

TALLINN UNIVERSITY OF TECHNOLOGY

Rene Eller 193339IADB

# **AV Equipment Rental and Invoice Management System**

Home project in subject ICD0024 Web Applications with C#

**Tallinn 2024**

# Table of contents

<b>1. Introduction.....</b>	<b>3</b>
1.1.Backstory.....	3
1.2.Use cases.....	3
<b>2. Project Analysis.....</b>	<b>4-6</b>
2.1. Project Requirements.....	4
2.2. Key Features.....	5
2.3. Technologies.....	5
2.4. User Stories.....	6
<b>3. Development stages.....</b>	<b>7-8</b>
3.1.Initial Stage: Planning and Design.....	7
3.2.Initial Stage: Development.....	7
3.3.Further Development .....	8
<b>4.UI.....</b>	<b>9</b>
4.1 Customer Portal.....	9
4.2 Staff Portal.....	9
4.3 UI Sketches.....	10-11
<b>5. Entity Relationship Diagram.....</b>	<b>12</b>
<b>6.Conclusion.....</b>	<b>13-14</b>
6.1. Done.....	13
6.2. Still do be done.....	14

# 1. Introduction

The goal of this project is to create an application where customers can rent AV equipment and staff members can add equipment to rent and generate invoices for customers.

## 1.1 Backstory

New company wants to make a webpage to make renting AV equipment easy and so they can process rental requests online and keep track on which items are the most popular and see which customer has ordered which equipment. Also they want to generate invoices automatically(based on rental equipment request) and see who was the staff member who generated the invoice. Staff members need also to make rental requests since sometimes requests come by email or phone and need to be added to the system.

Customers can choose equipment filtered by type(audio, dj , video , lightning , special effects , decoration & stage , truss & rigging , electrical, etc)

If they choose exact dates , then the equipment list will show only available items for those specific dates. Then they can add items to basket and whenever they are done , they can see the total price and submit the request. Then staff member has to confirm the request and invoice will be generated.

## 1.2 Use Cases

### Customer:

1. Browse available equipment by type and date.
2. Add items to a basket and view the total price.
3. Submit a rental request for selected equipment.
4. View rental history and invoice details.
5. Provide feedback on their rental experience.

### Staff Member:

1. Add, edit, and delete equipment items from the rental list.
2. Confirm rental requests and generate invoices for customers.
3. View customers' rental history and invoice details.
4. Search and filter equipment, customers, and invoices.
5. Generate reports and analytics to analyze rental data.
6. Manage user accounts and permissions.
7. Integrate with external systems for inventory management and payment processing.

## 2. Project Analysis

The AV Equipment Rental and Invoice Management System is a comprehensive web application designed to streamline the AV equipment rental process for a new company. The system will provide a user-friendly interface for customers to browse and select equipment, place rental requests, and generate invoices. Staff members will have access to manage equipment, confirm rental requests, and generate invoices. This document outlines the project requirements, key features, technologies, and user stories to help guide the development process.

### 2.1. Project Requirements:

- 1. User Authentication:** Implement a secure user authentication system to allow customers and staff members to access the application.
- 2. Customer Portal:** Create a user-friendly interface for customers to browse available equipment, filter by type, choose dates, add items to a basket, and submit rental requests.
- 3. Equipment Management:** Implement functionality for staff members to add, edit, and delete equipment from the rental list. Include features to track equipment availability and rental history.
- 4. Rental Request Management:** Enable staff members to confirm rental requests and generate invoices for customers. Implement features to manage request status (pending, confirmed, cancelled) and track invoice generation.
- 5. Invoice generation:** Develop a system to automatically generate invoices based on rental equipment requests. Include features to display invoice details, calculate total prices, and associate invoices with staff members.
- 6. Customer History:** Implement functionality for staff members to view customers' rental history, including the equipment rented, dates, and invoice details.
- 7. Search and Filter:** Provide search and filter options for staff members to easily locate specific equipment, customers, or invoices.
- 8. Reporting and Analytics:** Develop reporting and analytics features to help staff members analyze rental data, identify popular equipment, and track customer preferences.

## **2.2.Key Features:**

- 1. Integration with External Systems:** Consider integrating with external systems such as payment gateways, inventory management software, or customer relationship management (CRM) systems to enhance the user experience and improve data accuracy.
- 2. Mobile App:** Develop a mobile app for customers to access rental information, place requests, and view invoices. This will make the rental process more convenient and accessible.
- 3.Equipment Reservation:** Implement a reservation system that allows customers to reserve equipment for future dates, ensuring availability and reducing the risk of overbooking.
- 4.Inventory Management:** Integrate an inventory management system to track equipment stock levels, prevent overbooking, and ensure accurate rental availability.
- 5.Customer Feedback:** Implement a system for customers to provide feedback on their rental experience, which can be used to improve the quality of service and equipment.
- 6.Security and Compliance:** Ensure the application meets security and compliance requirements, such as data encryption, access controls, and privacy policies.
- 7.Scalability and Performance:** Design the application to handle a large number of users and equipment, ensuring smooth performance and scalability.
- 8.Localization:** Implement localization features to support multiple languages and currencies, making the application accessible to customers worldwide.

