

INVENTORY MANAGEMENT SYSTEM
FOR KELAB FOTOKREATIF
(IFOTO)

LIO KOCK HOCK

UNIVERSITI TEKNOLOGI MALAYSIA

**DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND
COPYRIGHT**

Author's full name : Lio Kock Hock

Date of Birth : 19/11/2003

Title : Inventory Management System for Kelab Fotokreatif

Academic Session : 2024/2025 2

I declare that this thesis is classified as:



CONFIDENTIAL

(Contains confidential information under the Official Secret Act 1972)*



RESTRICTED

(Contains restricted information as specified by the organization where research was done)*



OPEN ACCESS

I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
 - a. The thesis is the property of Universiti Teknologi Malaysia
 - b. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
 - c. The Library has the right to make copies of the thesis for academic exchange.

Certified by:



SIGNATURE OF STUDENT

A22EC0185

MATRIX NUMBER

SIGNATURE OF SUPERVISOR

DR. MUHAMMAD
KHATIBSYARBINI BIN ABD RAHIM

NAME OF SUPERVISOR

Date: 1 MAY 2025

Date: 26 JUNE 2025

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

“I hereby declare that we have read this thesis and in my opinion this thesis is sufficient in term of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering) with Honours”

Signature : _____

Name of Supervisor : DR. MUHAMMAD KHATIBSYARBINI BIN ABD RAHIM

Date : 26 JUNE 2025

INVENTORY MANAGEMENT SYSTEM
FOR KELAB FOTOKREATIF
(IFOTO)

LIO KOCK HOCK

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computing
Universiti Teknologi Malaysia

MAY 2025

DECLARATION

I declare that this thesis entitled "*Inventory Management System for Kelab Fotokreatif (IFoto)*" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :



Name : LIO KOCK HOCK
Date : 1 MAY 2025

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have been that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with numbers of people, Kelab Fotokreatif members and researchers. They have contributed towards my understanding and thoughts. I wish to express my sincere gratitude to my main thesis supervisor, Dr. Muhammad Khatibsyarbini Bin Abd Rahim, for encouragement, guidance, critics, motivaton, advices and friendship. Without his continued support and interest, this thesis would not have been the same as presented here.

I want to express my sincere appreciation to my stakeholder as well, the Kelab Fotokreatif members, especially the leader of Equipment Executive Committee of the club, Muhammad Alif Nabil Bin Junaidi and the president of the club, Muhammad Taufiq Bin Mohd Zaid who have been very cooperative in providing me business logic and their requirements. My sincere appreciation also extends to a close friend of mine, Ahmad Saifudin Bin Nardi Susanto, the one that spark my interest in developing Inventory Management System for Kelab Fotokreatif (IFoto) by exposing me the challenges faced by the Kelab Fotokreatif members in managing their inventory. Without a doubt, this club has changed my life for the better.

ABSTRACT

Kelab Fotokreatif UTM is a club that organize photography and video services and activities that involves academic, practical, and entrepreneurship for all members to produce high quality and professional videos and photos. As of today, this club manage their inventory manually through phone calls or messaging platform which causes inefficiencies and heavy workload during an event. The purpose of this thesis is to specify the business requirements, design the architecture, implement, and test the Inventory Management System for Kelab Fotokreatif (IFoto). This system caters the problems faced by the club by providing a centralized platform for real-time equipment tracking, responsibility assignment, manage equipment requests and returns and the rental of the equipment with online payment gateway. This system also utilizes Role-Based Access Control (RBAC) to manage the authentication and authorization of the user of the system in order ensure distinct permission according to roles such as Administrator, Equipment Executive Committee, Equipment Committee Member, and User. This project utilizes the Agile methodology and Software Development Lifecycle (SDLC) in which it emphasizes modularity in development and each module follows the SDLC process where it starts with business requirements gathering, system analysis and design, system implementation using React.js, Java Spring and MySQL and testing using Black-box testing, White-box testing and user testing. In short, this system aims to improve the work experience of the club members by increasing their efficiencies and reduce their manual workload.

ABSTRAK

Kelab Fotokreatif UTM merupakan sebuah kelab yang menganjurkan perkhidmatan dan aktiviti fotografi dan video yang melibatkan akademik, praktikal, dan keusahawanan untuk semua ahli menghasilkan video dan gambar yang berkualiti tinggi dan profesional. Sehingga hari ini, kelab ini menguruskan inventori mereka secara manual melalui panggilan telefon atau platform pemesejan yang menyebabkan ketidakcekapan dan beban kerja yang berat semasa sesuatu acara. Tujuan tesis ini adalah untuk menentukan keperluan perniagaan, mereka bentuk seni bina, melaksanakan, dan menguji Sistem Pengurusan Inventori untuk Kelab Fotokreatif (IFoto). Sistem ini menangani masalah yang dihadapi oleh kelab dengan menyediakan platform terpusat untuk pengesahan peralatan masa nyata, penugasan tanggungjawab, mengurus permintaan dan pemulangan peralatan dan penyewaan peralatan dengan gerbang pembayaran dalam talian. Sistem ini juga menggunakan Kawalan Akses Berasaskan Peranan (RBAC) untuk mengurus pengesahan dan kebenaran pengguna sistem bagi memastikan kebenaran berbeza mengikut peranan seperti Pentadbir, Jawatankuasa Eksekutif Peralatan, Ahli Jawatankuasa Peralatan dan Penyewa. Projek ini menggunakan metodologi “Hybrid Agile-Waterfall” dan Kitaran Hayat Pembangunan Perisian (SDLC) di mana ia menekankan modulariti dalam pembangunan dan setiap modul mengikut proses SDLC di mana ia bermula dengan pengumpulan keperluan logik, analisis dan reka bentuk sistem, pelaksanaan sistem menggunakan “React.js”, “Java Spring” dan “MySQL” dan ujian menggunakan ujian kotak Hitam, ujian kotak putih dan ujian pengguna. Secara ringkasnya, sistem ini bertujuan untuk meningkatkan pengalaman kerja ahli kelab dengan meningkatkan kecekapan mereka dan mengurangkan beban kerja manual mereka.

TABLE OF CONTENTS

	TITLE	PAGE
DECLARATION		ii
DEDICATION		iii
ACKNOWLEDGEMENT		iv
ABSTRACT		v
ABSTRAK		vi
TABLE OF CONTENTS		vii
LIST OF TABLES		x
LIST OF FIGURES		xi
LIST OF ABBREVIATIONS		xiv
LIST OF APPENDICES		xv
CHAPTER 1 INTRODUCTION		1
1.1 Introduction	1	
1.2 Problem Background	2	
1.3 Project Aim	3	
1.4 Project Objectives	3	
1.5 Project Scope	4	
1.6 Project Importance	4	
1.7 Report Organization	5	
CHAPTER 2 LITERATURE REVIEW		7
2.1 Introduction	7	
2.2 Case Study	7	
2.2.1 Company Organization Structure	8	
2.2.2 Manual Operation	8	
2.3 Current System Analysis	9	
2.3.1 Rentman	9	
2.3.2 Cheqroom	10	

2.4	Comparison between existing systems	11
2.5	Literature Review of Technology Used	12
2.5.1	React.js	12
2.5.2	Spring Boot	12
2.5.3	MySQL	13
2.5.4	React Redux	13
2.5.5	GitHub	14
2.5.6	Visual Studio Code	15
2.5.7	Enterprise Architect	15
2.5.8	Postman	15
2.5.9	Jira	16
2.6	Chapter Summary	17
CHAPTER 3	SYSTEM DEVELOPMENT METHODOLOGY	19
3.1	Introduction	19
3.2	Phases of the Chose Methodology	19
3.2.1	Requirement Analysis	20
3.2.2	System Design	20
3.2.3	Coding & Implementation	21
3.2.4	System Testing	21
3.2.5	System Deployment	21
3.3	Technology Used Description	22
3.4	System Requirement Analysis	23
3.4.1	Software Requirements	23
3.4.2	Hardware Requirements	24
3.5	Chapter Summary	25
CHAPTER 4	REQUIREMENT ANALYSIS AND DESIGN	26
4.1	Introduction	26
4.2	Requirement Analysis	26
4.3	Project Design	33
4.3.1	System Architectural Style and Pattern	33
4.3.2	Complete System Design and its Subsystem	34

4.4	Database Design	42
4.5	Interface Design	50
4.5.1	User Interface for Equipment Request and Rentals Module	50
4.5.2	User Interface for Equipment Management Module	58
4.5.3	User Interface for Role and Event Management Module	61
4.6	Chapter Summary	63
CHAPTER 5	CONCLUSION	65
5.1	Introduction	65
5.2	Achievement of Project Objectives	66
5.3	Suggestions for Future Improvement	67
REFERENCES		69

LIST OF TABLES

TABLE NO.	TITLE	PAGE
Table 2.1	Comparison IFoto with existing systems.	11
Table 3.1	Software Requirements for IFoto	23
Table 3.2	Hardware Requirements for IFoto	24
Table 4.1	Description of Use Cases for IFoto	27
Table 4.2	Use Case Description for UC012 Update Equipment	29
Table 4.3	Data Dictionary for Inventory Management System for Kelab Fotokreatif (IFoto).	42

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
Figure 2.1	Kelab Fotokreatif's organization chart	8
Figure 2.2	User Interface of Rentman	10
Figure 2.3	User Interface of Dashboard of Cheqroom	10
Figure 2.4	ORM Technique in Java Spring Projects (Emon, 2024)	13
Figure 2.5	With and Without Redux in DOM Tree (Corman, 2020)	14
Figure 3.1	Hybrid Agile-Waterfall Methodology (Shinde, 2025)	20
Figure 4.1	Use case diagram of Inventory Management System for Kelab Fotokreatif (IFoto)	27
Figure 4.2	Activity Diagram for UC012 Update Equipment	31
Figure 4.3	Sequence Diagram for UC012 Update Equipment	32
Figure 4.4	Architecture Model of 3-Tier Architecture in Inventory Management System for Kelab Fotokreatif (IFoto)	33
Figure 4.5	Package Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)	34
Figure 4.6	Package Diagram for Inventory Management System for Kelab Fotokreatif (IFoto).	35
Figure 4.7	Complete Backend Class Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)	36
Figure 4.8	Frontend Component Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)	37
Figure 4.9	Class Diagram for Authentication and Authorization Module	38
Figure 4.10	Class Diagram for Equipment Request and Rental Module (Equipment Rent)	38
Figure 4.11	Class Diagram for Equipment Request and Rental Module (Equipment Request)	39
Figure 4.12	Class Diagram for Equipment Management Module	39
Figure 4.13	Class Diagram for Event Management Module	40
Figure 4.14	Entity Relationship Diagram (ERD) of Inventory Management System for Kelab Fotokreatif (IFoto)	42

Figure 4.15	Interface of List of Equipment (Rent Equipment)	50
Figure 4.16	Interface of Cart Page (Rent Equipment)	51
Figure 4.17	Interface of Rental Form (Rent Equipment)	51
Figure 4.18	Interface of List of Equipment Rents (Rent Equipment)	52
Figure 4.19	Interface of Specific Equipment Rent Details (Rent Equipment)	52
Figure 4.20	Interface of Invoice (Rent Equipment)	53
Figure 4.21	Interface of Receipt (Rent Equipment)	53
Figure 4.22	Interface of Payment Page (Rent Equipment)	54
Figure 4.23	Interface of Equipment Return Page (Rent Equipment)	54
Figure 4.24	Example of Interface for Successful Action (Rent Equipment)	55
Figure 4.25	Interface of List of Event (Request Equipment)	55
Figure 4.26	Interface of List of Equipment (Request Equipment)	56
Figure 4.27	Interface of Cart Page (Request Equipment)	56
Figure 4.28	Interface of Equipment Request Form (Request Equipment)	57
Figure 4.29	Interface of Specific Equipment Request Details (Request Equipment)	57
Figure 4.30	Interface of List of Club's Equipment (Manage Equipment)	58
Figure 4.31	Interface of Update Equipment Form (Manage Equipment)	58
Figure 4.32	Interface of Add Equipment Form (Manage Equipment)	59
Figure 4.33	Interface of List of Equipment Request and Rentals (Manage Equipment Requests and Rentals)	59
Figure 4.34	Interface of Managing Equipment Request and Rentals (Manage Equipment Requests and Rentals)	60
Figure 4.35	Interface of List of Equipment Returns (Manage Equipment Returns)	60
Figure 4.36	Interface of Managing Equipment Returns (Manage Equipment Returns)	61
Figure 4.37	Interface of List of Event	61
Figure 4.38	Interface of Manage Club Members	62

Figure 4.39 Interface of Add Event Form 62

Figure A.1 Example of Prompt Used 71

LIST OF ABBREVIATIONS

HTML	-	Hypertext Markup Language
CSS	-	Cascading Style Sheets
PHP	-	Hypertext Preprocessor
RDBMS	-	Relational Database Management System
MVC	-	Model-View-Controller
UI	-	User Interface
DOM	-	Document Object Model
JPA	-	Java Persistence API
ORM	-	Object Relational Model
SysML	-	Systems Modelling Language
BPMN	-	Business Process Model and Notation
API	-	Application Programming Interface

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
Appendix A	List of Prompts	71
Appendix B	Software Requirements Specification (SRS)	72
Appendix C	Software Design Document (SDD)	147
Appendix D	Software Test Documentation (STD)	259

CHAPTER 1

INTRODUCTION

1.1 Introduction

Kelab Fotokreatif UTM is a club that organize photography and video services and activities that involves academic, practical, and entrepreneurship for all members to produce high quality and professional videos and photos. This club consist of total 25 active members that work together to offer photography and videography coverage for events, including for external parties' event as well as internal club activities. So, during an event, the assigned Equipment Committee Member will request for the club's equipment for an event from the Equipment Executive Committee for media coverage purposes. The user would often rent equipment from Equipment Executive Committee via WhatsApp or verbal request and pay using QR code. Examples of the club's equipment are Camera's Battery, Battery Charger, Speedlight, SD Card, Tripod, Camera Strap, Canon Camera, Nikon Camera and many more.

This thesis introduces Inventory Management System for Kelab Fotokreatif (IFoto) designed to efficiently centralized and streamline the inventory management process of the club to shortages and improve the productivity with fewer errors to address, so that the club can focus on the providing higher-quality services efficiently (Hayes, 2024).

1.2 Problem Background

One key challenge faced by the club members and user using the current system is lack of structured tracking of the availability of the equipment as it led to miscommunication among the club members and users and double-booking problems since it lacks automated booking conflict detection. The club members and user rely on personal message or phone calls to inquire information from the Equipment Executive Committee. Consequently, Equipment Executive Committee Member could experience overwhelming workload during peak event periods.

Furthermore, Equipment Executive Committee currently handle the equipment usage log through Google Spreadsheet manually. This manual process is not scalable because as the numbers of events and equipment increases, managing the equipment manually will become inefficient, making difficult for the Equipment Executive Committee to keep the spreadsheet record up-to-date.

The process of renting and paying for the club's equipment can be troublesome and slow too as it requires manual processes mostly through WhatsApp or verbal request. The process is devoid of a central platform to provide space for rental requests, approve bookings, or track payments that will cause delay, lost messages, and data inconsistency. Besides that, the lack of online payment integration and automatic invoicing causes problem for users and the Equipment Executive Committee. The process is also more susceptible to human error, miscommunication, and difficulty in keeping an open audit trail for financial records.

1.3 Project Aim

The aim of this project is to develop Inventory Management System for Kelab Fotokreatif (IFoto) to streamline real-time equipment tracking and quality check, online equipment booking and payment, centralized equipment management, and role and event management.

