

# Leasing using Full Stack Web Application

Aravind R

Department of Information Technology , Sri Krishna College of Engineering and Technology,  
Coimbatore 641042, India

18euit007@skcet.ac.in@skcet.ac.in

**Abstract.** The creation of this web application arises from the need to facilitate and optimize the time spent in the entire management of rental apartments. It is becoming more and more common to acquire and rent properties. The administration of these can become a tangle of invoices, papers, appointments, and payments that worsens over time. The solutions that we can find today to this problem are basically two: delegate the process to a third party, which is an extra expense that many people are not willing to pay, or look for one of the few software that the market offers which are very complex and do not allow classifying and dividing the information of each apartment in a simple way and therefore do not solve the basis of the problem: the organization. So, why is this project innovative? The main feature of this application is the possibility of computerizing the entire process involved in leasing one or more assets and being able to manage them yourself without depending on a third party. You don't need hundreds of fields of information or calculators of net and gross profit as the few software's that exist in the market have. What inexperienced people who rent assets need is to be able to have their files, expenses controlled, and basic information organized. In summary, you will be able to save and classify the information of each of the leases by months, upload files, add notes, keep track of the expenses of each month along with the amount paid and owed, interact with your colleagues through general and specific forums for each asset, etc. and thus be able to access it when and where you need it in a fast and intuitive way.

**Keywords:** Full stack web application, Dot Net, Lease management

## 1 Introduction

Although nowadays it seems rare to find tasks that are done manually, there are certain jobs that small groups of people tend to carry out in this way. By the time we started to investigate and ask why they did not use any software or delegate this management to a third party. At this point, the project comes in handy to start the development of a web application managing rental assets. Web development is not only one of the fields that most fascinates me in computer science but developing something that is innovative and even something that can directly help financial companies. The Scope of this project is to make things easy for the leasing companies by making a software that handles every single Information about the leases, loans, assets, invoice details, sales tax and help to keep track. The goal of the project is to create and deliver transformative software solutions for leasing companies and remain committed to helping businesses meet the challenges of an ever-evolving leasing environment with the most advanced technology available. Integration of various modules developed for each real time entity according to the business functionality would manage the real-time leases. Primitive leasing system will be requiring more human efforts to document and manage the lease system. Computerizing lease system will require less human effort comparatively and management of them becomes handier in case of a software. Booking a lease for an asset in a primitive manner requires lot of levels or stages through which they approve or reject it. This workflow also can be computerized and integrated with software like pending, reviewing or approval step and can be integrated with specific module. Real time roles of an individual can also be computerized with limiting the accessibility of actions in the software. The leasing methods might change in future and the web application software needs to be changed accordingly to the new change. Advanced technologies might be helping financial companies like smart leasing etc. Dynamicity, reliability, and scalability were much needful for the software to manage nature of lease management. These properties can be managed by various technical architectures of which I chosen dot net architecture. As

this architecture possesses common language runtime, any programming languages can be integrated into the business logic layer of the web application. The need of different programming languages depends on the usage. For using machine learning analytics or smart leasing, python might be helpful. For performance in basic modules, C, C++, C# might be helpful. Dot Net generally accepts all programming languages and if integration of those technologies in lease management system will be boosting the productivity of the financing company. I chose C# programming logic for implementation of business logic in the application as it offers object-oriented supported and convenience for me on development. The nature of a data can be more changing more frequently. For example, exchange rate between currencies, tax rate in a country etc. How technically we can achieve this is via two ways either we should be having directly updating them to the situations or we should be integrating a third-party service which offer the latest data via API to the application. Both can be done through the business logic layer of the application. We use SQL database to store the captured data provided in the web application user interface or form. SQL stands for “Structured query language” which is more reliable and scalable. SQL server and dot net framework was backed up by Microsoft corporation private ltd. The common IDE used for development in Dot Net environment was Visual studio and for Microsoft SQL server, SQL server management studio. Issues might arise while several modules were developed and integration happening between them. So, whenever we develop a new module, testing of the its usage or dependency on other modules should be analyzed. Leasing grows exponentially all around the globe. The trend on leasing an asset is increasing day by day. It would be so hard to manage and track leases with primitive documentation methods. Transformative lease web application can solve this huge issue in the world. User interface of the application should be developed with more comfort and all details necessary for the module using various UI elements or fields. Using HTML for defining the skeletal form of the web application, using CSS for designing the form accordingly to the business needs and JavaScript for basic validations on the forms, all together contributes the frontend mounted on middle ware or business logic layer which is C# and stored in Microsoft SQL database which forms the full stack web application.