## **2.3.Technologies:**

- 1.Backend:.**NET Core, Entity Framework Core (EF Core), ASP.NET Identity Core
- 2.Frontend:** Razor Pages, Bootstrap, jQuery, Blazor, Vue
- 3.Database:** Microsoft SQL Server, PostgreSQL, SQLite
- 4.Authentication and Authorization:** ASP.NET Identity Core, OAuth 2.0
- 5.Integration:** RESTful APIs, SignalR
- 6.Testing:** xUnit, Selenium
- 7.Deployment:** Azure App Service, Docker

## **2.4.User Stories:**

### **As a customer, I want to:**

1. Browse available equipment by type and date.
- 2.Add items to a basket and view the total price.
- 3.Submit a rental request for selected equipment.
- 4.View my rental history and invoice details.
- 5.Provide feedback on my rental experience.

### **As a staff member, I want to:**

- 1.Add, edit, and delete equipment from the rental list.
- 2.Confirm rental requests and generate invoices for customers.
- 3.View customers' rental history and invoice details.
- 4.Search and filter equipment, customers, and invoices.
- 5.Generate reports and analytics to analyze rental data.
- 6.Manage user accounts and permissions.
- 7.Integrate with external systems for inventory management and payment processing.

## **3. Development stages**

### **3.1.Initial Stage: Planning and Design**

1. Define project requirements and objectives.
2. Identify the key stakeholders and their roles in the system.
3. Create a detailed project plan, including timelines, resources, and milestones.
4. Design the user interface and user experience, considering the key functionalities and interactions.
5. Define the data structures and relationships for the system, including entities, attributes, and relationships.
6. Create a database schema and design the necessary tables and relationships.
7. Identify the necessary external systems and APIs that will be integrated with the system.
8. Develop a high-level architecture and design for the system, including components, modules, and their interactions.
9. Create a prototype or mockup of the user interface and user experience.
10. Conduct a thorough analysis of the project requirements, objectives, and design, ensuring that it meets the needs of the stakeholders.

### **3.2.Initial Stage: Development**

1. Set up the development environment, including the necessary tools, frameworks, and libraries.
2. Create the necessary project structure and folders.
3. Implement the user interface and user experience, following the design and prototype.
4. Develop the backend components, including the necessary controllers, services, and repositories.
5. Implement the necessary database schema and create the necessary tables and relationships.
6. Integrate with external systems and APIs, as identified in the planning and design stage.
7. Implement the necessary business logic and workflows, including rental request processing, invoice generation, and reporting.
8. Test the system thoroughly, ensuring that all key functionalities and interactions work as expected.
9. Perform a code review and ensure that the code is clean, well-structured, and follows best practices.
10. Deploy the system to a staging or testing environment for further testing and evaluation.

### **3.3.Further Development**

- 1.** Implement additional features and functionalities, such as equipment reservation, equipment tracking, and customer feedback management.
- 2.** Optimize the system for performance, scalability, and security, considering factors like database optimization, caching, and security measures.
- 3.** Implement user authentication and authorization, ensuring that only authorized users can access and modify the system.
- 4.** Implement error handling and exception management, ensuring that the system can handle and recover from errors gracefully.
- 5.** Implement logging and monitoring capabilities, allowing for the tracking and analysis of system activities.
- 6.** Implement internationalization and localization support, allowing the system to be used by customers in different languages and regions.
- 7.** Implement unit testing and integration testing, ensuring that the system works as expected and that all key functionalities are covered.
- 8.** Perform regular maintenance and updates to the system, addressing any bugs, issues, or performance concerns.
- 9.** Continuously monitor and analyze the system's usage and performance, making necessary improvements and optimizations.
- 10.** Implement additional features and functionalities as needed, based on user feedback and requirements.

## 4.UI

### 4.1.Customer Portal:

Home Page: Display available equipment categories, search bar, and popular equipment.

Equipment List: Show a list of available equipment based on category and date filters.

Equipment Details: Display detailed information about a specific equipment item, including images and availability.

Basket: Show the selected equipment items, total price, and a checkout button.

Rental Request: Allow customers to submit a rental request for selected equipment, including contact information and payment options.

Rental History: Display a list of past rental requests and invoices for the logged-in customer.

