**Science Journal of Circuits, Systems and Signal Processing**
2020; 9(1): 16-23
http://www.sciencepublishinggroup.com/j/cssp
doi: 10.11648/j.cssp.20200901.12
ISSN: 2326-9065 (Print); ISSN: 2326-9073 (Online)

# Development of a Model Architecture for Job Scheduling

**Eduok Uyuho Isaac, Amannah Constance Izuchukwu**

Department of Computer Science, Ignatius Ajuru University of Education, Port Harcourt Rivers State, Nigeria

**Email address:**
uyeduok@gmail.com (E. U. Isaac), aftermymsc@yahoo.com (A. C. Izuchukwu)

**To cite this article:**
Eduok Uyuho Isaac, Amannah Constance Izuchukwu. Development of a Model Architecture for Job Scheduling. *Science Journal of Circuits, Systems and Signal Processing*. Special Issue: *Circuits, Systems, and Signal Processing*. Vol. 9, No. 1, 2020, pp. 16-23.
doi: 10.11648/j.cssp.20200901.12

**Received**: March 29, 2020; **Accepted**: April 22, 2020; **Published**: May 18, 2020

**Abstract:** Job scheduling has a long history. Job schedulers have been one of the major component of IT infrastructure since the early mainframe system, at first, stacks of punched cards were processed one after the other, hence the term batch processing. The aim of a job scheduler was to arrange, control and optimize work and workload in a production process. The study aimed at achieving a job scheduler that can assign jobs to available resources in such a way that workload leveling can be achieved. The objectives of the study included to; ascertain the existing schedule system, design a model platform for job scheduling, implement the model platform for job scheduling, and test and deploy the model platform for job scheduling. The study adopted the spiral development model. The proposed system was developed using java script which is a client scripting language which is used for creating web and windows application and it has almost all online application since the system is an online system. The developed application was tested with field Meta data and the outcome was according to specification and of desired output. The application was hosted on three different servers, one server for the frontend, another server for the backend and the third server for the database, hosting on the three different servers did the magic by making the system faster and easy to optimize.

**Keywords:** Model, Architecture, Job Scheduling, Check off Jobs, Scheduling System, Labor Load Levelling, Optimization

## 1. Introduction

Statement of the Problem

The pen-and-paper method is the old way of job scheduling for a workload, in which all job scheduling is funneled through an operation manager or the business owner. In this model, the personnel in charge manually write down the available jobs and available staff / machines to carry out the jobs, with this method labor load leveling becomes so hard to achieve, and thereby leads to chaos in a situation where workers are paid according to their workload. With large workgroups or very busy schedules, this system can rapidly become untenable, record keepings becomes difficult, as books use to manually record the scheduling get filled up, and there is no real time information and can also have negative effect on productivity. This project will design an automated job scheduler that will carry out repetitive tasks as required in a schedule-based calendar and event conditions. There is no reason to manually check off jobs in a log as they finish, the entire process will be automated thereby producing real time information, labor load leveling will be achieved, time wasted in the manual process will be reduce thereby increasing productivity and efficiency of the organization.

Purpose of the Study

The main aim of the study was to develop a model architecture for job scheduling system. The objectives of the study include the following.

1. To ascertain the existing schedule system.
2. Design a model platform for job scheduling.
3. Implement the model platform for job scheduling.
4. Test and deploy the model platform for job scheduling.

Scope of the Study

The study is limited to a job scheduler that is automated, a job scheduler that will be arranging, controlling and optimizing work and workloads of IKODOTENE solutions, achieving labor load leveling, producing real time information and increasing efficiency of the organization. This work will develop a tool that is limited to generating schedules and assigning shifts based on employee's qualifications, availability and a variety of other customizable parameters.

## 2. Related Literature

Job Scheduling Architecture

Job scheduling architecture is the operations' architecture and governance that is responsible for the standards and frameworks to support a unified, centrally administered job scheduling capability across all computing platforms within a data center, Wicken [1].

Multiple Resource Theory (MRT)

Wicken's MRT in proposes that the human operator does not have one single information [2]. Applying Wicken's theory help to identify and assign jobs that can be performed concurrently to same or different resources, and task that would interfere with each other are assigned separately and are given priority to avoid interference. Task that other tasks depend on to be executed are assign to the needed resource and are given high priority to avoid loss of performance of other tasks that depend on it.

In 1903 Gantt, Gantt proposes that the main record shows what each worker should do and did do, and the daily balance shows the amount of work to be done and the amount that is done, [3]. In modern job scheduling software this is known as the 'management information' this information is use for management decision making, this information include what each worker should do and what was done by each worker, it also shows the daily or weekly or monthly (depending on the system design) jobs to be executed and the amount that is executed.