## 2 Related Work

Management systems has been playing a vital role in every organization to keep track of all processes. Several kinds of research were taken to managing several similar cases in an organization through computers and databases. This section presents the related works carried out for management.

Data entry forms Using web UI elements on the web pages can capture the data automatically based on certain inputs on the fields or manually through the inputs provided by the user in any role. Booking a lease typically consists of lessor, lessee and assets involved with it. Lessor details depends on the financial company. Lease was the dependent entity on two independent entities asset and customer. Each customer or asset were consisting of unique identifier like ID or number to avoid repetition of same record. The group of assets involved in the lease contract were imported and lessor and lessee details were provided can create a lease of those assets with any period and money/income schedules in the web application. The data collected were validated and if the data not passing the validation, then we display a message of the issue, and the valid data will be stored in database. Since a full stack web application may contain vulnerabilities and bugs which could be exploited to dump data, the web application should go with penetration testing with cyber security team. In other hand, web application can be mounted on block chain network where hacking becomes almost impossible. Based on the data provided by the user, logically computed fields were shown in the form either stored or not stored in database. Entity Framework (EF) is an open-source object-relational mapping (ORM) framework for ODE.Net. It was originally shipped as an integral part of .NET framework. Starting with Entity Framework version 6, it has been delivered separately from the .NET Framework. The Entity Framework is a set of technologies in ADO.NET that supports the development of data-oriented software applications. Architects and

developers of data-oriented applications have typically struggled with the need to achieve two very different objectives. They must model the entities, relationships, and logic of the business problems they are solving, and they must also work with the data engines used to store and retrieve the data. The data can span multiple storage systems, each with its own protocols; even applications that work with a single storage system must balance the requirements of the storage system against the requirements of writing efficient and maintainable application code. This problem is generally referred to as the “object relational impedance match”

Many object-relational mapping (ORM) tools (aka "object-relational managers") have been developed to enable developers to work with data in the form of domain-specific objects and properties, such as customers and customer addresses, without having to concern themselves with the underlying database tables and columns where this data is stored. With an ORM, developers can work at a higher level of abstraction when they deal with data and can create and maintain data-oriented applications with less code than in traditional applications. Entity Framework is the ORM solution currently promoted for use within the Microsoft development stack.

### 3 Proposed System

Full stack web application to lease management system is specially designed to help financial companies in lease booking. The system introduces the digitalization to effectively track and manage leases with all applicable latest technology. It boosts the productivity of lease financing companies and help them keep track of all data relevant to a lease.