Invoice Details: Show detailed information about a specific rental invoice, including equipment rented, dates, and total price.

Feedback: Allow customers to provide feedback on their rental experience.

### 4.2.Staff Portal:

Dashboard: Show a summary of recent activities, such as pending rental requests, upcoming rentals, and popular equipment.

Equipment Management: Allow staff members to add, edit, and delete equipment items, including images and availability.

Rental Request Management: Show a list of pending rental requests, allowing staff members to confirm or cancel requests.

Invoice Generation: Allow staff members to generate invoices for confirmed rental requests.

Customer History: Display a list of past rental requests and invoices for a specific customer.

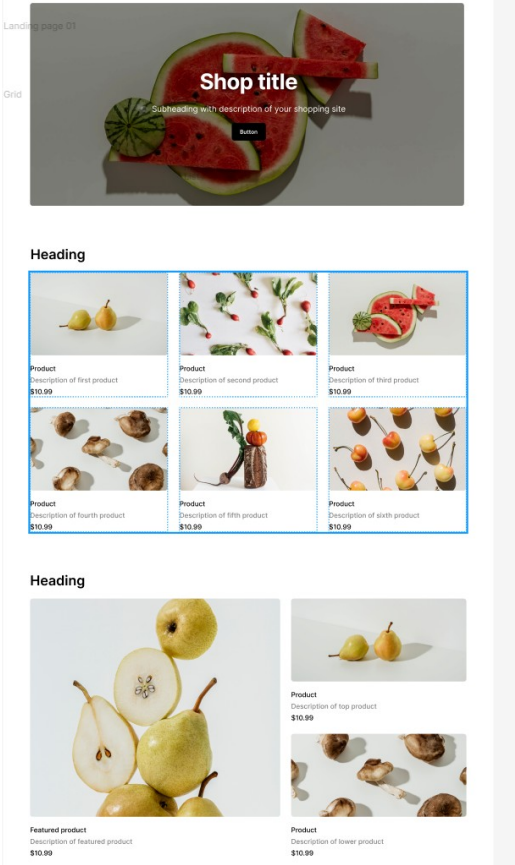
Search and Filter: Provide search and filter options for equipment, customers, and invoices.

Reporting and Analytics: Show various reports and analytics, such as popular equipment, customer preferences, and rental trends.

User Management: Allow staff members to manage user accounts and permissions.

# 4.2.UI Sketches

## Main Page



## Managing Products for Admin/Staff




Product Details View for customer

Site name

PagePagePage

Button



\$254.019

Product name

Subheading


\$10.99

Body text for describing why this product is simply a must-buy

Add to cart

Text box for additional details or fine print


Related products



Product

Body text for first product


\$10.99



Product

Body text for second product

\$10.99



Product

Body text for third product

\$10.99

Rental confirmation form for customer

EMAIL:

asd@asda.eee

FIRST NAME:

Peep

LAST NAME:

Jeep

RENTAL START TIME:

Sun, May 7, 2023 12:00 PM  
SELECT DATE & TIME

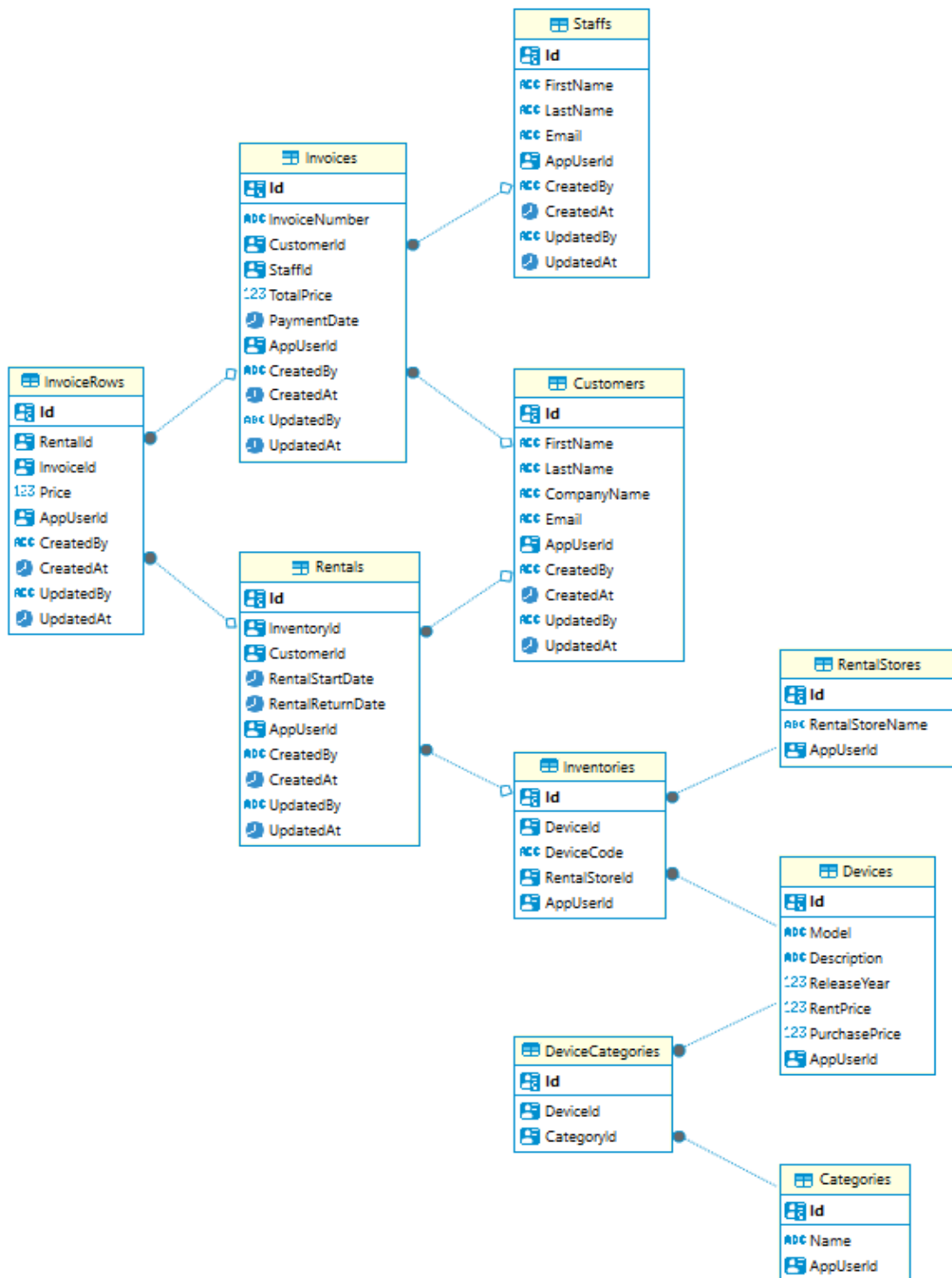
RENTAL RETURN TIME:

Wed, May 22, 2024 2:00 AM  
SELECT DATE & TIME

☒ ACCEPT TERMS AND CONDITIONS

Send Post

## 5. Entity Relationship Diagram



# 6.Conclusion

## 6.1.Done

### **Planning and design:**

Defined project requirements and objectives.

Identified key stakeholders and their roles in the system.

Created a detailed project plan, including timelines, resources, and milestones.

Designed the user interface and user experience, considering key functionalities and interactions.

Defined the data structures and relationships for the system, including entities, attributes, and relationships.

Created a database schema and designed the necessary tables and relationships.

Identified the necessary external systems and APIs that will be integrated with the system.

Developed a high-level architecture and design for the system, including components, modules, and their interactions.

Created a prototype or mockup of the user interface and user experience.

Conducted a thorough analysis of the project requirements, objectives, and design, ensuring that it meets the needs of the stakeholders.

### **Initial development:**

Set up the development environment, including the necessary tools, frameworks, and libraries.

Created the necessary project structure and folders.

Implemented the user interface and user experience, following the design and prototype.

Developed the backend components, including controllers, services, and repositories.

Implemented the necessary database schema and created the necessary tables and relationships.

Integrated with external systems and APIs, as identified in the planning and design stage.

Implemented the necessary business logic and workflows, including rental request processing, invoice generation, and reporting.

Tested the system thoroughly, ensuring that all key functionalities and interactions work as expected.

Performed a code review and ensured that the code is clean, well-structured, and follows best practices.

Deployed the system to a staging or testing environment for further testing and evaluation.

## **6.2.Still to be done:**

### **Further development:**

Implement additional features and functionalities, such as equipment reservation, equipment tracking, and customer feedback management.

Optimize the system for performance, scalability, and security, considering factors like database optimization, caching, and security measures.

Implement user authentication and authorization, ensuring that only authorized users can access and modify the system.

Implement error handling and exception management, ensuring that the system can handle and recover from errors gracefully.

Implement logging and monitoring capabilities, allowing for the tracking and analysis of system activities.

Implement internationalization and localization support, allowing the system to be used by customers in different languages and regions.

Implement unit testing and integration testing, ensuring that the system works as expected and that all key functionalities are covered.

Perform regular maintenance and updates to the system, addressing any bugs, issues, or performance concerns.

Continuously monitor and analyze the system's usage and performance, making necessary improvements and optimizations.

Implement additional features and functionalities as needed, based on user feedback and requirements.