In 1916 Gantt, Gantt also proposes that each day a list of jobs to be executed be given to the foreman, [4]. A job scheduling system can only function efficiently when the jobs to be executed are made available to the system on a daily or weekly basis (depending on the system design and nature of jobs) for efficient scheduling.

Work front

Silicon slopes in 2019, Designed a modern work management application platform (Work front), the main objective of this design was to help people do their best work, so companies can thrive in a digital world [5].

Work front was designed for people, effortlessly connecting teams and easily integrating into existing applications and systems, it was implemented using PHP and MySQL, with one platform to set goals, track goals, manage work, track progress, and measure results. When work front was implemented, it has the ability to track progress, integration into existing system has helped thousands of companies successfully transform their businesses into modern enterprises that increase revenue, improve customer experiences, and reduce cost. It was develop using a lightweight programming language, it is easy to access via web by the workers to check their schedule. Work front help organizations solve the essential challenge of large enterprises aligning people and teams to effectively and efficiently achieve company objectives, [6, 7].

People do their best work when they know their role, have clear goals, and understand how the work they are doing supports company objectives," one of the developers of work front [8]. The limitation of work front was that jobs were just assigned to each worker without specifying the deadline for each job. This will be addressed in the current study. The current study will provide a user interface, the interface will help the user create jobs and specify deadline for each job created.

Open Shop Scheduling in a Manufacturing Company Using Machine Learning

Strand's work was limited to a manufacturing environment (scheduling task between machines). Hence did not address staff scheduling. The proposed system of the current study is designed to address this gap in knowledge which was left open in the work of Strand [9]. The current study will provide an interface in which workers of an enterprise can view their schedule, shift and jobs allocated to them.

Optimization Techniques for Task Allocation and Scheduling in Distributed Multi-Agent Operations

In Tompkins's work, jobs were only assigned to available agent at real time, agent break time, off duties were not captured by the system [10]. This will be addressed in the current study. The current system will give the human resources personnel the opportunity to assign jobs to available resources to be executed later (depending on the nature of the job), and can also assign jobs at real time.

Dynamic Resource Management and Job Scheduling for High Performance Computing

Prabhakaran's work in Strand [9], was limited to scheduling task in a computing environment, and did not handle enterprise workload scheduling. The proposed system of the current study is designed to address this gap in knowledge which was left open in the work of Prabhakaran in Strand [9]. The current system will handle enterprise workload scheduling, by assigning real life jobs to available staff of an enterprise, and ensure workload leveling between workers of an enterprise, [11].

## 3. System Analysis and Design

Analysis of the Existing System

The existing system is a manual job scheduling system. The Human Resources Personnel (HR) of IKODOTENE solution has been carrying out workload scheduling manually. It involve HR Personnel identifying the jobs to be executed for a new week over the weekend before the week commences, most times work changes as the week goes on, the personnel has to check every morning jobs to be executed and available resources to execute the jobs, if need be for rescheduling, he has to reschedule. And he has to check on daily basis to ensure workload leveling is achieved to some extent. The file containing this schedules are manually stored.

Limitation of the Existing System

The following are the bottleneck of the existing system;

1. It is stressful and tedious.
2. It is time wasting, thereby having negative effect on productivity.
3. Labor load leveling is hard to achieve.

4. No real time information, so therefore management decision can't be made at real time.

### Analysis of the Proposed System

The proposed system is an automated job scheduling system. This system automatically schedule available jobs among available resources. A software engine is responsible for scheduling jobs for available resources as against the manual process in the existing system. When available jobs are entered into the system, the system allocates these jobs dynamically to the available resources which are stored in the database of the software engine. The proposed system has interfaces which help to define workflows and or job dependencies. The system tracks each job and automatically submits executions.

### Justification of the Proposed System

The proposed system provides the following merits over the existing system:

1. An organized database of available resources.
2. Save time and improve productivity.
3. Produce Real time information.
4. Achieve labor load leveling.
5. Speed-up business processing.

### Method used in the Study

The method adopted for this study is the spiral development model. This method provides solution to problem in corkscrew approach such that one solution step leads to another until the overall solution is achieved.

### Architecture of the Proposed System

The architecture of the proposed system is made up of one interface, the graphical user interface (GUI). It is structured to coordinate schedule jobs, status and history, as well as control jobs. Consequently, it also offers components that enable the administrator to create new resources into the resources database, and also retrieve jobs assign to resources from the map tasks to resources database.

Below is the explanation for the various parts of the proposed system:

Job Scheduler: it maintains records of incoming jobs, queue status of jobs, details of processed and processing jobs and the next jobs scheduled for processing. It starts the task mapping agent process and feeds it the necessary information to retrieve available resources and assign it to the jobs, and also submit the mapping result to the map tasks to resources database via the Scheduler agent, it access data directly from the database.