#### 3.1 Design Architecture

### ASP.NET Core Architecture

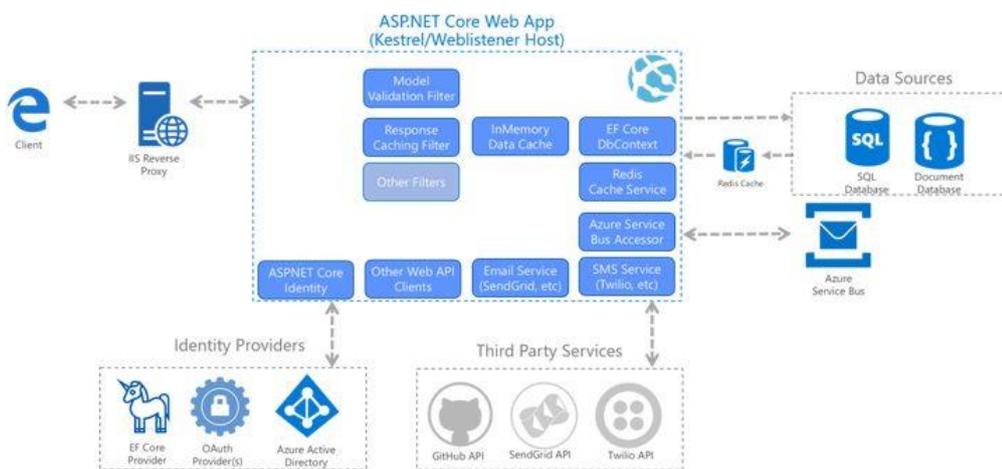


Fig.1 Technical architecture

Fig.1 depicts the design architecture of the proposed system. User should log in to the web application using their devices like laptop and smartphone. In the web application, when the lease is being booked, relevant data were captured. These data were then validated, processed, and shown in form wherever needed like total amount or whatever needed in business case. The HTML form elements style design nature will be changing accordingly to the nature of field required. Let us say a field which should be populated with amount, then a decimal like amount field will be set where only numbers were allowed to provide. Let us say an asset in a lease form which will be in form of a table as there can be multiple assets in a lease. There can be a reference of another entity in the form which will appear to select one of the collective values dumped from the middleware on the repository with whatever filter criteria. For example, displaying all active customers in a grid in the lease screen of which one will be chosen. There can also be a drop down for a field of which we choose one of the restricted set of values displayed in a drop down.

CSS does not get involved with business logic, but it is more important to provide a good-looking interface for a web application. It plays a vital role in providing sleeky and professional look or theme specific to lease financing company.

JavaScript plays a good role in resolving basic validations over the fields. Still, it could not cover much like in the middleware layer.

C# which I chosen to be the middleware layer which performs all complex computations with its object-oriented methodology and the powerful namespaces provided in it. Middleware is the layer where third party integrations can happen. Complex business logic was implemented in this layer in a more efficient way. C# also offers multi-threading and multi-processing technologies in its namespaces. Heavy loading tasks might require more time which were we use the above two concepts for more performance.

The interface between multiprocessing concepts and web application frontend user interface is the job module. Job module can accept whatever data required for a task from the user in the form of task parameters. Job module can be used in various business purposes like pulling or loading up data on certain criteria from or into the database.

### 3.2 Job Module

The Job module purpose is to give more performance on bult tasks. Let us say a scenario of uploading assets about 20000 assets into the system from the pre-existing system through an excel or other document types. If we try to do this on the main thread or main process, the user interface of the web application module will no longer be responsive while the code execution. The code execution also takes much time when the count of the data is huge. This is the exact scenario where job module comes in handy to provide good performance.

Job module runs a specific piece of business logic code in a separate process.

This process divides the process into sub tasks and each sub task will be running as separate task. After the completion of process, these sub tasks were integrated to produce the actual expected result. These sub tasks run simultaneously at a time. It reduces the workload on the main thread and thereby it optimizes the performance of the application.

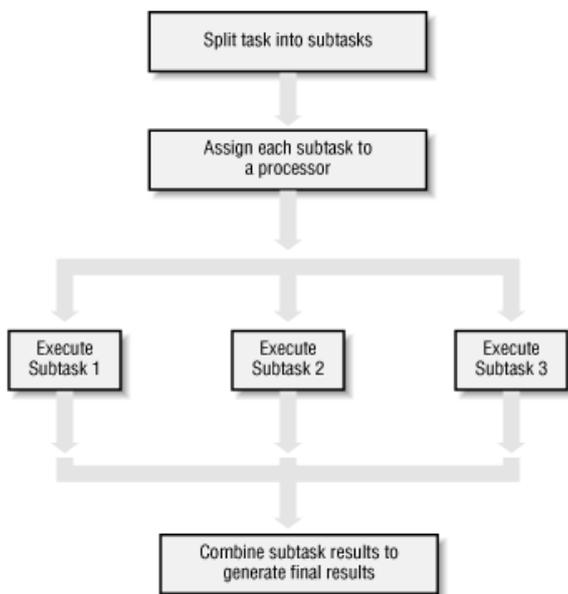
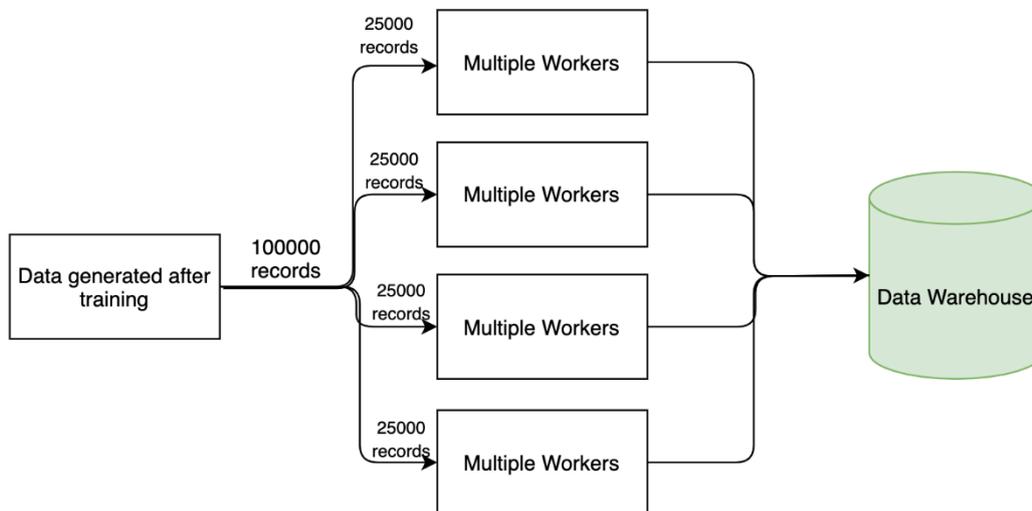