1.4 Project Objectives

The objectives of the project are:

- (a) To analyse and determine the necessary requirements to develop the Inventory Management System for Kelab Fotokreatif (IFoto).
- (b) To design the web-based application that provide intuitive interaction for the users that is easy to navigate.
- (c) To develop the Inventory Management System for Kelab Fotokreatif (IFoto) based on the stakeholders' requirements.
- (d) To manage the testing process on functionality and non-functionality of the Inventory Management System for Kelab Fotokreatif (IFoto) with the targeted users.

1.5 Project Scope

The scopes of the project are:

- (a) The Inventory Management System for Kelab Fotokreatif (IFoto) is built as web application for broader range of device accessibility.
- (b) The system includes four main users for Administrators, Equipment Executive Committee, Equipment Committee Member, and User.
- (c) The Administrators are responsible to manage the club member roles and monitor the equipment.
- (d) The Equipment Executive Committee are responsible to monitor and manage the club's equipment and equipment booking workflow.
- (e) The Equipment Committee Member can request for the club's equipment for an event.
- (f) The User can rent club's equipment and make payment through the web application.

1.6 Project Importance

The importance of realizing this project is to provide the Kelab Fotokreatif's members a centralized the inventory management process to promote better accuracy, efficiency, and transparency for the club members. It reduces manual workload which reduces the risk of human errors as well as improves communication and teamwork among club members regarding the quality control of their equipment. Overall, the system help club operate in a more organized and professional environment.

1.7 Report Organization

This part will briefly outline the organisation of the report of this thesis. Chapter 1 talks about the introduction, background of the problem, aim of the project, objectives, scope, and significance of the project. In Chapter 2, we shall talk about the literature review of the thesis that is to study and analyse the existing systems with Inventory Management System for Kelab Fotokreatif (IFoto) and technology and tools that will be used to develop the proposed system. Chapter 3 describes how Inventory Management System was developed in Kelab Fotokreatif (IFoto). Chapter 4 talks about the requirement analysis and design of the Inventory Management System of Kelab Fotokreatif (IFoto). Chapter 5 demonstrates the implementation and testing of the system. Chapter 6 talks about the end of the PSM 1 thesis.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This literature review covers different points concerning inventory management system, real-time equipment tracking, user experience, equipment request and renting management, role-based access control (RBAC) and automation. It will also enable us to enhance the Inventory Management System of Kelab Fotokreatif (IFoto) by studying the available systems, frameworks, and research. In addition, the chapter will also talk about the analysis of existing systems, the technology and tools that will be employed in the design, development, and testing of the system.

2.2 Case Study

A related case study would be a book titled “Rancang Bangun Aplikasi Inventory Berbasis Web dengan Menggunakan Model MVC” authored by Ariandi Nugroho, Dewi Rahma Sari, Heru Dwi Permana, and Raka Surya Negara (Nugroho, Sari, Permana, & Negara, 2021). The web-based inventory application is developed using popular technologies which are Hypertext Preprocessor (PHP), Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), Bootstrap, Xampp and PhpMyAdmin for database management. This thesis discusses about the development of web-based inventory application by implementing Model-View-Controller (MVC) architecture. The system aims to provide a centralized platform to track or monitor incoming or outgoing goods as well as manage daily and monthly data. In this thesis the authors emphasize crucial benefit of implementing such

applications in operational activities which directly related to my topic, Inventory Management System for Kelab Fotokreatif (IFoto).

2.2.1 Company Organization Structure

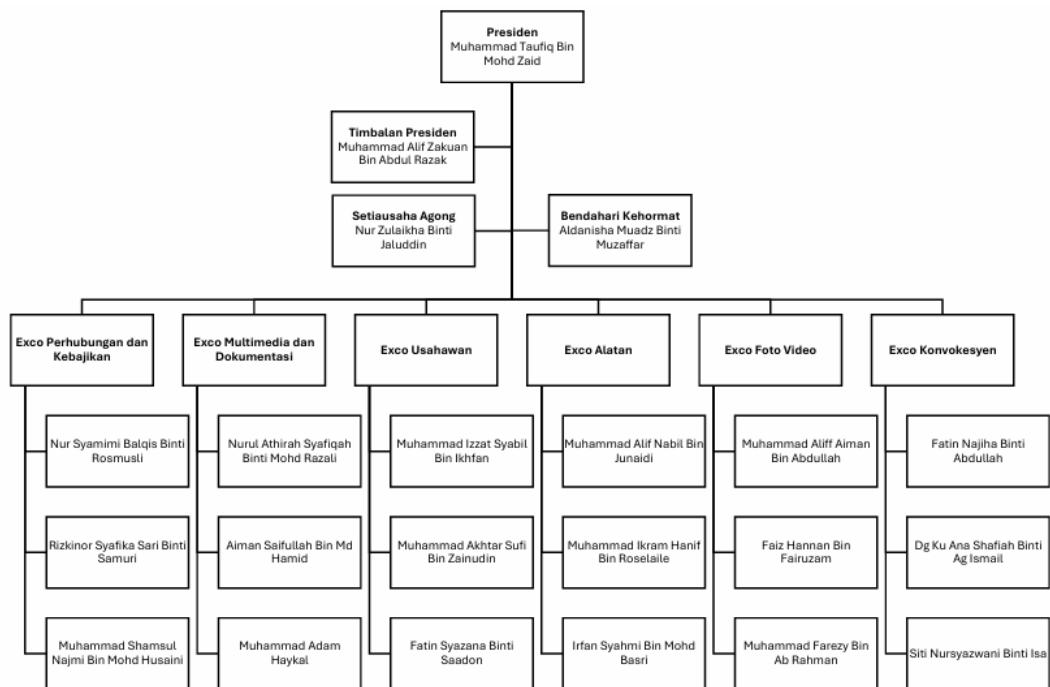


Figure 2.1 Kelab Fotokreatif's organization chart

Based on Figure 2.1, “Exco Alatan” also known as equipment executive committee would be my main stakeholder for Inventory Management System for Kelab Fotokreatif (IFoto). Other than that, the president and vice president represent the administrator for the system.

2.2.2 Manual Operation

Kelab Fotokreatif UTM is a club that organize photography and video services and activities that involves academic, practical, and entrepreneurship for all members to produce high quality and professional videos and photos. This club consist of total 25 active members that work together to offer photography and

videography coverage for events, including for external parties' event as well as internal club activities. So, during an event, the assigned Equipment Committee Member will request for the club's equipment for an event from the Equipment Executive Committee for media coverage purposes. The user would often rent equipment from Equipment Executive Committee via WhatsApp or verbal request and pay using QR code. Examples of the club's equipment are Camera's Battery, Battery Charger, Speedlight, SD Card, Tripod, Camera Strap, Canon Camera, Nikon Camera and many more.

2.3 Current System Analysis

Among the many existing systems, Rentman and Cheqroom are popular but they serve different audiences as they provide distinct features. Rentman is one of the best equipment management systems with project scheduling feature. On the other hand, Cheqroom is more focused on enterprise asset tracking.

2.3.1 Rentman

Rentman (2015) that handle rentals and event productions in a single platform and work on projects, tasks, and the right information reaching the right individuals. It is popular among video production houses and film production crews with robust inventory and crew management features as well scheduling feature for project planning. Other than that, it also provides real-time availabilities tracking and equipment feature along with rentals management.

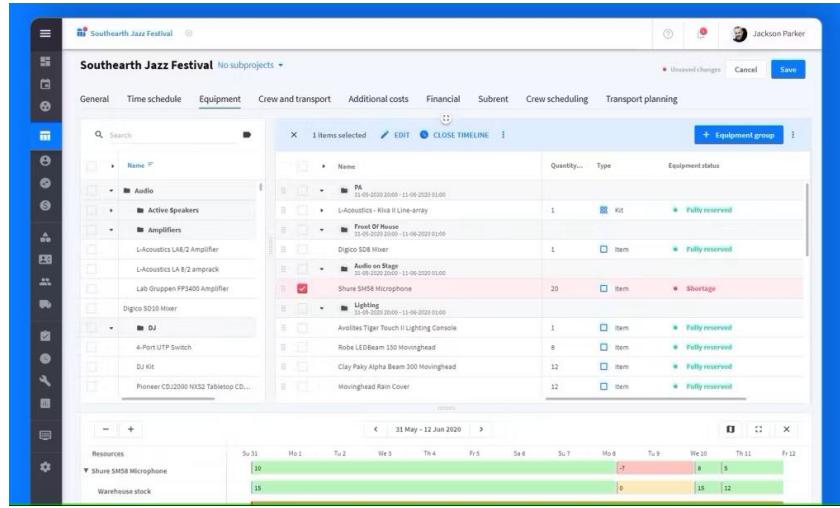


Figure 2.2 User Interface of Rentman

2.3.2 Cheqroom

Cheqroom (2014) is a software to manage equipment, which is used by teams that share equipment among people, projects, and places. It is popular among corporate IT department, universities, schools, and government institution in which it focusses on equipment management and tracking as well as asset auditing and tracking. Besides that, it also provides Equipment checkouts and reservation that prevents scheduling conflicts.

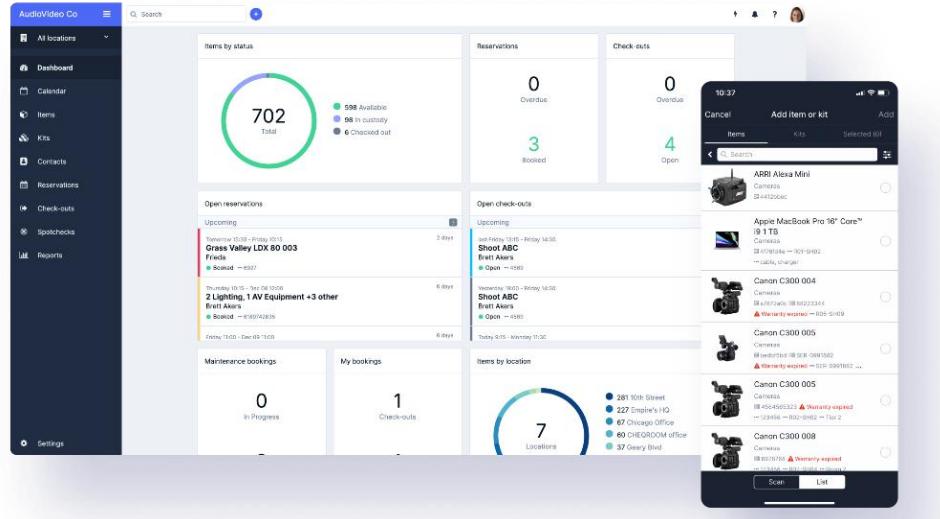


Figure 2.3 User Interface of Dashboard of Cheqroom

2.4 Comparison between existing systems

Table 2.1 Comparison IFoto with existing systems.

Features	Rentman	Cheqroom	IFoto
Target Users	AV/event rental companies and film crews	Schools, universities, and corporate IT teams	Kelab Fotokreatif's members
Purpose	Equipment rental, crew scheduling, logistics	Equipment check-in/check-out and asset tracking	Club-oriented equipment management and rental system
Supports External Rentals	Yes	No	Yes
Online Payment Integration	No	No	Yes
Multi-role Access	Yes	Yes	Yes
Designed for Small Clubs	No	No	Yes
Real-time Availability Tracking	Yes	Yes	Yes
Mobile Accessibility	Yes	Yes	Yes
Event-Based Equipment Assignment	Yes	No	Yes
Cost	Subscription-based	Subscription-based	Free

2.5 Literature Review of Technology Used

2.5.1 React.js

React or ReactJS is a JavaScript library and framework that is open-source and created by Facebook, currently known as META. We can develop frontend of a system in a short time and with much less written code and save time with its huge library and community. React.js is used as V or View in Model-View-Controller (MVC) architecture that takes care of the view layer of the application. React allows separated UI as a reusable unit or component. It is superior to the normal HTML, which must code the entire UI as single unit. To render web pages more quickly and create highly dynamic and responsive web applications, the React framework combines the speed and efficiency of JavaScript with a more efficient way by manipulating the DOM (Herbert, 2022).

2.5.2 Spring Boot

Spring Boot is an open-source Java-based framework and vast library developed by Pivotal Software that helps developers to build an independent, production-quality system applications by reducing complex configuration and boilerplate code (Walls, 2016). In Model-View-Controller (MVC) architecture, Spring Boot serves as "C" or "Controller" which handles the business logic of the web-based application in the backend and API requests from frontend. It fits for creating expandable online applications since it offers quick development by using auto-configuration, embedding web servers, and starting dependencies. Additionally, RESTful APIs is an application programming interface (API) follows the principles of the Representational State Transfer (REST) architectural style and it is a way to enables communication between front-end and back-end application over the internet, allowing them to request and manage the data (Bigelow & Gillis, 2024).

2.5.3 MySQL

MySQL is popular open-source relational database management systems (RDBMS) was first created by Swedish company, MySQL AB in 1995. The structured data in the database are managed and accessed by using Structured Query Language (SQL). In fact, well-known websites such as Facebook, Twitter, Booking.com, GitHub, YouTube, etc leverage it due to its scalability, reliability, performance, data consistency and ease of integration with web applications which lead to large number of communities. Other than that, modern backend technologies such as PHP, Laravel, Java, Spring Boot, Node.js, Python, etc is compatible to MySQL. In this project, MySQL integration with Spring Boot via Spring Data JPA which allows object-relational mapping (ORM) (Emon, 2024).

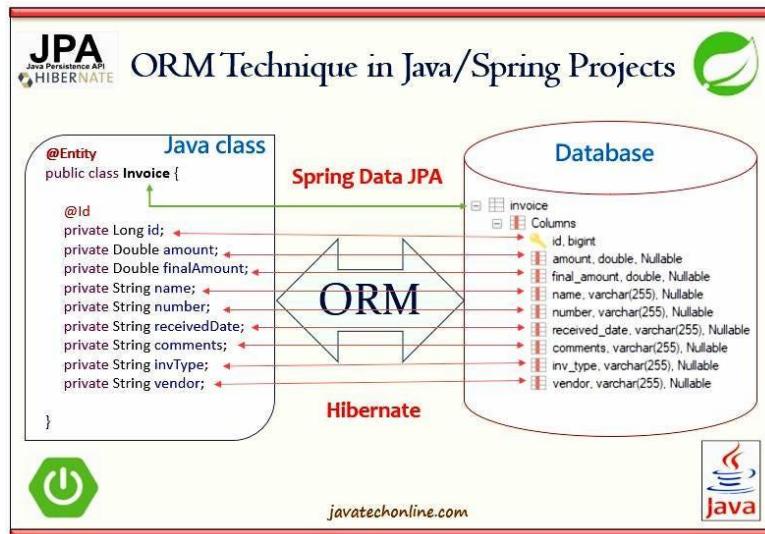


Figure 2.4 ORM Technique in Java Spring Projects (Emon, 2024)

2.5.4 React Redux

The official Redux UI binding library for React is called React Redux (Abramov and Redux documentation, 2021).