Client GUI: this interface helps you do the following;

1. Load input data (new resources) into resources database.
2. Submit jobs.
3. Extract assigned jobs from the map tasks to resources database.

Resources database: All of the information about resources is stored here.

Task mapping agent: This agent is part of the job scheduler, it does the task to resource mapping.

Scheduler agent: this agent is also part of the job scheduler; it takes the job mapping from the task mapping agent and sends to the map tasks to resources database.

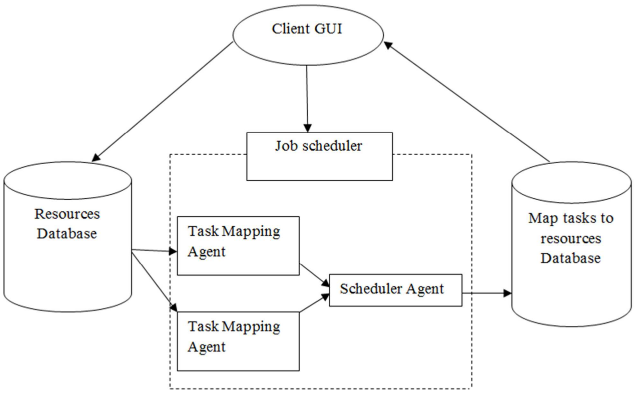


***Figure 1.*** *Proposed Architecture.*

### Input Model

The input design describes the process through which data filled in by the user is passed to the software engine for manipulation. A client input the jobs for scheduling with the aid of a keyboard and click on next to input another job till all jobs are inputted, and then click on submit button which triggers the process.

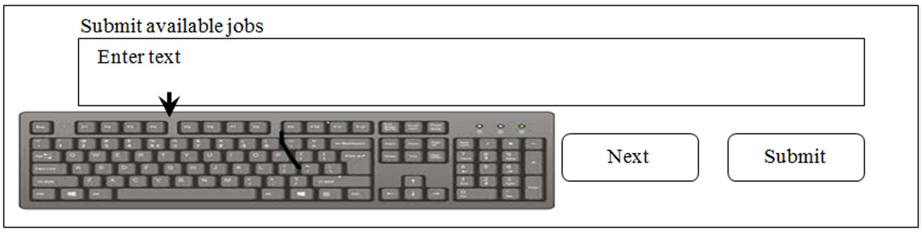


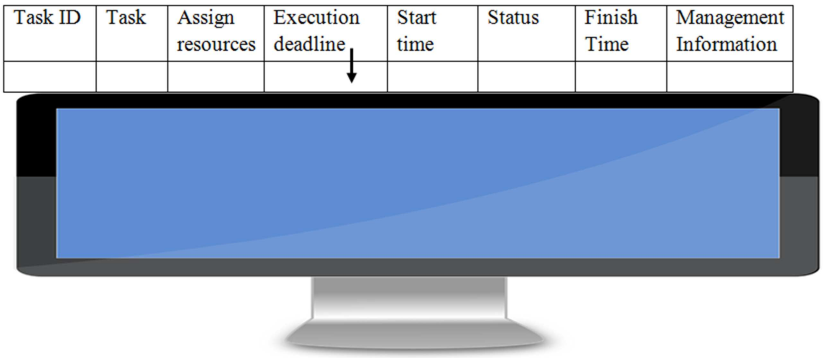
***Figure 2.*** *Input model template.*

### Process Model

The process begins with the processing of the various tasks entered. When the administrator enters the task, it is first stored and then the available resources are crosschecked to examine which one will best suit the job. The deadline of the task is examined and the speed at which the resource can execute the task is also examined if the resource cannot execute the task within the deadline, the task is assigned to another resource or resources that can meet up with the deadline. The process design is explained with a flowchart for comprehension of the steps.

### Output Model

The output model is dependent on the jobs inputted by the user and also the available resources. Once list of workflow is entered in the user interface, it is displayed on the screen and when the process is triggered upon clicking the submit button, the mapping of jobs to resources is displayed.



| Task ID | Task | Assign resources | Execution deadline | Start time | Status | Finish Time | Management Information |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

