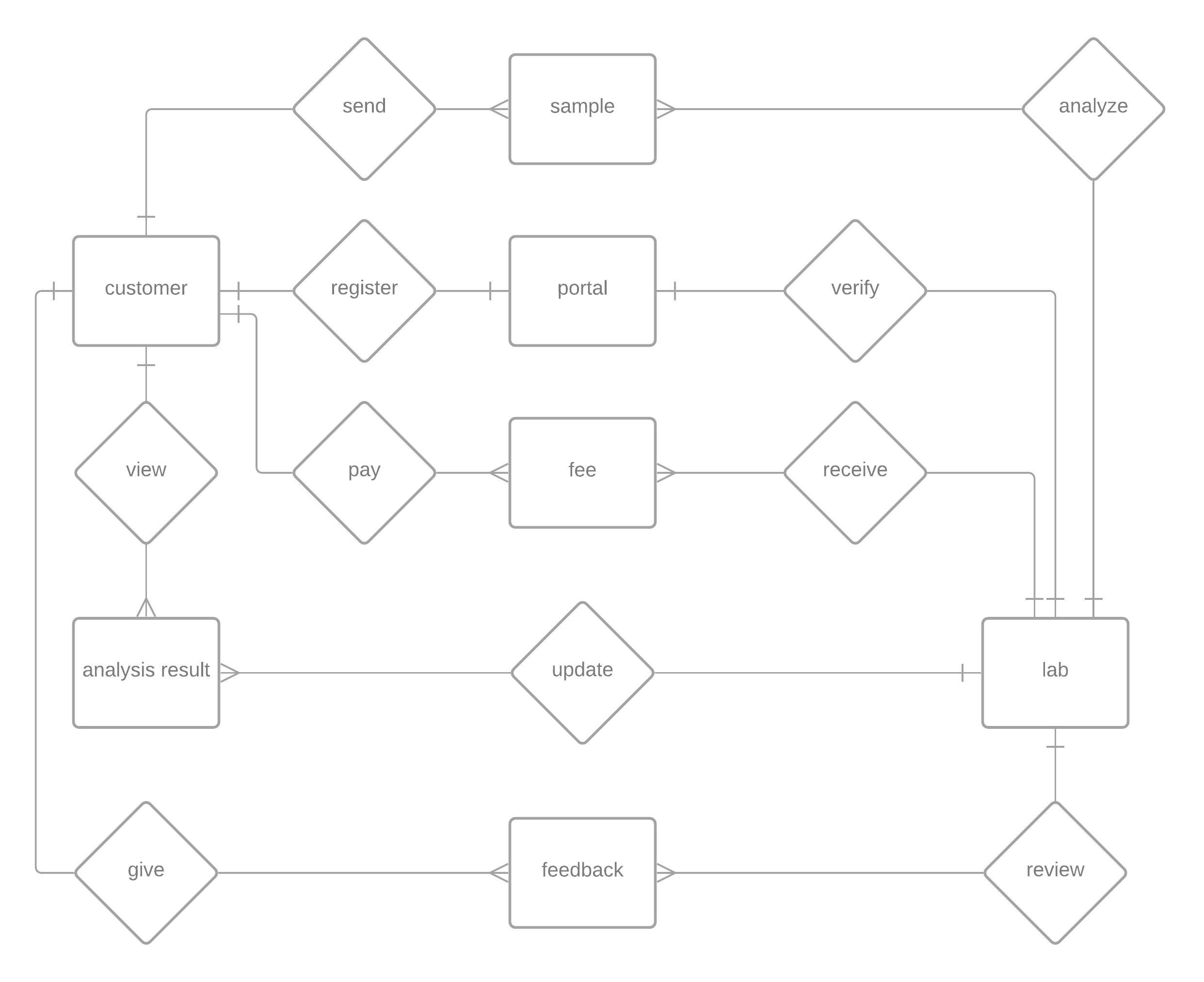
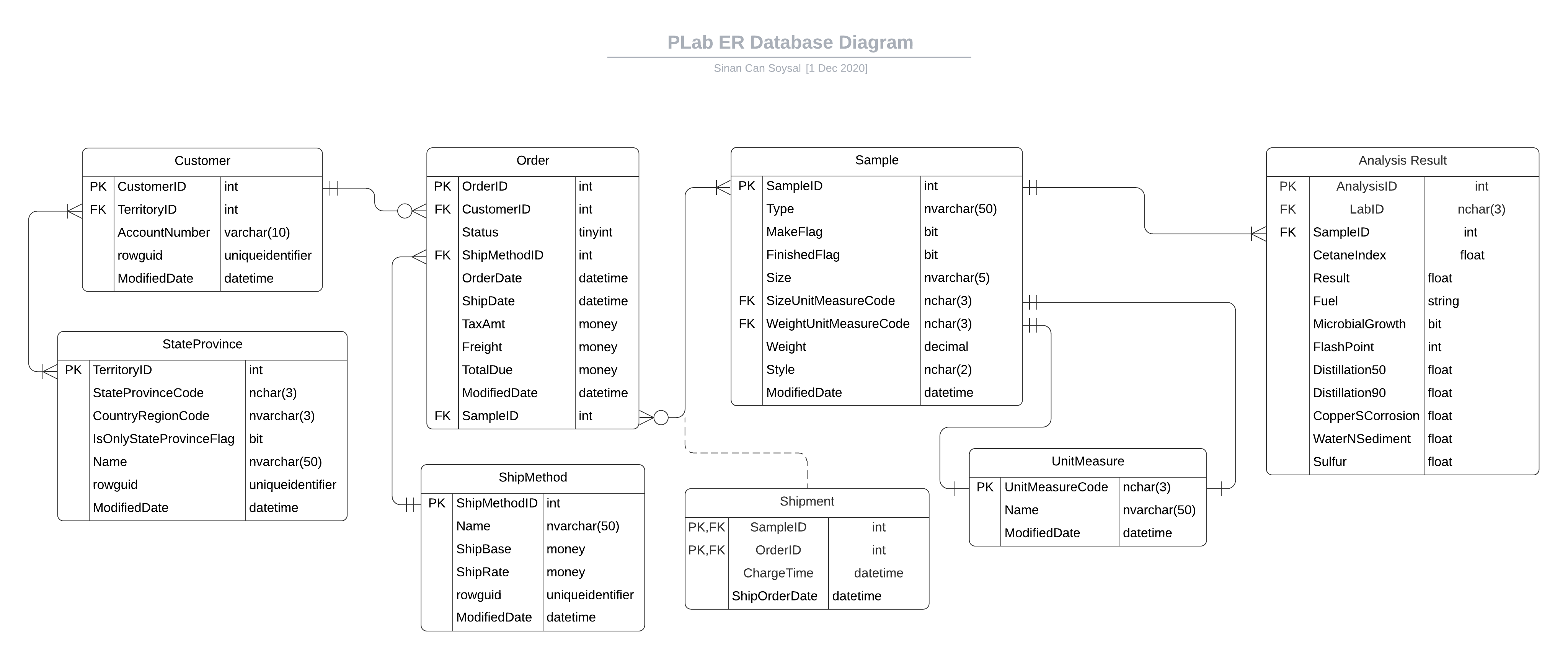
### Entity relationship data model (ER Diagram)