Fig.2 Internal functionality of job module

### Data output process with Multi-Processing



**Fig.3 Internal splitting of sub tasks in job module**

Fig 2 and Fig 3 clearly provides the visualization of the working of job module.

### 3.3 Data entry Based Modules

To capture the Realtime data, we use data entry forms created with basic HTML, CSS, JS, etc. Data entry modules include lease, asset, and customer module where we manually or automatically populate the field entries depending upon the functionality and business logic requirement. After entering data, we have workflow to approve or reject the data specified. These data were the analysed, validated, processed and stored in the database.

## 4 Result

This section provides the implementation results obtained. As mentioned, the web application along with modules developed can be integrated for lease management.

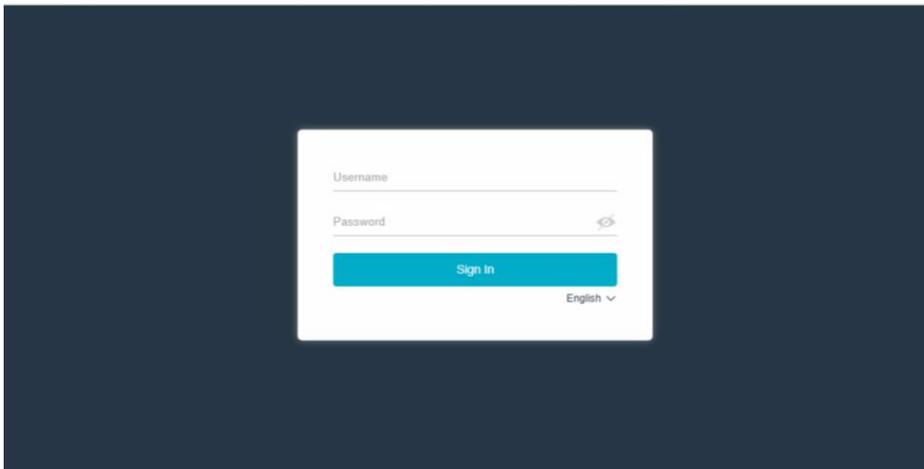


Fig.4 General login page

Login page for the user with any rule to login. Roles assignment and user creation will be done by administrator who have special credentials.