*Figure 3.* *Output Design.*

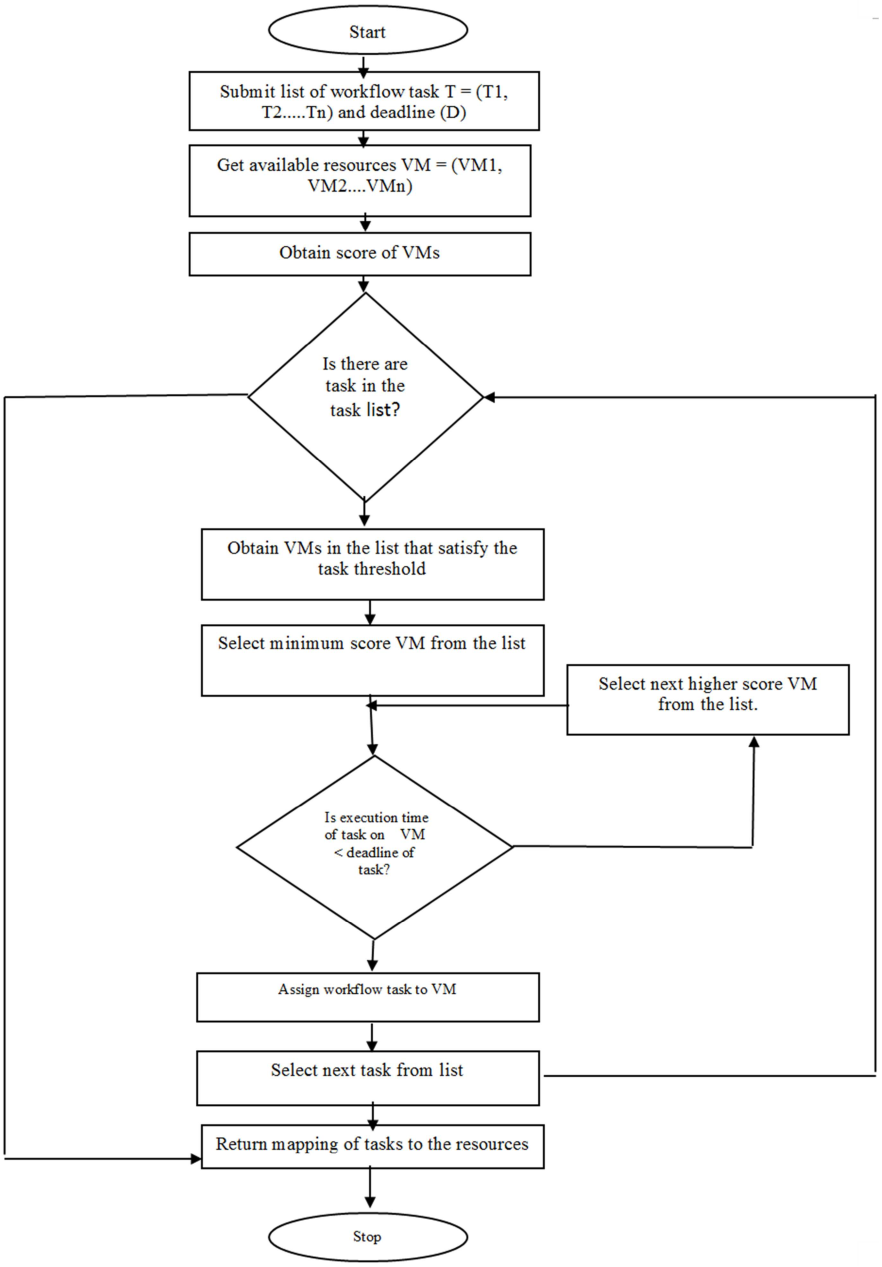

Start
Submit list of workflow task T = (T1, T2.....Tn) and deadline (D)
Get available resources VM = (VM1, VM2....VMn)
Obtain score of VMs
Is there are task in the task list?
Obtain VMs in the list that satisfy the task threshold
Select minimum score VM from the list
Select next higher score VM from the list.
Is execution time of task on VM < deadline of task?
Assign workflow task to VM
Select next task from list
Return mapping of tasks to the resources
Stop

*Figure 4.* *Program Flowchart.*

Database of the Proposed System

The proposed system database will help to read (view data via queries), create (add data, table, rows or columns), update (change data, table, rows or columns) and delete data. Conditions are added to perform such function which includes:

1. The field name: This is for alphanumeric data (all digits and printable characters).
2. The field type: Determines the type of data which the user can enter into the field (alphanumeric, numeric etc).
3. The field width: This is the percentage of the outer container.