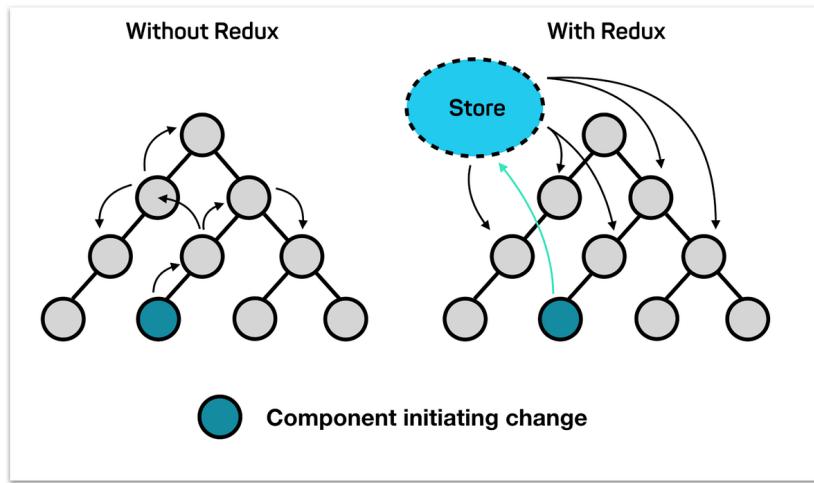


Figure 2.5 With and Without Redux in DOM Tree (Corman, 2020)

Redux offers a method of storing all the state apps in one place, or store, in React. When Redux is not used, the data or state flow in figure 2.5 becomes a two-way or direct component-to-component communication. It will lead to spaghetti code, which complicates the comprehension of the code (Westfall, 2017), and will complicate the data flow tracking. Redux offers state management through the use of unidirectional flow of data between parent and child node or component. The react component will either access the updated data state of the store using the Selector method or the dispatch method to update the data state to store and subscribe.

2.5.5 GitHub

GitHub offers cloud-based services for hosting Git repositories. The system code is stored and managed via this website and cloud-based service, which also uses version control to track and manage changes. Git makes the complete repository accessible on any developer's PC and enables code branching and merging (Jackson, 2023). Unbelievably, Git's user-friendly interface makes it possible for novice programmers to use it. Without GitHub, using Git typically just necessitates command-line expertise and a basic comprehension of technical concepts.

2.5.6 Visual Studio Code

A robust source code editor, Visual Studio Code is a free, small program that can be used online and on PCs. It comes with built-in support for Node.js, TypeScript, and JavaScript. Extensions that modify and improve Visual Studio Code can support additional languages and runtimes, including C++, Java, and Python (Microsoft, 2024). Furthermore, it includes capabilities of Visual Studio technology, such as graphical debugging, linting, multi-cursor editing, parameter hint, navigation, refactoring, and IntelliSense code completion for variables, methods, imported modules (Heller, 2022).

2.5.7 Enterprise Architect

Sparx Systems Enterprise Architect is a comprehensive modelling tool that helps software engineers, system architects, and business analysts to handle enterprise architecture, software development, and data modelling projects. It can deal with a variety of modelling standards and frameworks, such as UML, SysML, and BPMN, and thus is a flexible technology analysis and design tool.

The tool supports the complete software development process, including requirements gathering, analysis, and design modelling, testing, and maintenance, which makes it particularly appropriate to software developers (Sparx Systems Ltd & SparxSystems Software GmbH, 2024).

2.5.8 Postman

Postman is an Application Programming Interface (API) platform that is utilized in the development, testing and management of APIs. In addition to standard API testing, Postman offers numerous extra tools that simplify the entire API development process. It makes it possible to build real-world interactions and bogus

APIs, both of which are helpful for testing (Postman, 2021). Additionally, it offers robust testing features that allow one to perform various HTTP requests and find any issues before deployment. Postman is essential for developing effective and thoroughly documented APIs for my system because of all these designs and testing tools.

2.5.9 Jira

Atlassian, an Australian software business, is the developer of the Jira software application. It is recommended for the development stage for team that adopted agile methodology. Additionally, it helps several groups, such as marketing, human resources, and software developer by providing them the ability to automate workflow, report performance, release projects, plan projects, and monitor difficulties. The software provides Scrum, kanban, bug tracking, and DevOps (Atlassian, 2024).

2.6 Chapter Summary

This chapter forms the basis of developing Inventory Management System of Kelab Fotokreatif (IFoto), a web based centralized inventory management system. It starts by examining two industry systems that are Rentman and Cheqroom. Rentman that organize rentals and event productions on a single platform and work on projects, tasks, and make sure the right information reaches the right people. Cheqroom is an equipment management software that helps teams that share assets across people, projects, and places. A comparison table highlights how Inventory Management System for Kelab Fotokreatif (IFoto) stacks up against these competitors in terms of features like external equipment rentals online payment gateway, multi-role access, real-time availability tracking, event-based equipment assignment and target audience and purpose. Then, the chapter goes into details of the 9 technologies selected to create IFoto. ReactJS is a JavaScript framework that is employed on the frontend side to create the user interface efficiently. React Redux supports the state management in the React application. The SQL database used is MySQL because it is scalable, performs well, flexible and reliable in processing data. Spring Boot is an open-source Java-based framework and extensive library that forms the basis of back-end development and integration between the frontend and the backend is achieved through the use of RESTful APIs. The chapter concludes by outlining the software for project management and development tools that will be utilised. UML diagram will be done with Enterprise Architect. GitHub Desktop, a cloud-based service for hosting Git repositories, will help with version control. Jira will be the preferred tool for managing workflow, tracking issues, and planning projects. The development team will make use of the robust and portable code editor Visual Studio Code. The system's APIs will also be designed, tested, and documented with the help of the API platform Postman.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

The chapter deals with the system development methodology. The specific issues that were discussed in this chapter include the methodology adopted in the development of Inventory Management System of Kelab Fotokreatif (IFoto) and the rationale behind it. In addition to that, the phases of the methodology adopted have also been discussed and elaborated in this chapter, the technology adopted to develop the system and analyse the system requirements.

3.2 Phases of the Chose Methodology

The methodology that the project will be implementing is Hybrid Agile-Waterfall Methodology and it is chosen due to flexibility to customize the Methodology to my needs and it gives full control over the implementation (Żurawiecki, 2024). By leveraging hybrid methodology that combines both Waterfall and Agile methodology, it can minimize the disadvantages of both methods while preserving advantages of both methods (Shinde, 2025).

The Hybrid Agile-Waterfall Methodology has been divided into several phases which are requirement analysis, design, coding, testing and deployment. These phases are group based on the methodologies adopted in this hybrid methods in which requirement analysis and system design will be implemented as per Waterfall model while coding, testing and deployment will be implemented as per Agile model. Figure 3.1 shows the overview of Hybrid Agile-Waterfall Methodology.

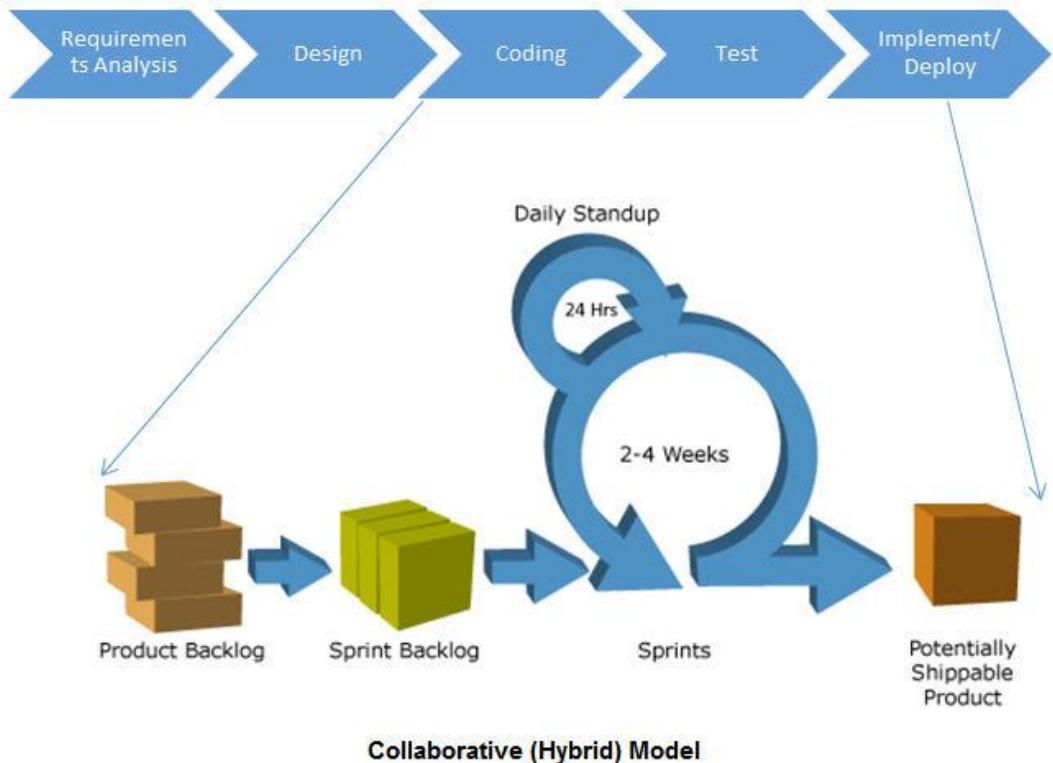


Figure 3.1 Hybrid Agile-Waterfall Methodology (Shinde, 2025)

3.2.1 Requirement Analysis

The first phase of the Waterfall model, requirement analysis, gathers and records all user demands, system functional requirements, non-requirement, and project constraints. Stakeholders are visited during this phase to grasp system goals and expectations. The aim is to produce an entirely comprehensive and unambiguous Software Requirements Specification (SRS) document that forms the basis for all upcoming development activity.

3.2.2 System Design

Following demand analysis comes system design, in which the architecture of the system is developed in line with the stated needs. Creating high-level, thorough design documentation including use case description, activity diagrams, sequence

diagram, database schemas, system architecture and detailed design falls under this stage. The purpose of this phase is to ensure the system meet the specified business requirement and non-functional needs such as scalability, maintainability, performance, etc.

3.2.3 Coding & Implementation

In this phase, coding and implementation are done iteratively and incrementally obeying the Agile methodology. The use cases are divided into sprints to be developed iteratively to ensure frequent releases, quick feedback loops, and continuous collaboration with stakeholders which are Kelab Fotokreatif's members, which allows the system evolves and adapt quickly according to user needs.

3.2.4 System Testing

During this phase, the system testing is done on each sprint continuously as new features are being developed that encourages quicker development. This stage involves unit testing, integration testing, system testing and user acceptance testing to identify defects and bugs early in the development process to reduce complexity of finding defects later in the development process since the more complex a system has become, the more difficult it is to find defects. Agile testing also concentrates on user-centred testing by working together to make sure that the end product is what the user wants and expects.

3.2.5 System Deployment

In this phase, the new features or updates are regularly and incrementally released in small batches to users, making it easier to gather feedback and adapt to changing requirements quickly. By being able to perform improvement quickly for

every sprint accelerates the delivery of high-quality system and achieve greater customer satisfaction.

3.3 Technology Used Description

The Inventory Management System of Kelab Fotokreatif (IFoto) has a lot of technologies selected to be constructed. Firstly, ReactJS is an open-source JavaScript framework and it is applied to develop frontend of a system in a short time and with much less written code and save time. React contains Redux UI binding library offers a method through which React can store all state apps in one place. Next, Spring Boot is an open-source Java-based framework that helps developers to build an independent, production-quality system applications by reducing complex configuration and boilerplate code (Walls, 2016). Additionally, RESTful APIs is an application programming interface (API) follows the principles of the Representational State Transfer (REST) architectural style and it is a way to enables communication between front-end and back-end application over the internet, allowing them to request and manage the data (Bigelow & Gillis, 2024). In addition to that, MySQL is a famous open-source Relational Database Management Systems (RDBMS) and it is administered and used through Structured Query Language (SQL). Unified Modelling Language (UML) modelling and design will be done using Enterprise Architect. GitHub installed in Visual Studio Code, a cloud-based Git repository hosting service, will be used to facilitate version control. During system implementation and development, Jira will be used as the tool of choice in project planning, issue tracking, and workflow management. In addition to that, a lightweight and powerful integrated development environment (IDE) or code editor, Visual Studio Code will be used to write all the source code of IFoto. Also, Postman, an API platform, will help design, test, and document APIs of the system.

3.4 System Requirement Analysis

The hardware or software must have the capacity to match the performance of the system in this web-based application. This section will describe the hardware and software requirements and the requirements to make the system capable of functioning optimally.

3.4.1 Software Requirements

The table 3.1 below shows the software requirements of Inventory Management System of Kelab Fotokreatif.

Table 3.1 Software Requirements for IFoto

Software Requirements		
Computer/Laptop		
Software		Requirement Details
Google Chrome		Version 80 or later
Microsoft Edge		Version 80 or later
Safari		Version 12 or later
Visual Studio Code		Version 1.60 or later
Programming Languages	Java	JDK 17 or later
	JavaScript (ES6+)	Any modern browser
Frameworks	Spring Boot	Version 2.7.x or 3.x (preferably with JDK 17 support)
	ReactJS	Version 18 or later
MySQL		Version 8.0 or later
GitHub		access via browser or Git CLI v2.30 or later
Smartphone		

Software	Requirement Details
Google Chrome on Android	Version 80 or later
Safari on IOS	Version 13 or later

3.4.2 Hardware Requirements

The table 3.2 below shows the minimum hardware requirements of Inventory Management System of Kelab Fotokreatif. A system specification that is superior to the one mentioned below can run IFoto without any problem.

Table 3.2 Hardware Requirements for IFoto

Software Requirements	
Computer/Laptop	
Software	Requirement Details (Minimum Requirements)
Operating System	Windows 7 (64-bit) or later version
	macOS 10.15 (Catalina) or later version
	Ubuntu 20.04 LTS or equivalent or later version
Processor	Intel Core i3 (8th Gen or later) or higher
	AMD Ryzen 3 (3000 series or later) or higher
	Apple M1 or M2 (for macOS users)
RAM	8 GB
Smartphone	
Software	Requirement Details (Minimum Requirements)
Operating System	Android 10 (Q) or newer
	iOS 13 or newer
Processor	Snapdragon 665 / Exynos 9611 / Equivalent or higher
	Apple A11 Bionic (iPhone 8) or higher

RAM	4 GB
-----	------

3.5 Chapter Summary

In summary, this chapter has addressed the selected system development methodology that is the Hybrid Agile-Waterfall Methodology. The stages, namely requirement analysis, design, coding, testing, and deployment were discussed in detail. Also, the technologies that will be employed in the development of the system like MySQL, Visual Studio Code, Spring Boot, Java, ReactJS, and many others were outlined. This chapter also represented the software requirement and hardware requirement analysis.

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

Chapter 4 introduction is concerned with the requirement analysis and architectural design of Inventory Management System of Kelab Fotokreatif (IFoto). It describes the key functions of the IFoto in a use case diagram that illustrates the interactions of four key actors namely User, Equipment Executive Committee, Equipment Committee Member and Administrator with their corresponding assigned use cases or functions. The system has fifteen use cases that support different modules like equipment management module, role and event management module, equipment requests and rental module as well as authentication and authorization module with the aim of simplifying the inventory management process to all actors.

4.2 Requirement Analysis

The requirement analysis is composed of use case diagram, use case description, activity diagram and sequence diagram to describe the business logic of the use case. The Figure 4.1 below illustrates the general use case diagram of the Inventory Management System of Kelab Fotokreatif (IFoto).