* Customer can register to portal
* Customer can pay for analysis
* Customer can send samples for analysis
* Customer can send feedback
* Customer can view analysis results
* Laboratory can verify/deny register request
* Laboratory can receive payment
* Laboratory can receive samples
* Laboratory can review feedback
* Laboratory can analyze samples
* Laboratory can update analysis results

### Database design model (ERD)



### Architecture design of the system

This portal will provide a bridge between customer and PLab Laboratories. The client must be able to see the results of a sent sample easily. Client logs into the PLab Portal in order to reach analysis results. One client can have multiple samples sent at the same time and the system must be able to handle all of them without malfunctioning. Laboratories across the country can update analyzed sample results throughout this portal. After analysis result uploaded, client can view it’s analysis reports visually.

There are 3 main packages of this comprised system:

* User Interface
* Data Storage
* Business Services

User interface will provide user interaction to the system. User interface must be able to support login, create sample requests, generate graphical representation of an analysis result, view results and send feedback to the system.

Data storage must provide reliable and secure data storage for the system. Client details, sample details, shipment details, fees, analysis results must be safely stored in the data storage. Business management and clients can reach needed data at any time from anywhere.

Business services contains the details of billing system, reading/updating data from database and managing client-business interactions.

Proposed system shall integrate with existing legacy system.

Our application works with large amounts of data provided by the labs around the country. In this situation, data will be accessed frequently by laboratories and clients. In order to accomplish this we must architect our application as a data-oriented system. The main reason of this architecture is to achieve integrality of data.

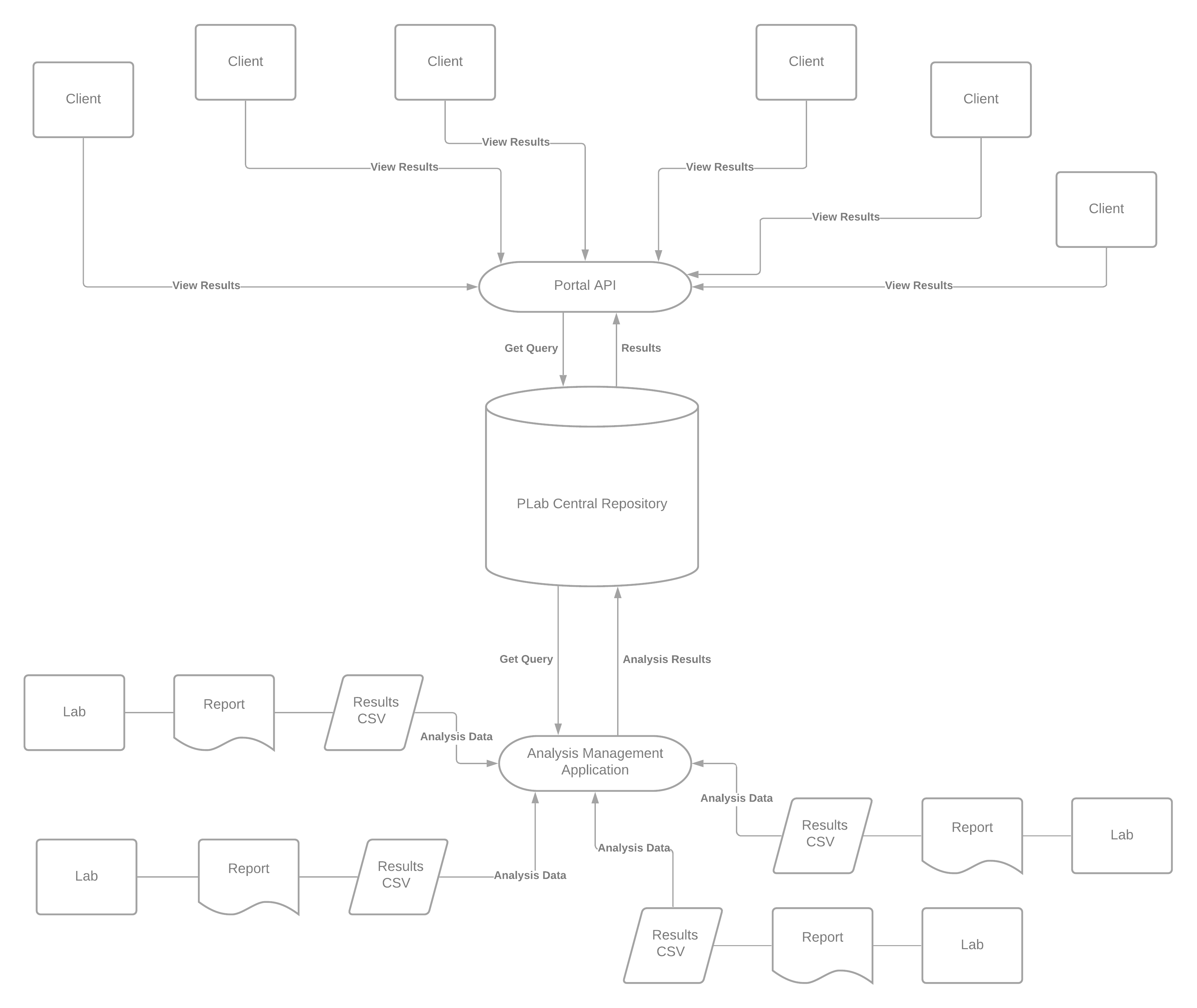
A centralized data repository will store all of the analysis results and it will be accessed through client side and business side API s. API’s will create queries according to the user’s needs.