***Table 1.*** *Database Design.*

| S/N | FIELD NAME | FIELD TYPE | FIELD WIDTH |
|---|---|---|---|
| 1 | Task description | Var Char | 25 |
| 2 | Assign resource | Var Char | 25 |
| 3 | Execution Deadline | DateTime | YYYY-MM-DD HH: MI: SS |
| 4 | Start time | DateTime | YYYY-MM-DD HH: MI: SS |
| 5 | Status | Char | 10 |
| 6 | Finish time | DateTime | YYYY-MM-DD HH: MI: SS |
| 7 | Remarks | Var Char | 25 |

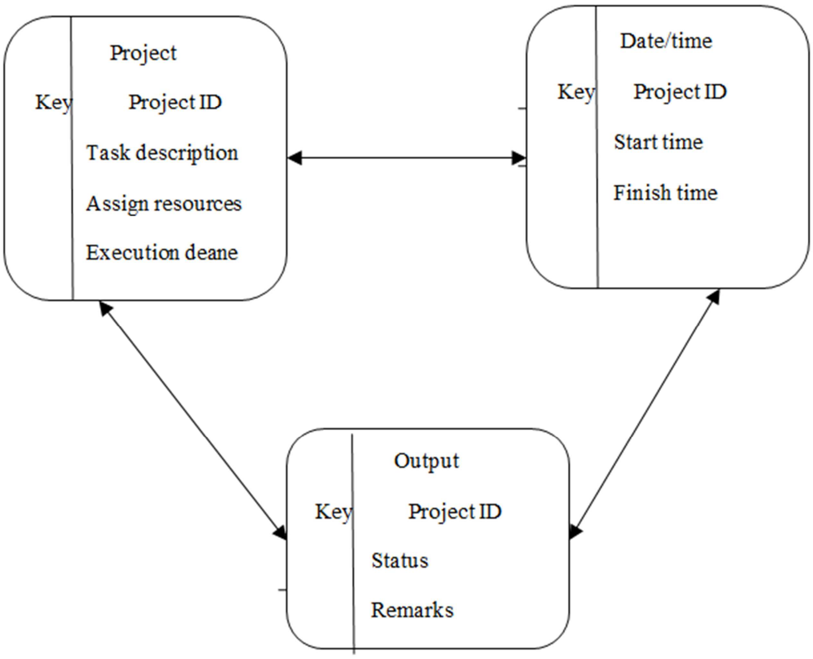


***Figure 5.*** *Logical Database Design.*

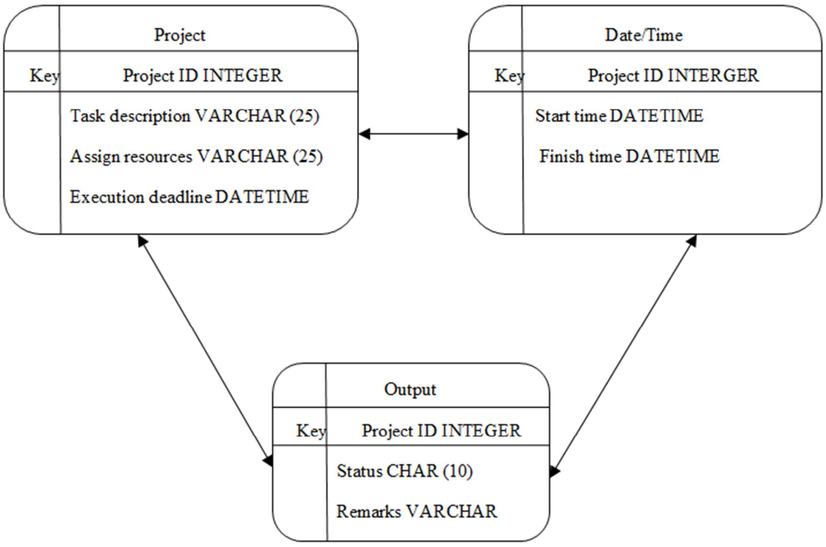


***Figure 6.*** *Physical Database Design of the Proposed System.*

## 4. System Implementation

Choice of Programming Language

The programming languages used in the implementation of this project are;

1. JavaScript.
2. HTML (Hypertext Mark-up Language).
3. MongoDB (NoSQL).

Justification of Programming Language

JavaScript is a client scripting language which is used for creating web and windows application.

Implementation Architecture

The implementation architecture provides for only one login form for users and admin. Once the first user of the system login, he can create other users, and he then becomes the admin. So there are two types of users which are; the admin and other users. Behind the scene when a user or admin login and the login is processing, the system checks if the user has the role of a user or an admin. If the role of the user returns admin, it takes the user to the admin panel, where he or she can input task and assign type of task, add new users, assign roles to user base on their areas of experience, view assign jobs and can also give a user admin privileges. But if the role of the user return user, it takes the user to the view job assignment page, were the user can view jobs assigned to him or her, as well as update the status of the job. When the admin input new jobs, add new users, add new admin, the data are stored in the database. When a user updates the status of his or her assigned job, it is stored in the database.