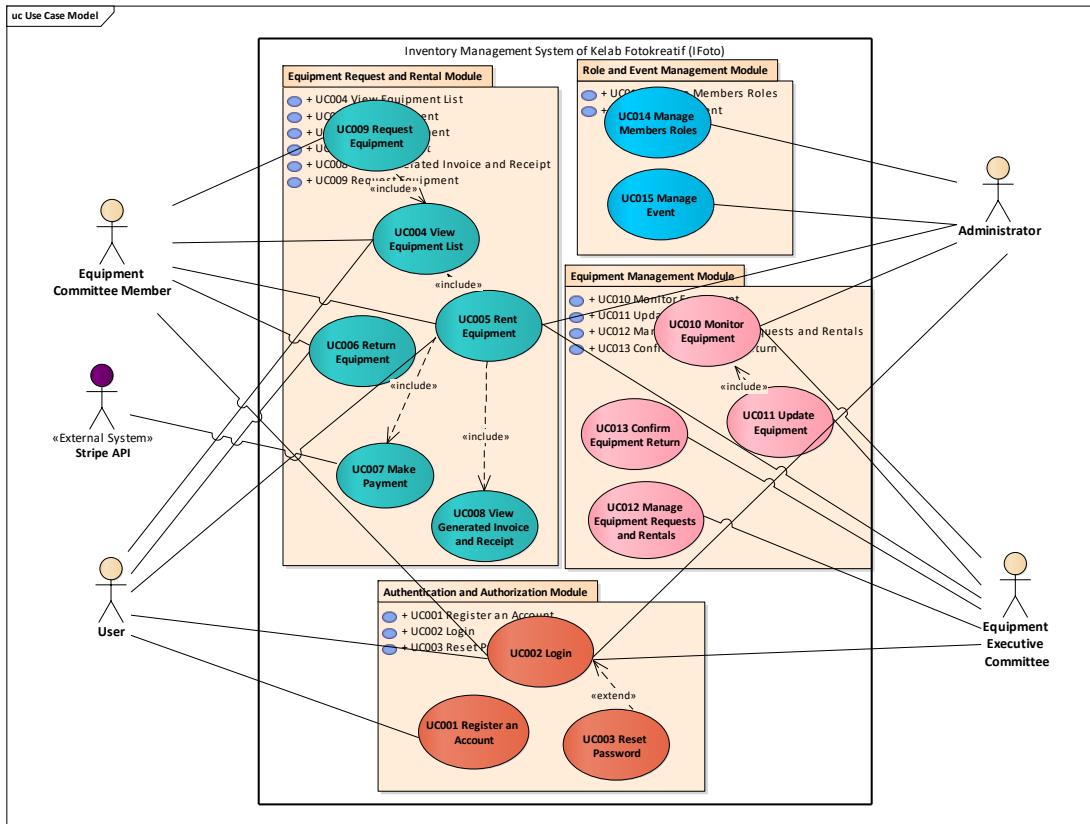


Figure 4.1 Use case diagram of Inventory Management System for Kelab Fotokreatif (IFoto)

The above use case diagram has four system actors namely the User, Equipment Executive Committee, Equipment Committee Member and Administrator. The system has mainly fifteen use cases that gives a high-level description of the functionalities provided by the IFoto to the actors, such as, equipment management, role and event management, equipment request and rental and authentication and authorization functionalities.

Table 4.1 Description of Use Cases for IFoto

Use case	Function	Description
UC001	Register an Account	This use case allows User to sign up as a normal user by providing personal information and credentials.
UC002	Login	This use case enables registered User to log in to the app using their username and password.

UC003	Reset Password	This use case allows User who forgot their password to reset it via email verification.
UC004	View Equipment List	This use case displays a list of all available equipment details such as name, quantity, status, and condition, allowing User and Equipment Committee Member to view it.
UC005	Rent Equipment	This use case allows any user to fill in the Club's Equipment Rental Form digitally to request for equipment formally.
UC006	Return Equipment	This use case facilitates the return process of rented by User and borrowed equipment by the Equipment Committee Member and update the inventory.
UC007	Make Payment	This use case allows User to make payments for equipment rentals through integrated payment gateways.
UC008	View Generated Invoice and Receipt	This use case allows system view the invoice generated after equipment Rent has been approved and view the receipt after payment is made.
UC009	Request Equipment	This use case allows Equipment Committee Member to request equipment in advance for planned events or activities.
UC010	Monitor Equipment	This use case enables the Equipment Executive Committee and Administrator to track equipment availability, status, and conditions.
UC011	Update Equipment	This use case allows Executive Equipment Committee to add, edit, or delete equipment details in the system.
UC012	Manage Equipment Requests and Rentals	This use case enables the Equipment Executive Committee to approve, reject, or manage ongoing rentals and requests.
UC013	Confirm Equipment Requests	This use case allows the Equipment Executive Committee to confirm the return of borrowed equipment after verifies its conditions and completeness.

UC014	Manage Members Roles	This use case allows Administrator to assign or revoke roles of the club members within the system.
UC015	Manage Event	This use case allows Administrator to create, update, and manage events that involve assignment of roles.

Then, the uses cases mentioned in Table 4.1 are described with the help of use case description, activity diagram and sequence diagram. Use case description or specification is a description of the functionality offered by the system. It includes the actor, pre-conditions, business flow, and post-conditions of the use case. In addition to that, activity diagram is used to show the flow of various activities that is the flow of business as mentioned in the use case description. Sequence diagram describes the way the technical functions are performed and the flow of data between the objects of classes in the system. The sequence is normally concealed to the stakeholder because it is full of technical jargon. The example of the use case description, activity diagram and sequence diagram of UC012 Update Equipment is presented below.

Table 4.2 Use Case Description for UC012 Update Equipment

Use Case ID	UC012
Use Case Name	Update Equipment
Description	This use case allows Equipment Executive Committee to add, edit, or delete equipment details in the system.
Actor(s)	Equipment Executive Committee.
Pre-condition(s)	<ol style="list-style-type: none"> 1. The Equipment Executive Committee is logged into the system. 2. The user has been assigned as Equipment Executive Committee by Administrator.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The Equipment Executive Committee navigates to the “Equipment Management” page. 2. The system displays the list of all equipment (UC011).

	<p>3. The Equipment Executive Committee can choose to add, edit, and delete equipment.</p> <ul style="list-style-type: none"> a. Add New Equipment (AF1) b. Edit Existing Equipment (AF2) c. Delete Equipment (AF3) <p>4. The system validates and applies the changes.</p> <p>5. The system displays “Update Successful” message and updates the list accordingly.</p>
Alternative Flow(s) - AF	<p>AF1. Add New Equipment</p> <ol style="list-style-type: none"> 1. Equipment Executive Committee selects the “Add Equipment” option. 2. The system displays a form for entering equipment details (name, category, quantity, condition, image, etc.). 3. The Equipment Executive Committee fills in the form and submits. <p>AF2. Edit Existing Equipment</p> <ol style="list-style-type: none"> 1. The Equipment Executive Committee selects the “Edit” option for a specific equipment item. 2. The system displays the existing details in an editable form. 3. The Equipment Executive Committee modifies the necessary fields and submits. <p>AF3. Delete Equipment</p> <ol style="list-style-type: none"> 1. The Equipment Executive Committee selects the “Delete” option for a specific equipment item. 2. The system asks for confirmation. 3. The Equipment Executive Committee click “Confirm.”
Exception Flow(s) - EF	<p>EF1. Deletion of Assigned Equipment</p> <ol style="list-style-type: none"> 1. The system displays warning message, “This item cannot be deleted until it is returned.”
Post-condition(s)	Equipment information is updated in the system database.

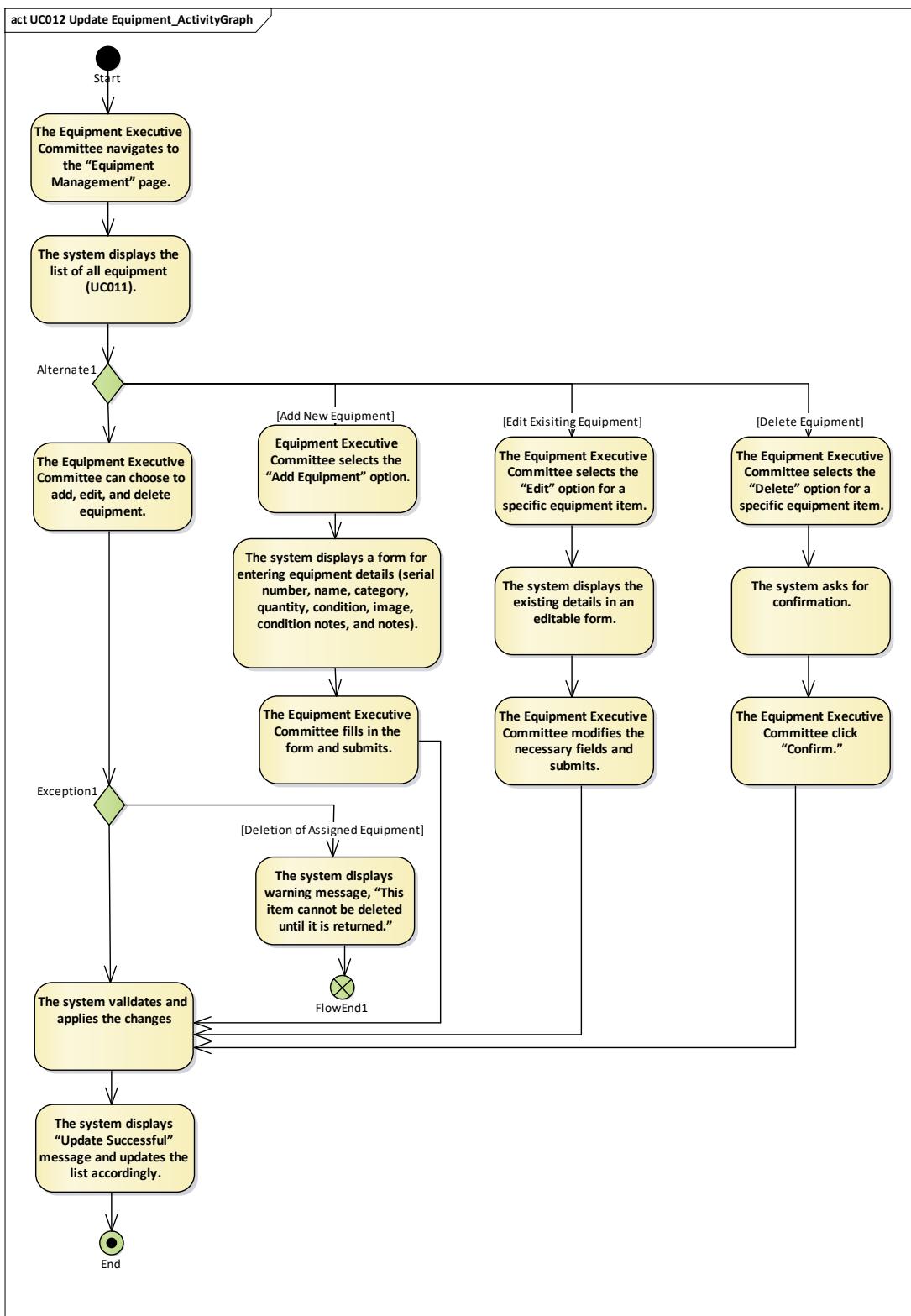


Figure 4.2 Activity Diagram for UC012 Update Equipment

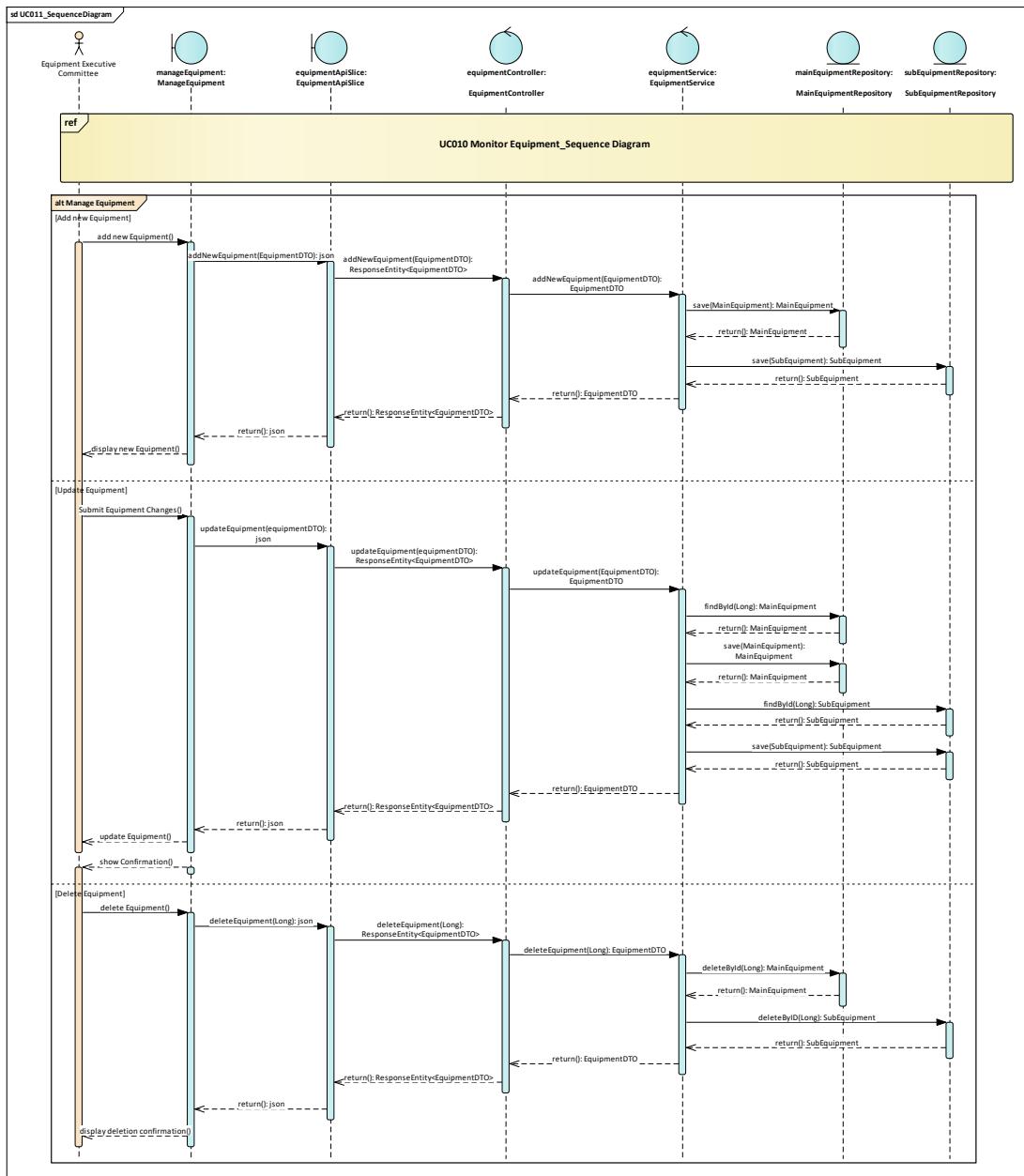


Figure 4.3 Sequence Diagram for UC012 Update Equipment

4.3 Project Design

4.3.1 System Architectural Style and Pattern

The architecture pattern and style adopted to the Inventory Management System of the Kelab Fotokreatif (IFoto) is Model-View-Controller (MVC) based on 3-tier architecture. This architecture is meant to divide the system into three levels. According to Figure 4.2, the View that is represented by the User Interface layer (React frontend), this layer oversees the user interface rendering and processing user interactions. It sends and receives data to and from the backend using API calls. Then, the Controller depicted by Business Logic layer consists of such components as Controllers, DTOs, Services, Repositories, and Models. This layer accepts the requests of the user via the REST endpoints, enforces the business rules, and formats the response in terms of DTOs to transfer the data. Lastly, the Model that is denoted by the Data Access Layer (MySQL database) stores and retrieve data. JPA Repository manages database and interacts with the Business Logic Layer via the JPA Entities to offer data persistence. These layers offer clear separation of concerns, leading to maintainability, scalability, and flexibility. Package Diagram in Figure 4.3 shows a much more details for each component in the 3-Tier Architecture where Package Frontend consist of User Interface Layer while Package Backend consist of Domain Layer and Data Access Layer.