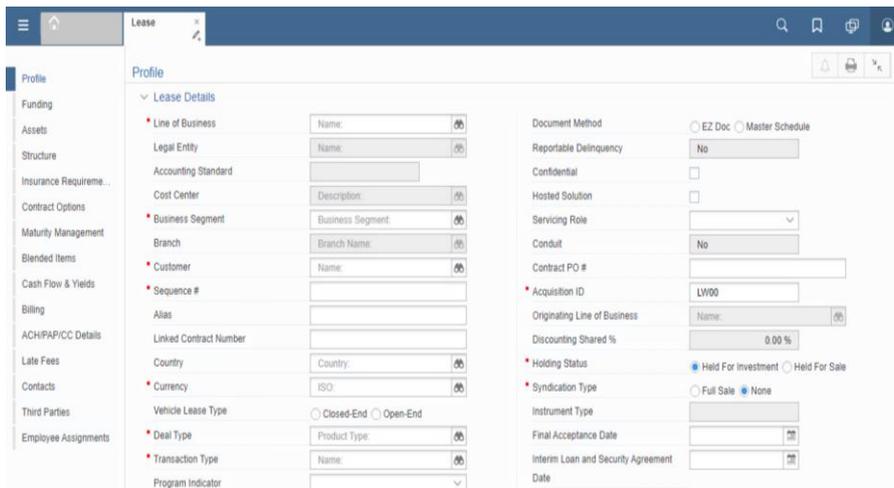
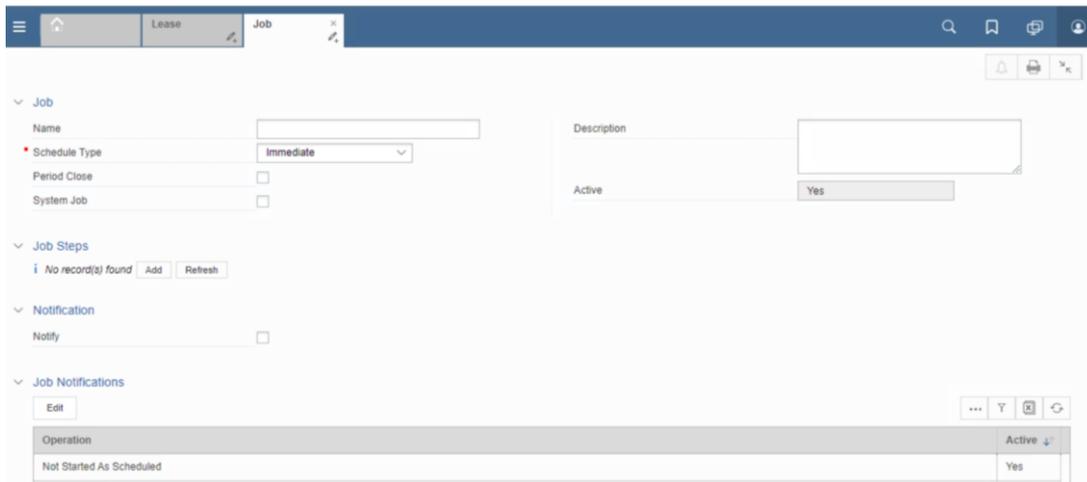


Fig.5 Lease module with all necessary fields to enter

Lease module is the main module of the web application where we create leases. We can observe different modules available related to lease in Fig.5. After filling up all necessary details and approval, lease will be booked in the web application and all data will be stored in the database



Operation	Active
Not Started As Scheduled	Yes

**Fig.6 Job module**

Fig.6 represents the job module of the web application. Here we can schedule the tasks, add multiple tasks in the sequential order. After filling all details, job will be running using a separate process and the result will be displayed.

## 5 Conclusion

Thus, the lease management system using full stack web application has been established with more reliability. Integration of the developed web modules and required third party services will be serving the actual need of the lease companies. The lease financing companies introduces new business concepts day by day. Lease web application also needs to be updated accordingly to the latest business needs.

The future work of this paper is to enhance the proposed system by incorporating various new modules required as per the dynamic lease growth. So that the reliability and scalability of the system can be ensured and can provide latest applicable technologies to the lease financing companies.

## References

1. [Web application development for the complete management of leasing](#) D Movilla Gayoso - 2021  
Development of Web and Mobile Application Based Online Buy, Sell and Rent Car System - <https://ieeexplore.ieee.org/document/9213208> (2020)
2. Car Renting... Its Development... And Future (2020). Retrieved from <https://www.automotive-fleet.com/147063/car-renting-its-development-and-future>
3. Ambrose, P. and Barlow, J. (1987), Housing Provision and House Building in Western Europe: Increasing Expenditure, Declining Output, Housing Markets and
4. Policies under Fiscal Austerity, London, Greenwood Press. [2] Cooper, M. (1998),
5. Agundu, P. U. C. and Mba, T. E. (2005), "Leasing Business in Nigeria: Operational issues and challenges" Multidisciplinary Journal of Empirical Research, 2(2), 1-7.
6. Lipe, R. C. (2001), "Lease accounting research and the G4+1 proposal", Accounting Horizons, 15(3), 299-310.
7. Metawa, S. A. (1995), "The evaluation of Lease Investment Opportunities: A decision - Support Methodology" Facilities, 13(8), 6-13
8. J. Crawford, K. Butler-Henderson, J. Rudolph. Malawi, M. Glowatz, R. Burton, P.Magni, and S. Lam, "COVID-19: 20 countries' higher education intra-period digital pedagogy responses," J. Appl. Learn. Teaching, vol. 3, no. 1, pp. 1-20, 2020.
9. Sage, K., Jackson, S., Fox, E. et al. The virtual COVID-19 classroom: surveying outcomes, individual differences, and technology use in college students. Smart Learn. Environ. 8, 27 (2021).