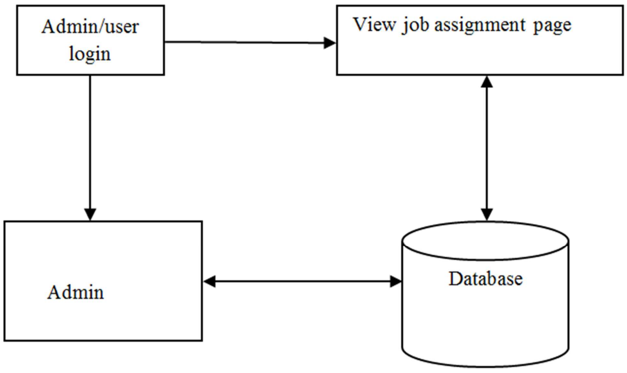


***Figure 7.*** *The Implementation Architecture.*

## 5. Summary and Conclusion

Summary

The purpose of this project was to develop a job scheduler that is automated, that will be arranging, controlling and optimizing work and workloads of an enterprise, achieving labor load leveling, producing real time information and increasing efficiency of the organization, the job scheduler achieve this by generating schedules and assigning shifts based on employee's qualifications, availability and a variety of other customizable parameters. The study adopted the spiral development model. This model encompasses the strengths of the waterfall model, while including risk analysis, risk management, support and manages processes.

The design of the proposed system was done using data flow diagram (DFD) and entity relation diagram (ERD). Data flow diagram (DFD) was used to show the flow of information in the system, which include data input and output, data stores and all the sub-processes the data move through. The proposed system was developed using java script. JavaScript is a client scripting language which is used for creating web and windows application and it has almost all online application since the system is an online system, then it becomes an ideal programming language that can be used to implement the proposed system effectively and the database was developed using Mongo db, which is an open-source document-oriented database. This work will be beneficial to the IKODOTENE solution's management board and other workers of the company who may come in contact with the system as well as their clients, and other related organizations.

Conclusion

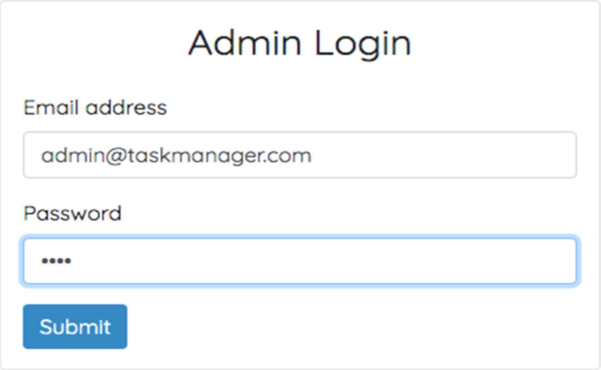
In this work, we have successfully developed an automated job scheduling system which automatically schedule available jobs among available resources; a software engine is responsible for scheduling jobs for available resources as against the manual process in the existing system. The application was hosted on three different servers to make it faster, one server for the frontend, another server for the backend and the third server for the database, for reason that if the application was hosted on one server, to access the frontend you also access the server, to access the backend you also access the server and so on which will eventually make the application slow. Hosting on three different server does the magic and easy to optimize if the code breaks, if it is the frontend code you just have to go to the front end server to fix it, same with backend code and the database. The workers of an enterprise can easily and conveniently view jobs assign to them at real time by logging into the system.

## 6. Recommendations

1. All enterprises should adopt an automated scheduling system.
2. Enterprises should encourage and train their staff to be computer literate, so that they can use the automated scheduler.
3. Enterprises that have adopted an automated scheduling system should ensure 24hours availability of internet.
4. System administrator and users of the automated system should have a good working relationship in order to provide accurate data for the system.