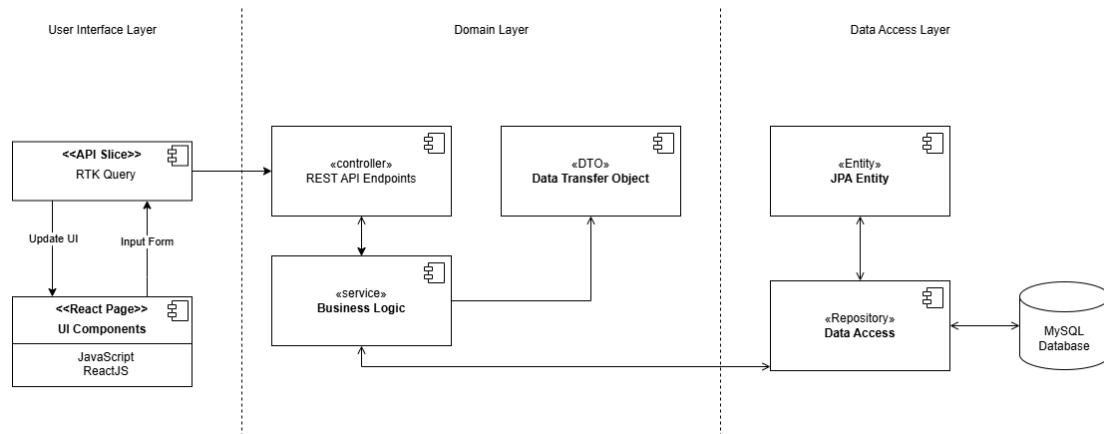


Figure 4.4 Architecture Model of 3-Tier Architecture in Inventory Management System for Kelab Fotokreatif (IFoto)

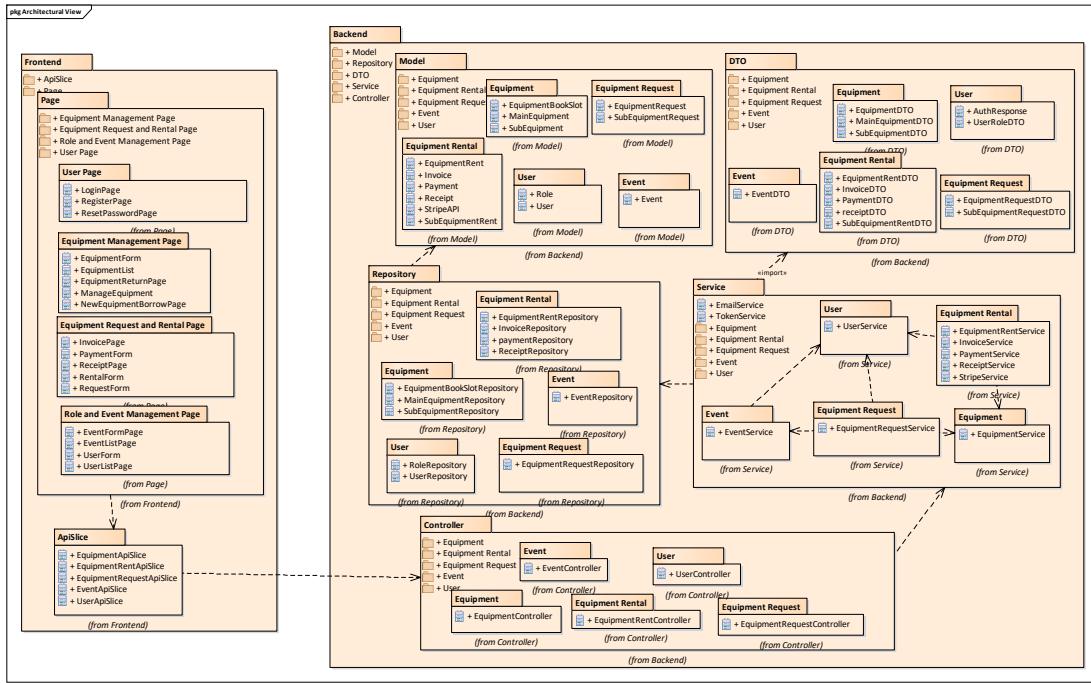


Figure 4.5 Package Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)

4.3.2 Complete System Design and its Subsystem

This subtopic also shall discuss about subsystem of IFoto and more details on architectural style and pattern leveraged for this system. Package Diagram in Figure 4.6 illustrate the overview of Inventory Management System for Kelab Fotokreatif (IFoto). This system consists of four main modules which are Equipment Management Module, Authentication and Authorization Module, Equipment Request and Rental Module, and Role and Event Management module. Each of this module follows the Model-View-Controller (MVC) architectural style and pattern where it consists of Page and APISlice as View, Controller, Service and Data Transfer Object (DTO) as Controller and Repository and Model (Entity) as Model of the system.

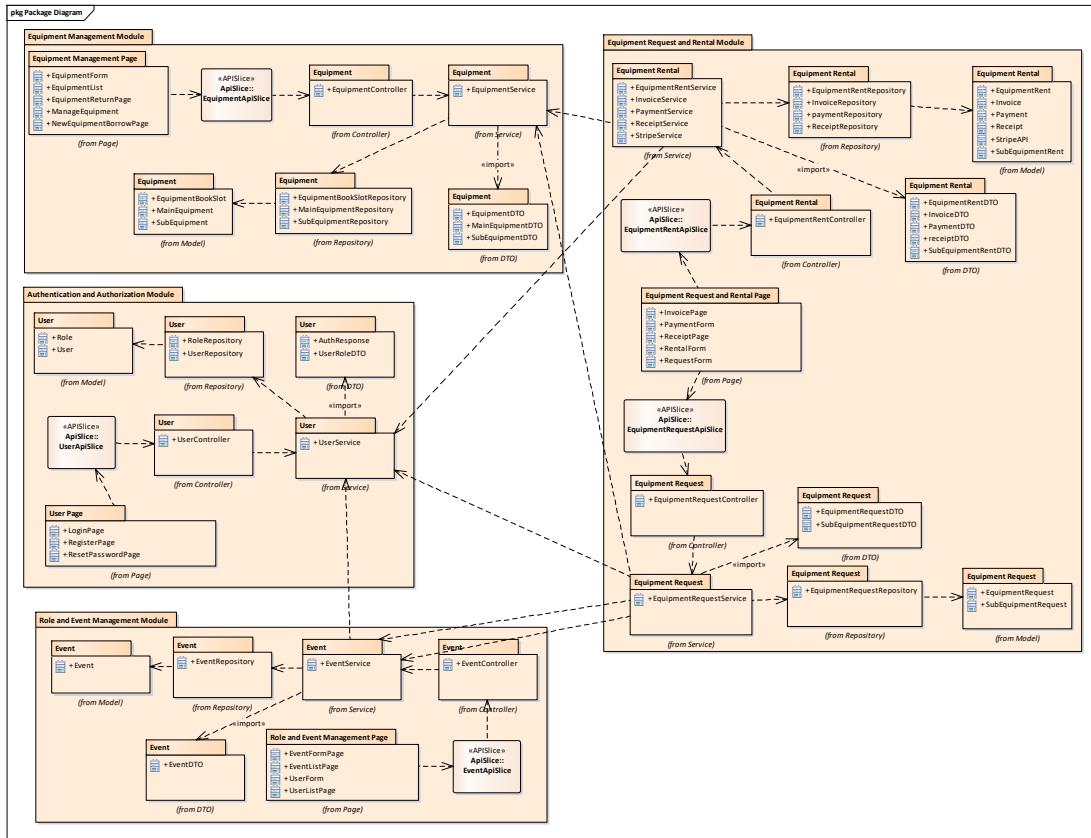


Figure 4.6 Package Diagram for Inventory Management System for Kelab Fotokreatif (IFoto).

Figure 4.7 represents the complete Backend of class diagram of IFoto and Figure 4.8 represents Frontend of Component Diagram for IFoto. Figure 4.9 until Figure 4.12 represents class diagram of the subsystem and each has independent purposes. Class Diagram also shows the relationship between these classes and subsystem and how they work together to achieve the stakeholder business requirement. For each subsystem of Class Diagram, there are multiple classes with distinct stereotype and functions that work together. These separation of concern or modularity is based on MVC architectural design which is crucial to develop a maintainable, testable, scalable, and reliable system.

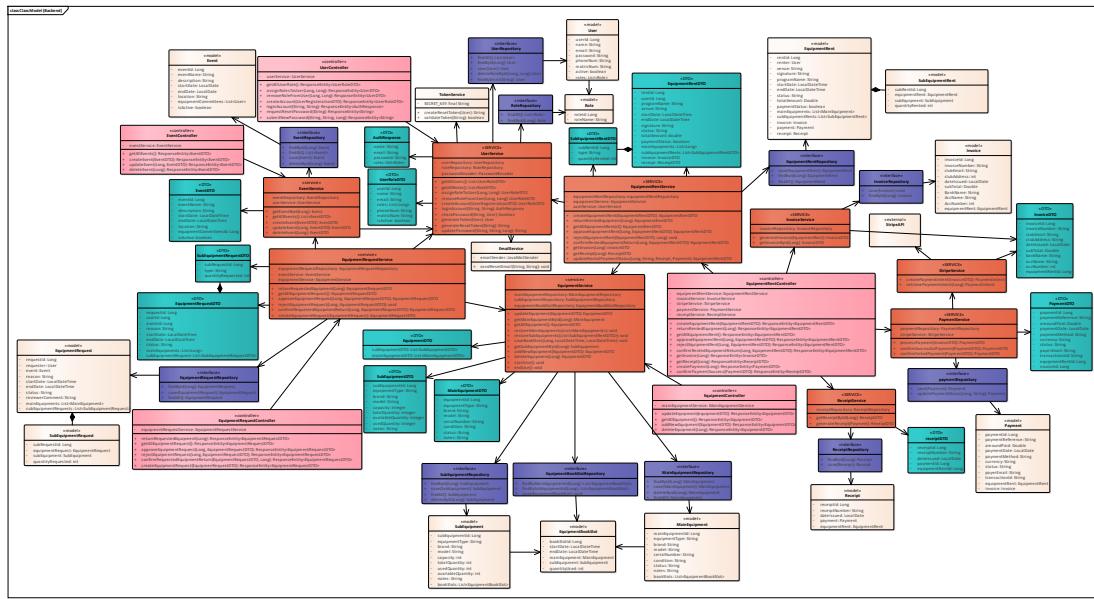


Figure 4.7 Complete Backend Class Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)

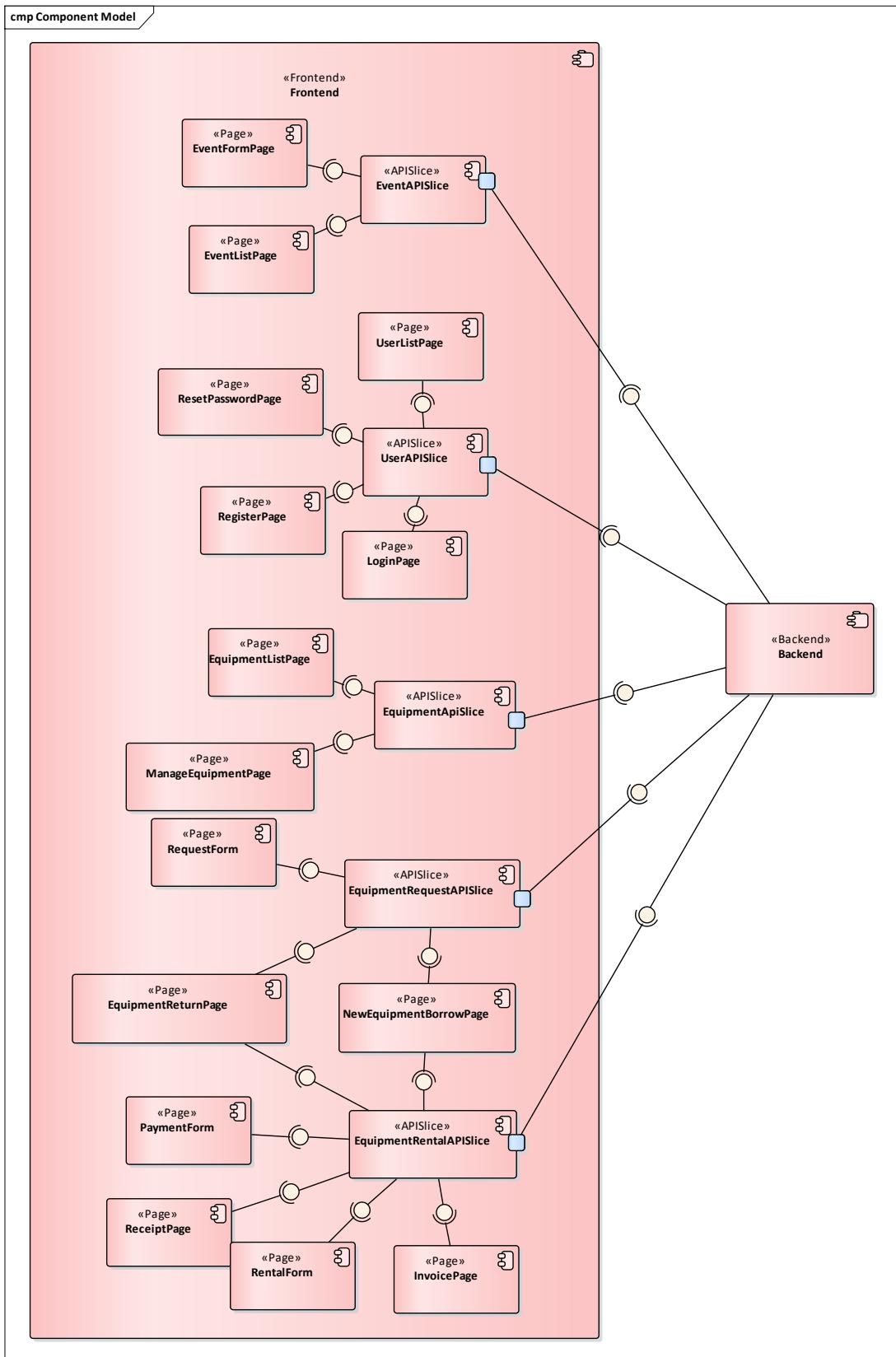


Figure 4.8 Frontend Component Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)