Advantages of this architecture:

* Data integrity
* Backup and restore features ( this is the most important feature because our clients’ must reach their data whenever they want and wherever they want)
* Decreases overflow of data between software components

Disadvantages:

* The system is open to malfunction
* Malfunctions will highly affect the clients



The system shall support up to 2000 simultaneous connections to central repository at any given time.

The system shall provide users to view older results with no more than 5 second latency.

The system must finish its transactions within 2 minutes.

According to these and predefined requirements document:

The selected architecture design must support the sizing and timing requirements through client-server architecture. The client part is implemented on PCs and ensured to use minimal disk and memory.

The architecture supports quality requirements:

* Linux, Windows, Mac OS support
* User interface shall be designed for easy use
* System must be available 24/7

### Software and hardware specifications

To successfully accomplish the requirements of the project, some modern technologies have been suggested for the better service and security by having an online portal an an online sample registration system. This document contains all the software and hardware technology that has been proposed for the new system.

Software Specifications:

Application will be controlled by:

* Python & ZingChart as front-end solution
* PostgreSQL as database solution
* GoLang as backend solution

Python is a rich programming language in terms of it’s data visualization libraries. User interface must be beautiful and simple at the same; ZingChart comes into play for the visuals of the portal. ZingChart is a free and powerful python library specifically developed for data visualization.