# Appendix

Sample Output



Admin Login

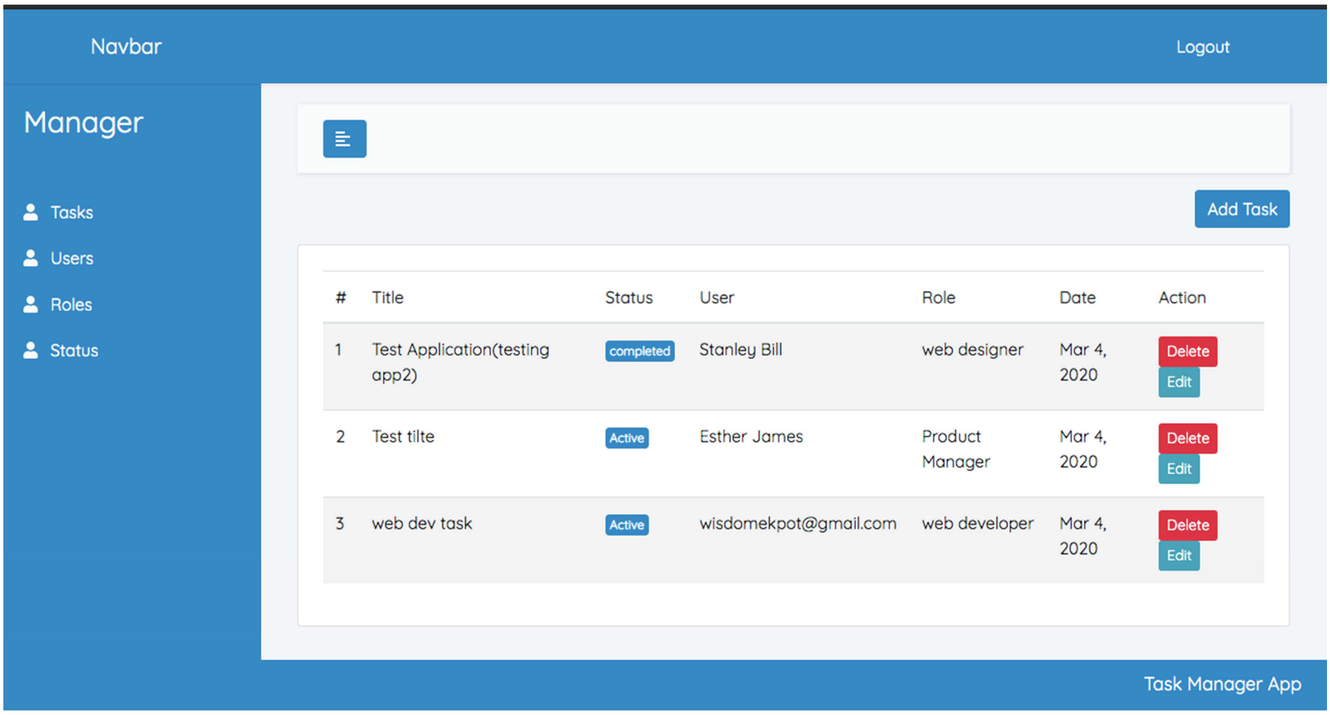
Email address

admin@taskmanager.com

Password

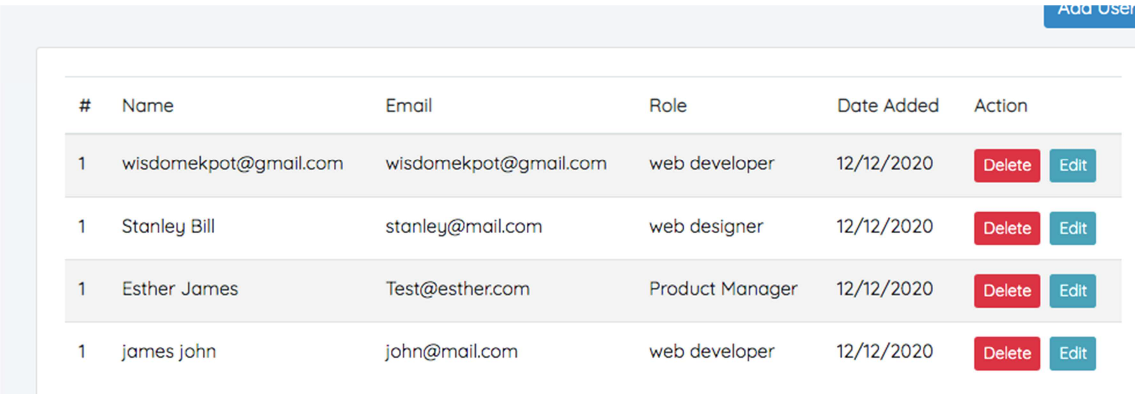
••••

***Figure 8.*** *Login profile.*



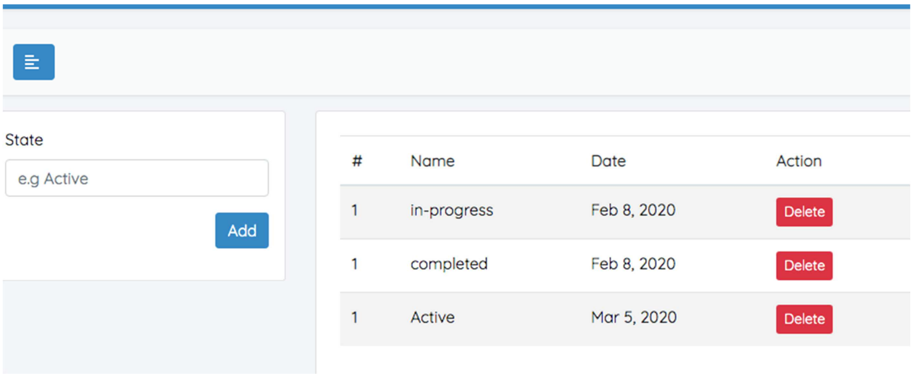
| # | Title | Status | User | Role | Date | Action |
|---|---|---|---|---|---|---|
| 1 | Test Application(testing app2) | completed | Stanley Bill | web designer | Mar 4, 2020 | Delete<br>Edit |
| 2 | Test tilte | Active | Esther James | Product Manager | Mar 4, 2020 | Delete<br>Edit |
| 3 | web dev task | Active | wisdomekpot@gmail.com | web developer | Mar 4, 2020 | Delete<br>Edit |