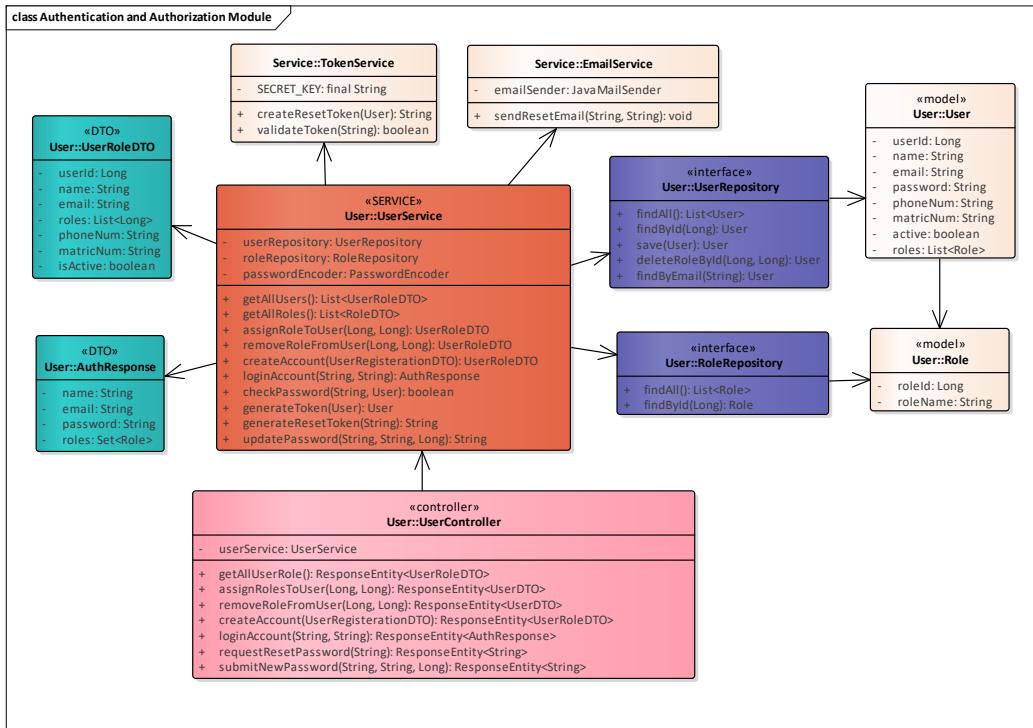


Figure 4.9 Class Diagram for Authentication and Authorization Module

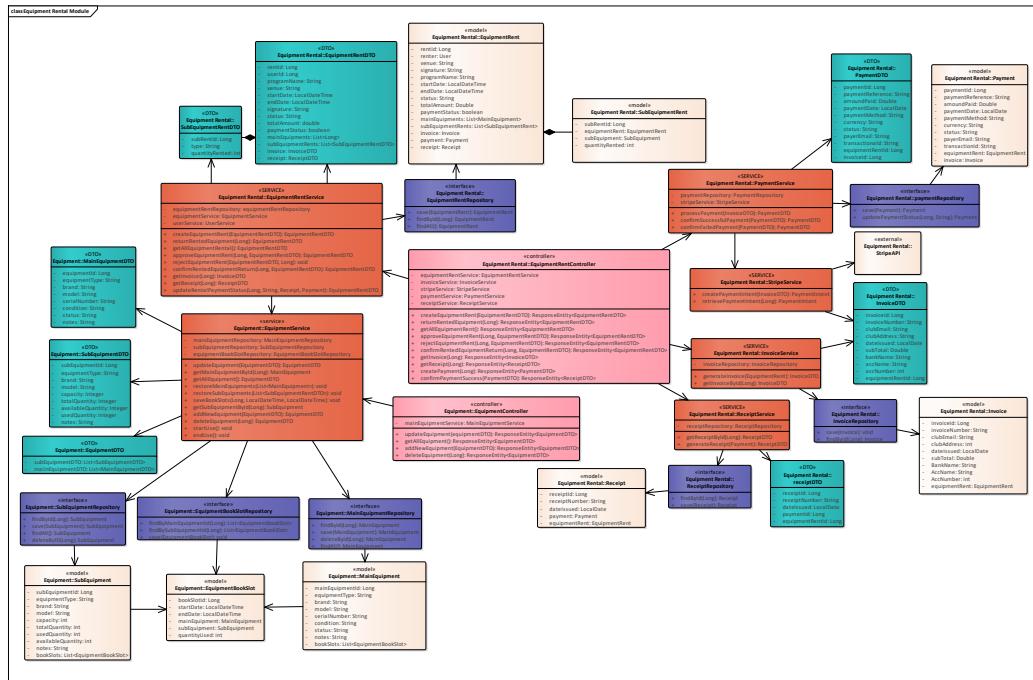


Figure 4.10 Class Diagram for Equipment Request and Rental Module (Equipment Rent)

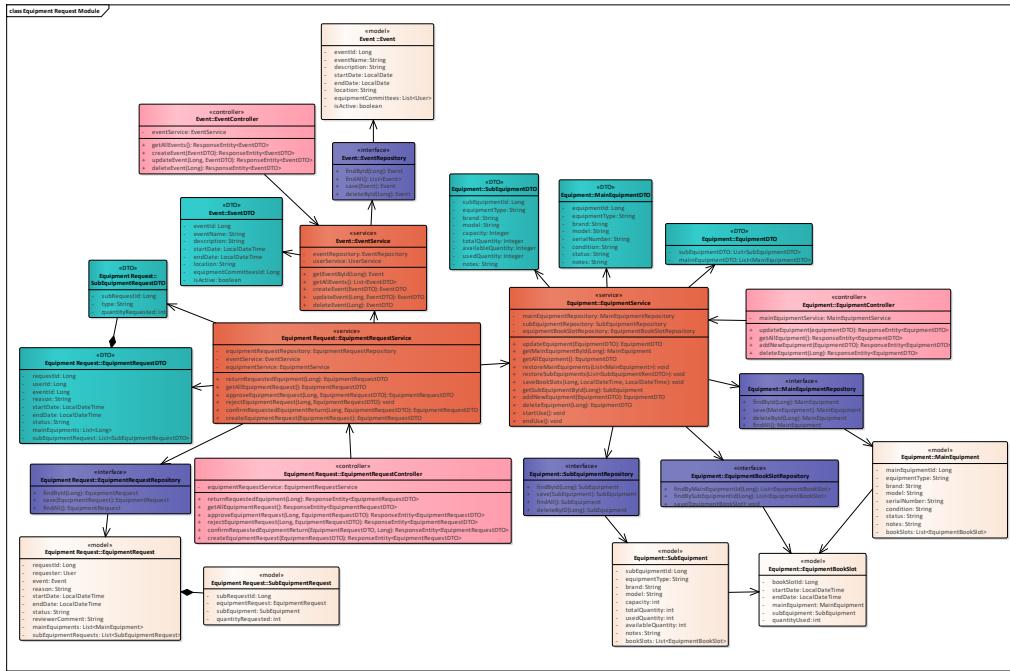


Figure 4.11 Class Diagram for Equipment Request and Rental Module (Equipment Request)

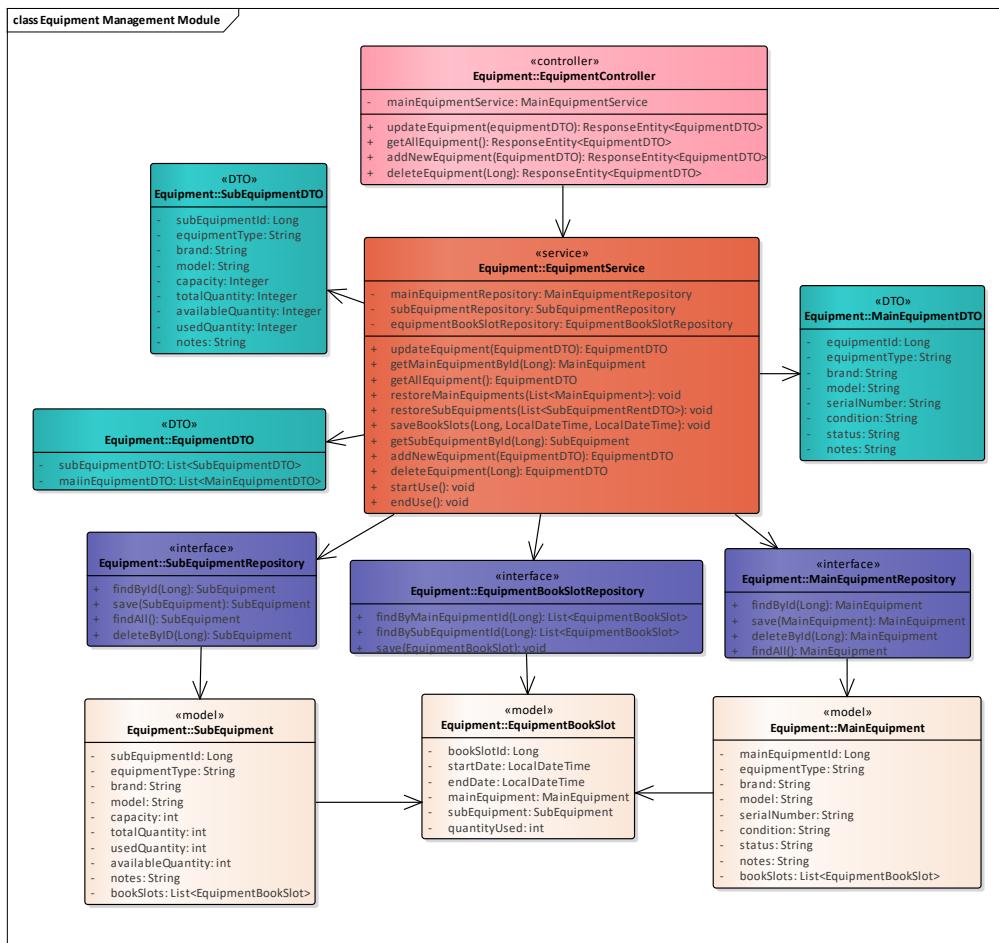


Figure 4.12 Class Diagram for Equipment Management Module

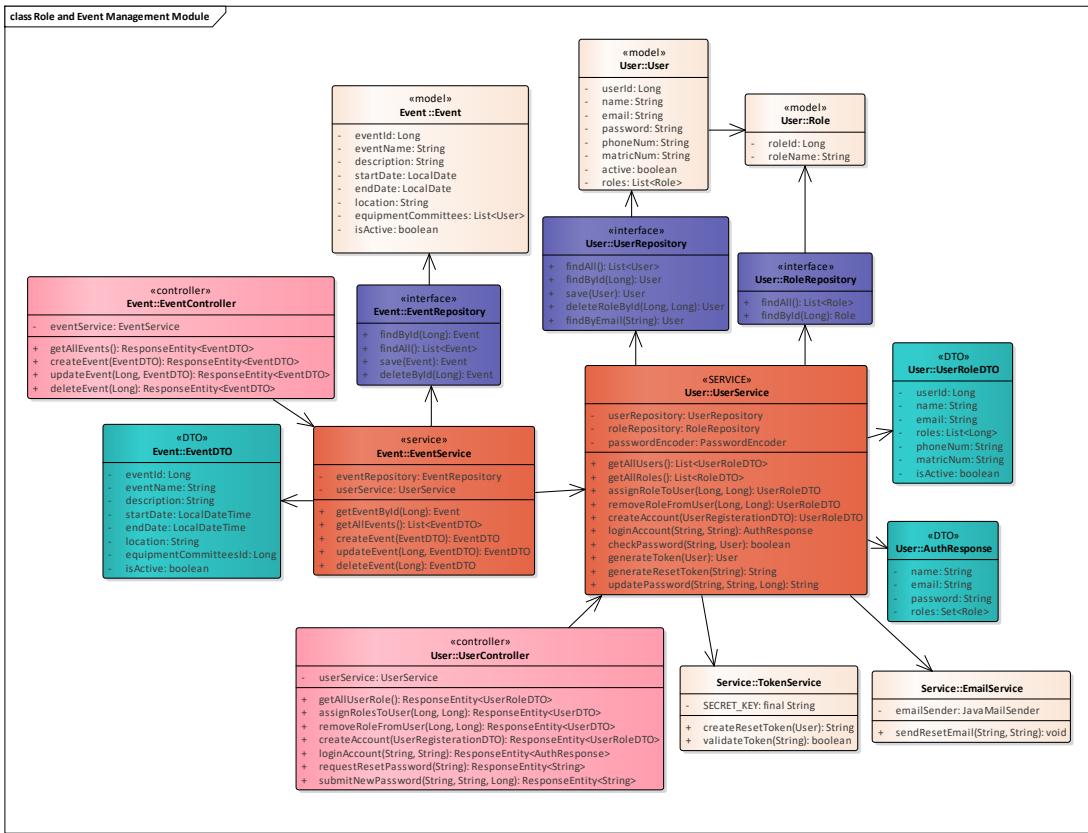


Figure 4.13 Class Diagram for Event Management Module

IFoto is based on Model-View-Controller (MVC) architectural pattern that divides the system into three interrelated parts that include Model, View, and Controller. Model is the data and logic of the system and it has access to the database and can manipulate it. Then, View is the frontend of the system that presents the required information in a particular form. Finally, Controller is a mediator between Model and View that processes the business process of the system.

Based Frontend Component Diagram in Figure 4.8, View consists of two stereotypes which are Page and APISlice stereotype. Page is responsible to display necessary information and interact with user with its user-friendly interface. Page will interact with APISlice that is responsible to provide a way to integrate with API endpoints to fetch and send data from and to backend. The examples of Page components are ManageEquipmentPage, RentalForm and RequestForm. The examples of APISlice components are EquipmentApiSlice, EquipmentRentApiSlice, EquipmentRequestApiSlice.

Next, Model consist of two stereotypes which are Model (Entity) and Repository. Model is a class that defines the schema of database table while Repository represents an interface that extends from JPRepository used to perform CRUD operations such as save(), findAll(), findById() and deleteById(). The examples of Model classes are User, Role, EquipmentRent, MainEquipment, SubEquipment and EquipmentRequest. The examples of Repository classes are UserRepository, RoleRepository, EquipmentRepository, EquipmentRentRepository and EquipmentRequestRepository.

Lastly, Controller consists of three stereotypes which are controller, service, and Data Transfer Object (DTO). Controller is set of API endpoints that clients or frontend use to interact with rest of the classes in backend. The data is forward to service layer that serves as bridge between controller and repository. Service layer also contains business logic and handles the mapping between DTOs and Models. DTOs is used to transfer data between layers and prevent exposing raw data from entity to promote better security and flexibility. The examples of controller classes are UserController, EventController and EquipmentRentController. The examples of service classes are EventService, UserService and EquipmentRentService. The examples of Data Transfer Object (DTO) classes are UserRoleDTO, EquipmentRentUserDTO, EquipmentRentDTO and SubEquipmentRentDTO.

4.4 Database Design

The database that this system will use is based on a Relational Database Management System (RDBMS) that is MySQL that manages data using Structured Query Language (SQL). The Entity Relation Diagram (ERD) of IFoto is shown in figure 4.12 below and the tables below represents the data dictionaries of entities with column of entity names, field names, data type, description.