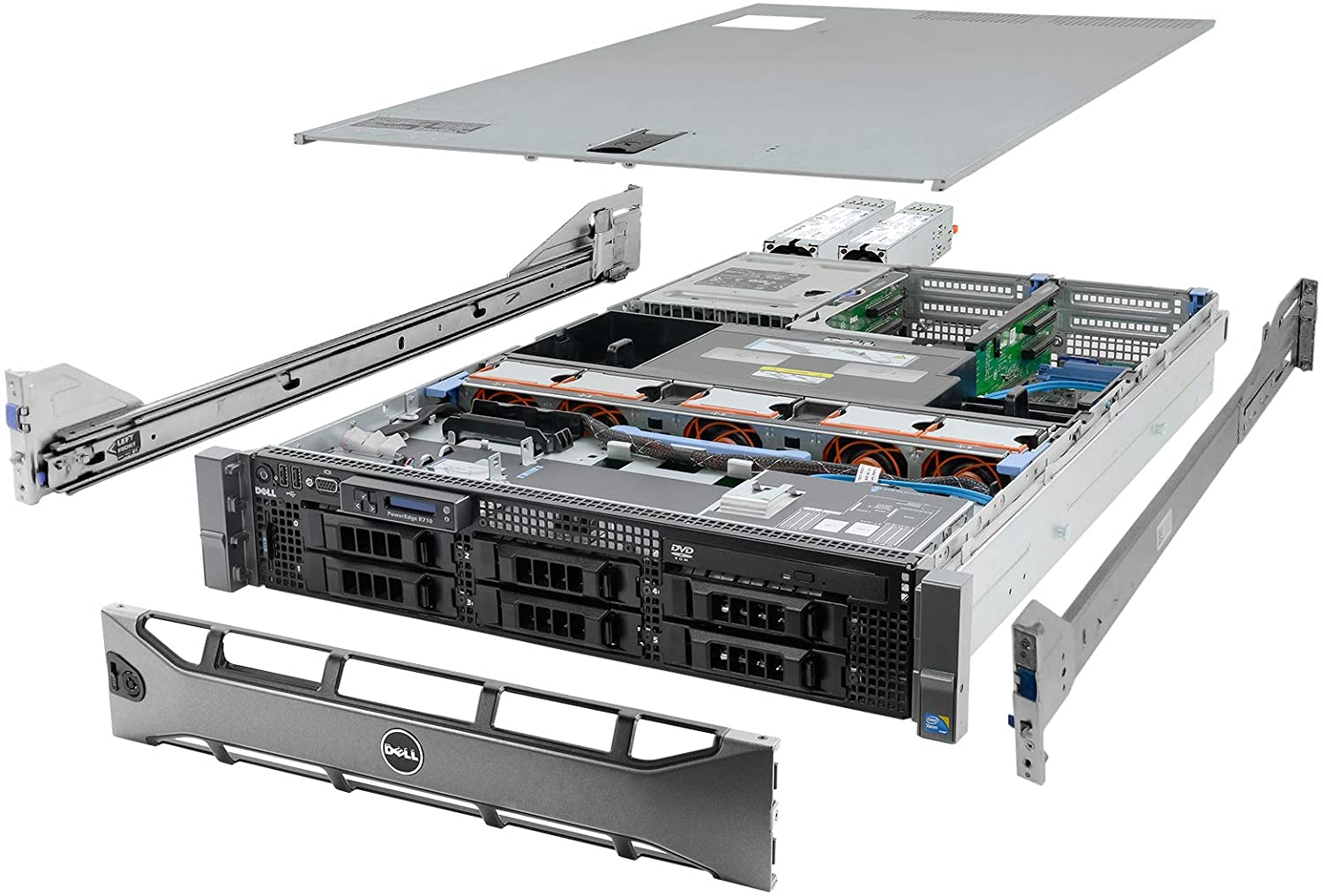
PostgreSQL is a free and open-source relational database which will get it’s job done in a fast and secure way.

GoLang is a programming language designed by Google. Go routines are GoLang’s most important future so far which allows concurrency in web based solutions.

Hardware Specifications:

**Dell PowerEdge R710 6B LFF Server**

R710 will be the main server for “PLab Analysis Laboratories” system. R710 will help us to operate efficiently and lower TCO with enhanced virtualization capabilities, improved energy efficiency, and innovative system management tools. It provides railings for easy installation and removing operations. Also it supports RAID operation with 12TB of storage. It’s CPU donated with 12-core Xeon processor and 12Mb of cache. The main unit manufactured by Dell, and the case is designed to allow accessing to hard disk drives easily. The server has got 2 power supply units for continuous working even if one of the power supply mulfunctions.



|  |  |
| --- | --- |
| Hardware Specification: | Price |
| System: Dell PowerEdge R710 6B LFF Server   * **Processors:** 2x 2.83GHz X5670 12-Cores Total * **Memory:** 144GB RAM * **Hard Drives:** 6x 2TB 3.5" HDD * **RAID:** H700 w/ 512MB * **Optical Drive:** DVD-ROM * **Power Supplies:** 2x PSU * **Bezel:** Yes * **Rails:** Yes * **Operating System:** None | $ 729.99 |

**Additional Hardware:**

Every laboratory has its own central computer so, we won’t be buying new computing hardware to the existing laboratories instead we will deploy a new software to existing computers. A communication link will be needed between all the laboratories and the PLab’s central repository. The link will be provided by our existing ISP and the whole communication structure will be called as ‘Chiral Network’.