***Figure 9.*** *Jocllectionb.*



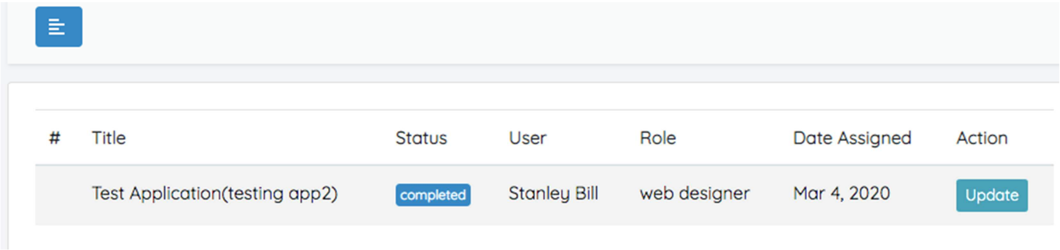
| # | Name | Email | Role | Date Added | Action |
|---|---|---|---|---|---|
| 1 | wisdomekpot@gmail.com | wisdomekpot@gmail.com | web developer | 12/12/2020 | Delete Edit |
| 1 | Stanley Bill | stanley@mail.com | web designer | 12/12/2020 | Delete Edit |
| 1 | Esther James | Test@esther.com | Product Manager | 12/12/2020 | Delete Edit |
| 1 | james john | john@mail.com | web developer | 12/12/2020 | Delete Edit |

***Figure 10.*** *Scheduled Jobs.*



| # | Name | Date | Action |
|---|---|---|---|
| 1 | in-progress | Feb 8, 2020 | Delete |
| 1 | completed | Feb 8, 2020 | Delete |
| 1 | Active | Mar 5, 2020 | Delete |

***Figure 11.*** *Executed jobs.*



| # | Title | Status | User | Role | Date Assigned | Action |
|---|---|---|---|---|---|---|
| | Test Application(testing app2) | completed | Stanley Bill | web designer | Mar 4, 2020 | Update |

***Figure 12.*** *Statistics of executed jobs.*

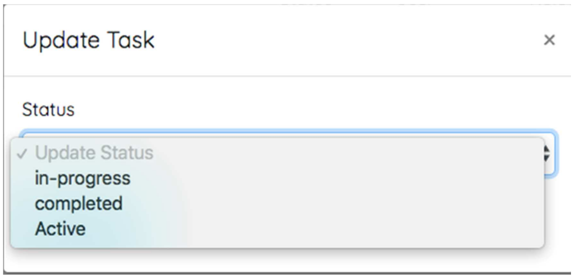


***Figure 13.*** *System update profile.*

## References

[1] Wickens, C. D. (2002). Structured design. *IBM Systems Journal, 13 (2), 115–139.*

[2] Wickens, C. D. (1984). Processing resources in attention, in R. Parasuraman & D. R. Davies (Eds.), varieties of attention. New York: Academic Press, 63–102.

[3] Gantt, H. L. (1903). A graphical daily balance in manufacture: Transactions of the American. Society of Mechanical Engineers, 24: 1322-1336.

[4] Gantt, H. L. (1916). Algorithms for solving production-scheduling problems, New Haven: Yale University Press.

[5] Pinedo, M. (2002). Scheduling theory algorithms and systems prentice hall, 1-6.

[6] Robert, V. (2005). Multiple resources and performance prediction. Theoretical issues in economics Science. *Journal of combinatorics 13 (6), 159–177.*

[7] Belenki, A. S., & Levner, E. V (2009). Scheduling models and methods in optimal freight transportation planning. *Journal of Automation and Remote Control, 23 (12), 1-56.*

[8] Lawler, E. L. (2019). Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Journal of mathematical science, 12 (9), 75–90.*

[9] Strand, H. H. (2016). Open shop scheduling for a manufacturing company using machine learning. *Journal of artificial intelligence, 10 (5), 123-346.*

[10] Tompkins, F. N. (2013). Optimization techniques for task allocation and scheduling in distributed multi-agent operations. *Journal of distributed agents, 15 (8), 14-15.*

[11] Dror, G. F., & Larry, R. (2007). Job scheduling strategies for parallel processing. Lecture notes in computer science, Michigan press, 17-64.