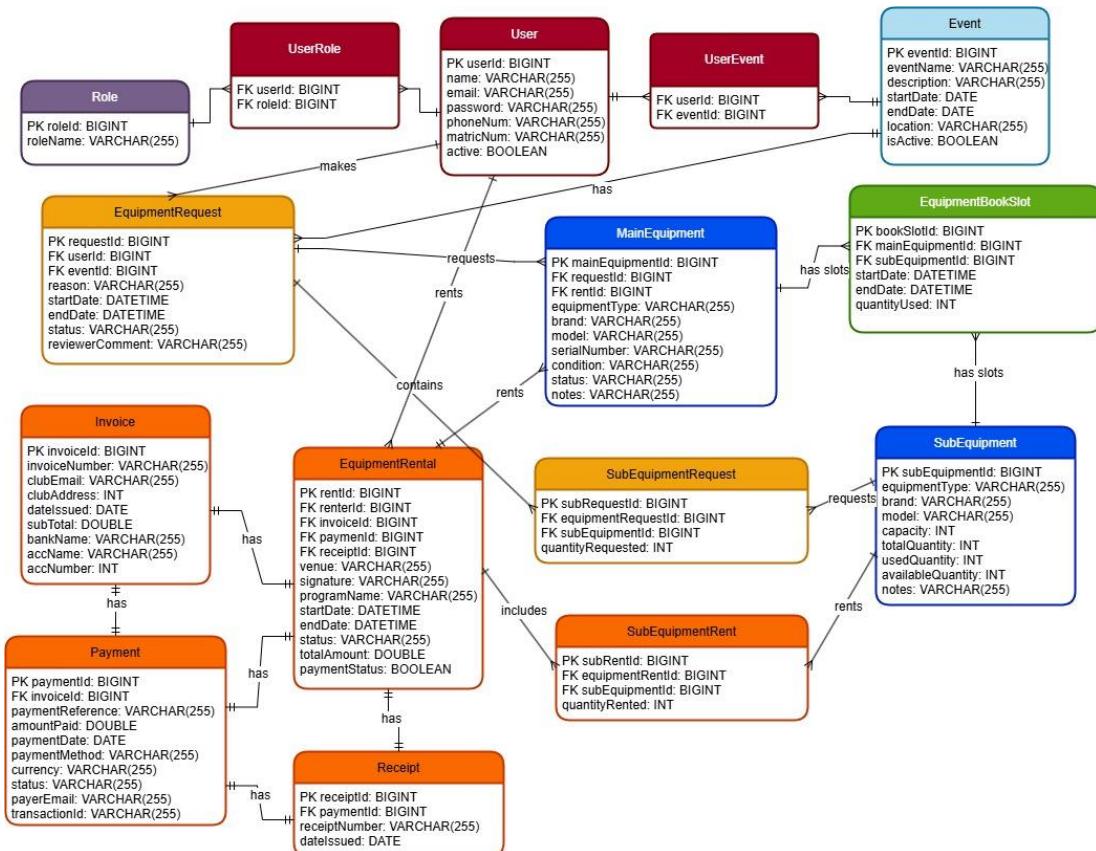


Figure 4.14 Entity Relationship Diagram (ERD) of Inventory Management System for Kelab Fotokreatif (IFoto)

Table 4.3 Data Dictionary for Inventory Management System for Kelab Fotokreatif (IFoto).

Entity Name	Field Name	Data Type	Description
Role	roleId	BIGINT	Primary key, unique identifier for a role

	roleName	VARCHAR(255)	Name of the role
User	userId	BIGINT	Primary key, unique identifier for a user
	name	VARCHAR(255)	Full name of the user
	email	VARCHAR(255)	Email address of the user
	password	VARCHAR(255)	Password for user authentication
	phoneNum	VARCHAR(255)	Phone number of the user
	matricNum	VARCHAR(255)	Matriculation number of the user (e.g., student ID)
	active	BOOLEAN	Indicates if the user account is active
UserRole	userId	BIGINT	Foreign key, references User.userId
	roleId	BIGINT	Foreign key, references Role.roleId
UserEvent	userId	BIGINT	Foreign key, references User.userId
	eventId	BIGINT	Foreign key, references Event.eventId
Event	eventId	BIGINT	Primary key, unique identifier for an event
	eventName	VARCHAR(255)	Name of the event
	description	VARCHAR(255)	Description of the event

	startDate	DATE	Start date of the event
	endDate	DATE	End date of the event
	location	VARCHAR(255)	Location where the event takes place
	isActive	BOOLEAN	Indicates if the event is active
EquipmentRequest	requestId	BIGINT	Primary key, unique identifier for an equipment request
	userId	BIGINT	Foreign key, references User.userId
	eventId	BIGINT	Foreign key, references Event.eventId
	reason	VARCHAR(255)	Reason for the equipment request
	startDate	DATETIME	Start date and time of the request period
	endDate	DATETIME	End date and time of the request period
	status	VARCHAR(255)	Status of the equipment request
	reviewerComment	VARCHAR(255)	Comments from the reviewer
	mainEquipmentId	BIGINT	Primary key, unique identifier for main equipment
	requestId	BIGINT	Foreign key, references EquipmentRequest.requestId
	renterId	BIGINT	Foreign key, references

			User.userId
	equipmentType	VARCHAR(255)	Type of the equipment
	brand	VARCHAR(255)	Brand of the equipment
	model	VARCHAR(255)	Model of the equipment
	serialNumber	VARCHAR(255)	Serial number of the equipment
	condition	VARCHAR(255)	Condition of the equipment
	status	VARCHAR(255)	Status of the equipment
	notes	VARCHAR(255)	Additional notes about the equipment
SubEquipmentRequest	subEquipmentRequestId	BIGINT	Primary key, unique identifier for sub-equipment request
	equipmentRequestId	BIGINT	Foreign key, references EquipmentRequest.requestId
	subEquipmentId	BIGINT	Foreign key, references SubEquipment.subEquipmentId
	quantityRequested	INT	Quantity of sub-equipment requested
SubEquipment	subEquipmentId	BIGINT	Primary key, unique identifier for sub-equipment
	mainEquipmentId	BIGINT	Foreign key, references MainEquipment.mainEquipmentId

			mentId
	equipmentType	VARCHAR(255)	Type of the sub-equipment
	brand	VARCHAR(255)	Brand of the sub-equipment
	model	VARCHAR(255)	Model of the sub-equipment
	capacity	INT	Capacity of the sub-equipment
	totalQuantity	INT	Total quantity available
	usedQuantity	INT	Quantity currently in use
	availableQuantity	INT	Quantity available for use
	notes	VARCHAR(255)	Additional notes about the sub-equipment
EquipmentRental	rentalId	BIGINT	Primary key, unique identifier for equipment rental
	renterId	BIGINT	Foreign key, references User.userId
	paymentId	BIGINT	Foreign key, references Payment.paymentId
	receiptId	BIGINT	Foreign key, references Receipt.receiptId
	venue	VARCHAR(255)	Venue of the rental
	signature	VARCHAR(255)	Signature for the rental agreement
	startDate	DATETIME	Start date and time of the rental
	endDate	DATETIME	End date and time of the

			rental
	status	VARCHAR(2 55)	Status of the rental
	totalAmount	DOUBLE	Total amount for the rental
	paymentStatus	BOOLEAN	Indicates if payment is complete
SubEquipmentRent	subRentId	BIGINT	Primary key, unique identifier for sub-equipment rent
	equipmentRentId	BIGINT	Foreign key, references EquipmentRental.rentalId
	subEquipmentId	BIGINT	Foreign key, references SubEquipment.subEquipmentId
	quantityRented	INT	Quantity of sub-equipment rented
Payment	paymentId	BIGINT	Primary key, unique identifier for payment
	invoiceId	BIGINT	Foreign key, references Invoice.invoiceId
	paymentReference	VARCHAR(2 55)	Reference number for the payment
	amountPaid	DOUBLE	Amount paid
	paymentDate	DATE	Date of the payment
	paymentMethod	VARCHAR(2 55)	Method of payment
	currency	VARCHAR(2 55)	Currency of the payment
	status	VARCHAR(2 55)	Status of the payment

	payerEmail	VARCHAR(255)	Email of the payer
	transactionId	VARCHAR(255)	Transaction identifier
Invoice	invoiceId	BIGINT	Primary key, unique identifier for invoice
	equipmentRentId	BIGINT	Foreign key, references EquipmentRental.rentalId
	invoiceNumber	VARCHAR(255)	Invoice number
	clubEmail	VARCHAR(255)	Club email address
	clubAddress	VARCHAR(255)	Club address
	dateIssued	DATE	Date the invoice was issued
	subTotal	DOUBLE	Subtotal amount
	bankName	VARCHAR(255)	Name of the bank
	accName	VARCHAR(255)	Account name
	accNumber	INT	Account number
Receipt	receiptId	BIGINT	Primary key, unique identifier for receipt
	paymentId	BIGINT	Foreign key, references Payment.paymentId
	equipmentRentId	BIGINT	Foreign key, references EquipmentRental.rentalId
	receiptNumber	VARCHAR(255)	Receipt number
	dateIssued	DATE	Date the receipt was issued

EquipmentBookSlot	bookSlotId	BIGINT	Primary key, unique identifier for equipment book slot
	mainEquipmentId	BIGINT	Foreign key, references MainEquipment.mainEquipmentId
	subEquipmentId	BIGINT	Foreign key, references SubEquipment.subEquipmentId
	startDate	DATETIME	Start date and time of the booking slot
	endDate	DATETIME	End date and time of the booking slot
	quantityUsed	INT	Quantity of equipment used in the slot

4.5 Interface Design

Below figures represents few examples of the User Interface (UI) of the IFoto. The users of the system are User, Equipment Executive Committee, Equipment Committee Member and Administrator. The UI below is the first version of the design and are subject to changes towards the final deployment of the system.

4.5.1 User Interface for Equipment Request and Rentals Module

The screenshot shows the 'Rent Equipment' module of the IFoto system. At the top, there is a search bar labeled 'Search product, supplier, order'. On the right side, there are icons for a bell通知, a user profile, and a cart. Below the search bar, there are navigation links: 'Rent Equipment' (selected), 'Orders', 'Settings', and 'Log Out'. The main content area is titled 'Rent Equipment' and 'Equipment List'. It contains two tables: 'Camera' and 'Camera Lens'. Both tables have columns for No., Brand, Model, Condition, Status, Notes, and Action (with an 'Add' button). The 'Camera' table has 8 rows, and the 'Camera Lens' table has 3 rows. The 'Camera' table includes a 'Filters' button. The 'Camera Lens' table also includes a 'Filters' button.

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	5D Mark IV	Good	Available	-	Add
2	Canon	5D Mark IV	Good	Available	-	Add
3	Canon	5D Mark II	Broken	Unavailable	EU	Add
4	Canon	5D Mark II	Good	Available	-	Add
5	Canon	60D	Good	Available	-	Add
6	Nikon	D90	Broken	Unavailable	KBV	Add
7	Nikon	D90	Good	Available	-	Add
8	Nikon	D7000	Good	Available	-	Add

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Add
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	Add
3	Canon	Tamron	EF 70-200mm	Broken	Unavailable	EU	Add

Figure 4.15 Interface of List of Equipment (Rent Equipment)

Figure 4.16 Interface of Cart Page (Rent Equipment)

Figure 4.17 Interface of Rental Form (Rent Equipment)

Rental List

Equipment List > Rental List

Manage Equipment Rent

- FutureReady (ID123124123123)**
1A/Krighnarajapuram, 3 rd street sulur
Muhammad Taufiq
20/6/2025 - 25/6/2025 Status: **Approved (Unpaid)**
Pay Now **View Invoice** **View Receipt**
- CONVOCATION - 68 (ID123124123123)**
1A/Krighnarajapuram, 3 rd street sulur
Ahmad Hazim
23/6/2025 ~ 26/6/2025 Status: **Rejected**
Pay Now **View Invoice** **View Receipt**
- WEDDING (ID123124123123)**
1A/Krighnarajapuram, 3 rd street sulur
Nabil Iman
29/6/2025 ~ 1/7/2025 Status: **Approved (Paid)**
Return Equipment **View Invoice** **View Receipt**

Settings **Log Out**

Figure 4.18 Interface of List of Equipment Rents (Rent Equipment)

FutureReady details

Equipment List > Rental List > FutureReady

FutureReady

FutureReady (ID123124123123)
1A/Krighnarajapuram, 3 rd street sulur Total Amount: **RM 75.00**
Muhammad Taufiq
20/6/2025 - 25/6/2025 Status: **Approved**
Pay Now **Cancel** **View Invoice** **View Receipt**

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	5D Mark IV	Good	Available	-	Edit Remove
2	Canon	5D Mark IV	Good	Available	-	Edit Remove
3	Canon	5D Mark II	Good	Available	EU	Edit Remove

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Edit Remove

Settings **Log Out**

Figure 4.19 Interface of Specific Equipment Rent Details (Rent Equipment)

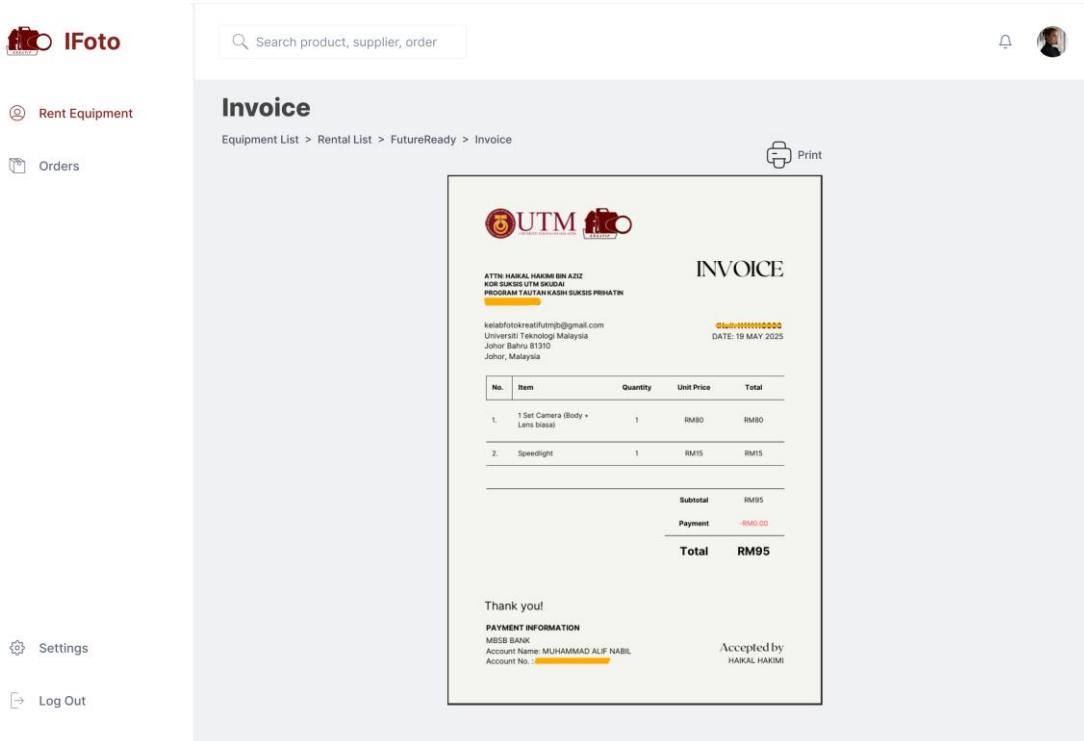


Figure 4.20 Interface of Invoice (Rent Equipment)

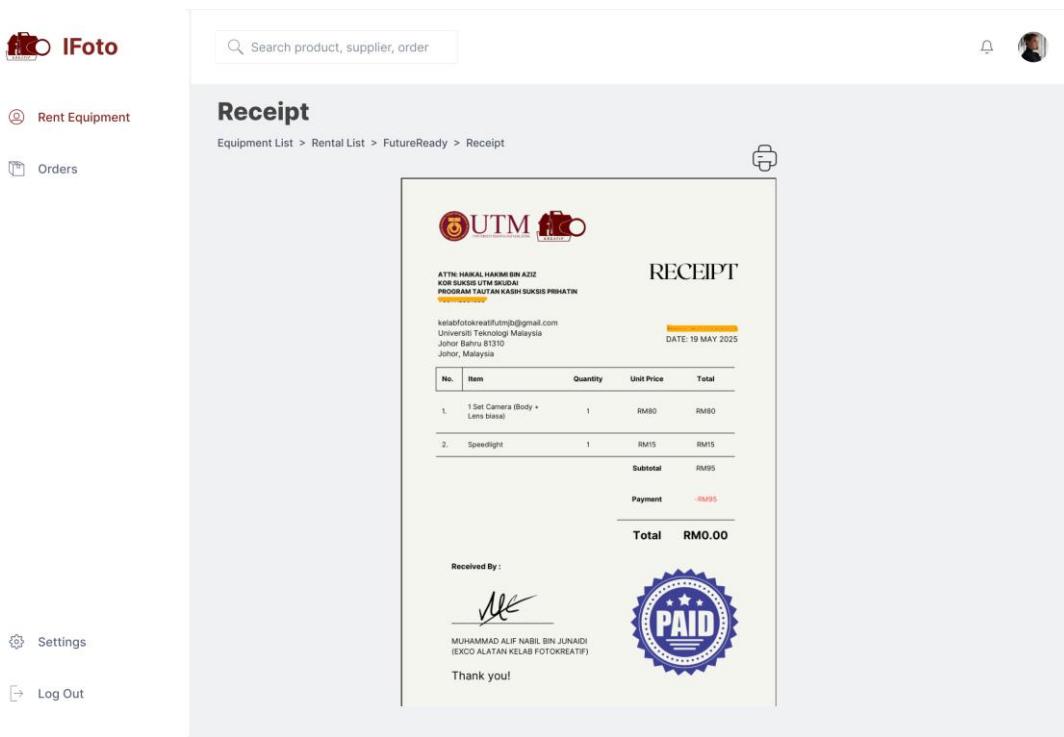


Figure 4.21 Interface of Receipt (Rent Equipment)

