Aggregate Communication Platform for Project Management

Pranav Kelkar[1] Nitish Kulkarni[2] Pranav Risbud[3]
[1,2,3]Department of Computer Engineering
[1,2,3]MMCOE, Pune

Abstract— In commercial industries, projects are time sensitive. Performance monitoring is an important aspect in case of such type of projects. The performance of an employee can be supervised and can be measured accordingly using a feasible and serviceable performance management system. This results in an improvement in the outcome of a particular project. The primary motive of this project is the transfer of information throughout a hierarchical structure of an organization. The main feature provided in this project enables the employees of an organization to update information regarding their work progress by means of a web application as well as an android device.

Key words: MySQL database, MongoDB

I. INTRODUCTION

The project in implemented using the client-server architecture and HTTP protocol. The server side of the application is implemented using GoLang- The Go Programming Language. The client side of the project consists of two parts- A web application and an Android application. The go server is coded using the standard gin framework for GoLang, which is a JSON-API framework. The server has access to two different databases, namely mongoDB and MySQL database. MongoDB database is used to handle transient data while MySQL database is used to handle non-transient data. Gin provides you an increase in performance and scalability. In addition to this, Gin framework provides support for multiple databases in your application, and a client to create, start and stop your applications. The client side is programmed using JavaScript, HTML, CSS and java for web and android applications respectively. The access to the web client is given to the employees working in the higher level of the hierarchy, while the android client can be used by lower level employees for data communication.

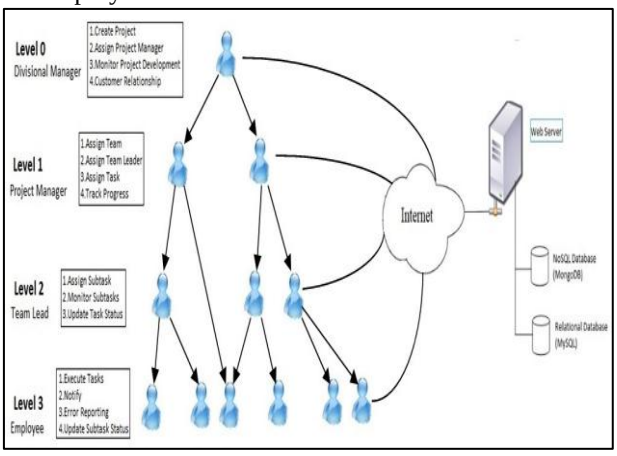


Fig. 1: The Hierarchical Model

II. SYSTEM IMPLEMENTATION

The system implements the client-server architecture which consists of a server implemented in GoLang, the GO programming language, and two types of clients, a web client and an android client. The implementation of these components is explained further in this paper.

A. Web Client Implementation:

The web client is used to supervise and oversee users, projects and their respective co-relations. The access to the web client is granted to the concerned higher level authorities of the company such as an administrator or manager. The administrator, through the web client, creates users, contracts, projects and establishes and edits their inter-dependencies. The web client synchronizes the data to be used for the android device clients. The web client is implemented using HTML with JavaScript. The web client communicates with the server using "POST" method and is accepted at the server side.

1) Administrator:

Once the administrator logs in with his user id and password, he can perform following tasks:

- Create new contract
- Edit contract
- Edit employee details

2) Owner:

The owner can perform the following tasks once he logs in:

- Create new project
- Create new user

3) Manager:

- Create new task
- Assign task to user

B. Android Client Implementation:

As the proposed system shows, android device is used mainly for the lower level Employee end.

Employees actually working (programming, coding) on various assigned project tasks can report the status of the same very efficiently with an android application installed on the mobile device which in turn reduces the latency as that of the conventional systems.

1) Steps in implementation:

- Login :
- Employee logs in with user_id and password defined by the respective corporation.
- Explore list of assigned tasks:
- Server keeps the record of assigned tasks for each and every employee. As the employee logs in , receives a list of tasks to be completed, from server under a particular project.

2) Implement Task handler:

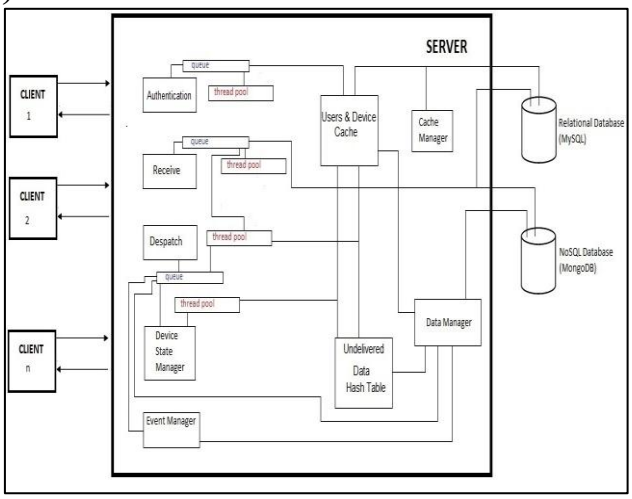
A task handler activity is provided in android application in which an employee can briefly get access to all the details related to a specific task such as, Task Description, Task id, task deadline, etc. Along with that an employee is given an interface by which, he/she can report the status of the corresponding task. The reported status automatically gets updated at server end and thus the higher level employees are informed.

3) Messaging Plug-in:

Android application is built with a messaging plug-in in case an employee needs to interact with another employee working under the same project regarding any issues in task handling.

C. Server Implementation:

1) Server Architecture:



2) Overview:

1) The server will have requests from web client as Well as android application (i.e. mobile client)--referred to as device.
2) All requests to the server will be dumped initially in a MAIN queue.
3) Both web and android devices are referred to as clients
4) It mainly contains of following modules:-
   - Authentication Manager
   - Cache Manager
   - Data Receiver
   - Data Dispatcher
   - Device State Manager
   - Data Manager
   - Event Manager

3) Databases:

- MySQL
- MongoDB

A request manager will check the type of the request and dequeue it from MAIN queue to put it into either

- Authentication queue.
- Data queue.

D. Authentication Manager:

Authentication Manager is responsible for logging of clients. It has its own thread worker thread pool. It will dequeue the request delivered to its queue and check in the user & device cache. The user & device cache is maintained so that the database is not accessed for every request causing latency. According to the result it will dispatch the output.

E. Cache Manager:

The cache manager maintains cache coherency.
Cache manager will update the user & device cache every time there is an update on database thus maintaining coherency.

The cache manager will also be responsible for flooding user & device cache on server start.

F. Data Receiver:

- The data requests are dumped in the data queue from the main queue after request type check.
- It has its own thread worker thread pool.
- The worker threads dequeue the request and dump it into the database.
- They also check the device state manager if the client to which the request is destined in legitimate and registered.
- If so the result is directly dumped in the dispatch queue for delivery.
- If not, the result is stored in the undelivered data hash table till the device is available for delivery.

G. Data Dispatcher:

- The data dispatcher is responsible for delivering the results in its queue.
- Data Dispatcher too has its own thread worker thread pool.
- It checks with the device state manager for availability of client and delivers the data accordingly.
- If client is not available then, it dumps data into undelivered data hash table for future delivery.

H. Device State Manager:

- Device State Manager is responsible to maintain the state of client.
- It is updated on every time a client pings the server.
- Device State Manager has its own thread worker thread pool.
- It acts in accordance with data manager and delivers data from undelivered data hash table when client is available.

I. Data Manager:

Data Manager does the job of delivering of data from the database as well as undelivered data hash table.
As said earlier it works in accordance with the device state manager.

It also cleans the undelivered data hash table from stale data by dumping the data into database.

J. Event Manager:

The event manager works with the data manager.
It scans the databases for deadlines, alarms and events and delivers data to concerned client.

K. Databases:

1) MySQL:

1) The MySQL database is a relational database.
2) It contains non-ephemeral data and but less-scalable and vital data.
3) It has the following main tables supplemented by other support tables-

- Owner Table
- User Table
- Corp Table
- Task Table

2) MongoDB:

1) MongoDB is a no-SQL database.
2) MYSQL database has limits for data storage.
3) It is difficult to scale with amount of data growing over time.
4) It contains less trivial data but which is going to amass.
5) Data is maintained in collections and documents in MongoDB.

III. FUTURE SCOPE

This application can be deployed in variety of organisations working in different domains. The communication between employees of a commercial organization becomes easy due to user friendly android application. It enables authorisation and assignment of tasks as required for different workgroups and individuals. It is easy to configure and deploy. It is web and mobile compatible. It is widely applicable to all industries.

IV. CONCLUSION

The Fixed Asset Reconciliation using Android Application is meant to simply and accelerates the process of asset reconciliation. Android Smartphone is easy to handle and use. Based on the data collected during reconciliation process various reports are generated which helps in the analysis of the company's financial status.

REFERENCES

[1] Zhi-bing Wang, Zhu-mei Sun, "The Realization of Web Service Description of Project Procurement Management Based on Semantic", June 2013.
[2] Sunil Patel, Brad Levin, Robert J. Gac, Douglas Harding, Anna K. Chacko, Ronald Wider, John Romlein, "Picture archiving and communication systems project management using web-based tools", May 2000.
[3] AtifSaeed, Muhamamd Shahid Bhatti, Muhammad Ajmal, Adil Waseem, Arsalan Akbar, Adnan Mahmood, "Android, GIS and Web Base Project, Emergency Management System (EMS) Which Overcomes Quick Emergency Response Challenges", 2013.