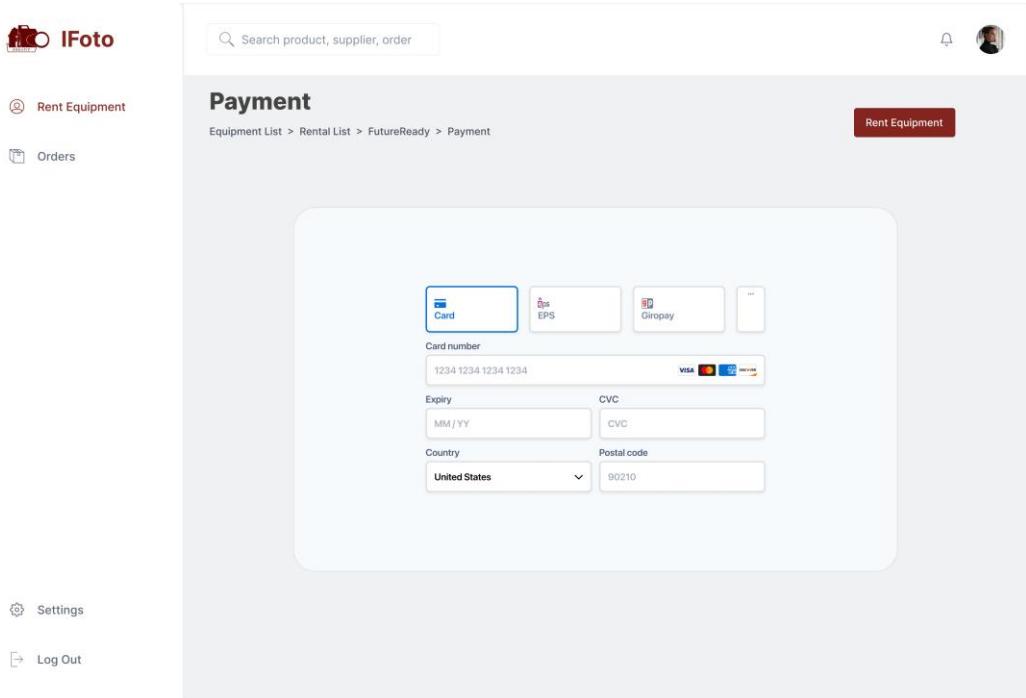


Figure 4.22 Interface of Payment Page (Rent Equipment)

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	5D Mark IV	Good	Available	-	<button>Edit</button> <button>Remove</button>
2	Canon	5D Mark IV	Good	Available	-	<button>Edit</button> <button>Remove</button>
3	Canon	5D Mark II	Good	Available	EU	<button>Edit</button> <button>Remove</button>

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	<button>Edit</button> <button>Remove</button>

Figure 4.23 Interface of Equipment Return Page (Rent Equipment)

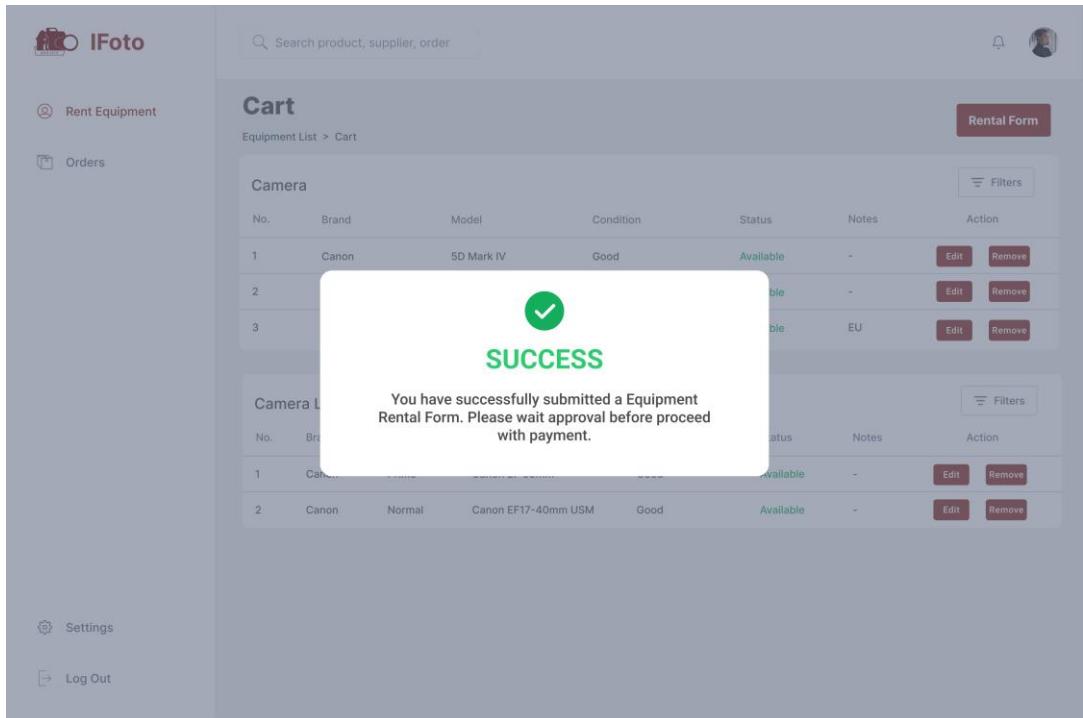


Figure 4.24 Example of Interface for Successful Action (Rent Equipment)

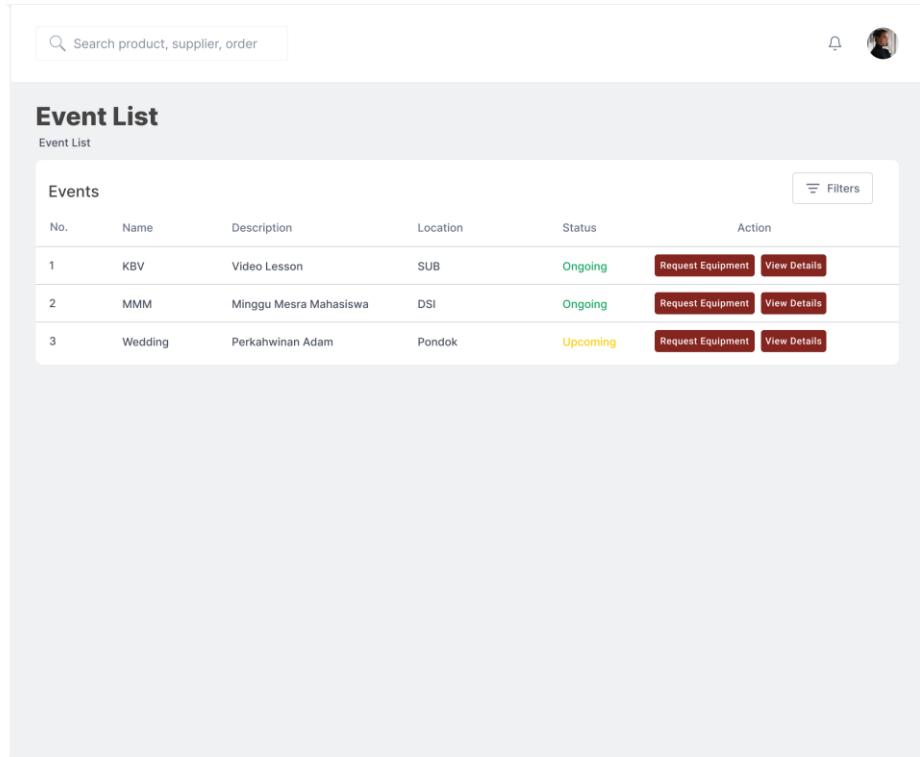


Figure 4.25 Interface of List of Event (Request Equipment)

The screenshot shows the 'Request Equipment for KBV' page. At the top, there is a search bar with the placeholder 'Search product, supplier, order'. To the right of the search bar are a bell icon and a user profile picture. Below the search bar, the title 'Request Equipment for KBV' is displayed, along with a breadcrumb navigation 'Event List > Request Equipment'. On the left side, there are several navigation links: 'Request Equipment' (highlighted in red), 'Orders', 'Settings', and 'Log Out'. The main content area is divided into two sections: 'Camera' and 'Camera Lens'. Both sections have a 'Filters' button in the top right corner. The 'Camera' section contains 8 items, and the 'Camera Lens' section contains 2 items. Each item row includes columns for No., Brand, Model, Condition, Status, Notes, and Action (with 'Add' buttons). The 'Camera' section has rows for Canon 5D Mark IV (Good, Available), Canon 5D Mark IV (Good, Available), Canon 5D Mark II (Broken, Unavailable), Canon 5D Mark II (Good, Available), Canon 60D (Good, Available), Nikon D90 (Broken, Unavailable), Nikon D90 (Good, Available), and Nikon D7000 (Good, Available). The 'Camera Lens' section has rows for Canon EF 50mm (Good, Available) and Canon EF17-40mm USM (Good, Available).

Figure 4.26 Interface of List of Equipment (Request Equipment)

The screenshot shows the 'Cart for KBV' page. At the top, there is a search bar with the placeholder 'Search product, supplier, order'. To the right of the search bar are a bell icon and a user profile picture. Below the search bar, the title 'Cart for KBV' is displayed, along with a breadcrumb navigation 'Equipment List > Cart'. On the left side, there are several navigation links: 'Rent Equipment' (highlighted in red), 'Orders', 'Settings', and 'Log Out'. The main content area is divided into two sections: 'Camera' and 'Camera Lens'. Both sections have a 'Filters' button in the top right corner. The 'Camera' section contains 3 items, and the 'Camera Lens' section contains 2 items. Each item row includes columns for No., Brand, Model, Condition, Status, Notes, and Action (with 'Edit' and 'Remove' buttons). The 'Camera' section has rows for Canon 5D Mark IV (Good, Available), Canon 5D Mark IV (Good, Available), and Canon 5D Mark II (Good, Available). The 'Camera Lens' section has rows for Canon EF 50mm (Good, Available) and Canon EF17-40mm USM (Good, Available). A 'Rental Form' button is located in the top right corner of the main content area.

Figure 4.27 Interface of Cart Page (Request Equipment)

Cart

Equipment List > Cart

Request Form

Renter Name	<input type="text"/>
Phone No.	<input type="text"/>
Email	<input type="text"/>
Matric No.	<input type="text"/>
Event Name	<input type="text"/>
Location	<input type="text"/>
Start Date	<input type="text"/>
End Date	<input type="text"/>

Rental Form

Status	Notes	Action
Available	-	<button>Edit</button> <button>Remove</button>
Available	-	<button>Edit</button> <button>Remove</button>
Available	EU	<button>Edit</button> <button>Remove</button>

Status	Notes	Action
Available	-	<button>Edit</button> <button>Remove</button>
Available	-	<button>Edit</button> <button>Remove</button>

Cancel **Submit Form**

Figure 4.28 Interface of Equipment Request Form (Request Equipment)

KBV details

Equipment List > Rental List > FutureReady

KBV

KBV (ID123124123123)	Return Equipment
1A/Krishnarajapuram, 3 rd street sulur	
Ahmad Saifudin	
20/6/2025 - 25/6/2025	Status: Ongoing

Camera

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	5D Mark IV	Good	Available	-	<button>Edit</button> <button>Remove</button>
2	Canon	5D Mark IV	Good	Available	-	<button>Edit</button> <button>Remove</button>
3	Canon	5D Mark II	Good	Available	EU	<button>Edit</button> <button>Remove</button>

Camera Lens

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	<button>Edit</button> <button>Remove</button>

Settings

Log Out

Figure 4.29 Interface of Specific Equipment Request Details (Request Equipment)

4.5.2 User Interface for Equipment Management Module

The screenshot shows the 'Club's Equipment List' interface. On the left sidebar, there are links for Rent Equipment, Manage Equipment, Equipment Requests, Equipment Returns, Orders, Settings, and Log Out. The main area has a search bar at the top. Below it, there are two tables:

Camera							
No.	Brand	Model	Serial Number	Condition	Status	Notes	Action
1	Canon	5D Mark IV	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
2	Canon	5D Mark IV	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
3	Canon	5D Mark II	XXX12390	Broken	Unavailable	EU	<button>Edit Equipment</button> <button>Remove</button>
4	Canon	5D Mark II	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
5	Canon	60D	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
6	Nikon	D90	XXX12390	Broken	Unavailable	KBV	<button>Edit Equipment</button> <button>Remove</button>
7	Nikon	D90	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
8	Nikon	D7000	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>

Camera Lens							
No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>

Figure 4.30 Interface of List of Club's Equipment (Manage Equipment)

The screenshot shows the 'Update Equipment Form' dialog box over the 'Club's Equipment List' interface. The dialog box contains fields for updating equipment details:

Camera	
No.	Brand
1	Canon
2	Canon
3	Canon
4	Canon
5	Canon
6	Nikon
7	Nikon
8	Nikon

Camera Lens	
No.	Brand
1	Canon
2	Canon

Form fields include:

- Image Upload: Drag image here or Browse image
- Brand: Canon
- Model: Canon EF 50mm
- Serial Number: Canon EF17-40mm USM
- Condition: Good
- Status: Available
- Notes: -
- Action: Edit Equipment, Remove

At the bottom of the dialog box are 'Cancel' and 'Save Equipment' buttons.

Figure 4.31 Interface of Update Equipment Form (Manage Equipment)

Add Equipment Form

No.	Brand	Model	Serial Number	Condition	Status	Condition
1	Canon					
2	Canon					
3	Canon					
4	Canon					
5	Canon					
6	Nikon					
7	Nikon					
8	Nikon					

No.	Brand	Model	Serial Number	Condition	Status	Condition
1	Canon	Prime	Canon EF 50mm	Good	Available	
2	Canon	Normal	Canon EF17~40mm USM	Good	Available	

Figure 4.32 Interface of Add Equipment Form (Manage Equipment)

Manage Equipment Requests and Rentals

Requests and Rentals List

Manage Equipment Rent

FutureReady (ID123124123123)	Manage
1A/Krihnarajapuram, 3 rd street sulur	
Muhammad Taufiq	
20/6/2025 - 25/6/2025	Status: Approved
View Invoice View Receipt	

CONVOCATION - 68 (ID123124123123)	Manage
1A/Krihnarajapuram, 3 rd street sulur	
Ahmad Hazim	
23/6/2025 - 26/6/2025	Status: Pending Approval
View Invoice View Receipt	

Manage Equipment Request

KBV (ID123124123123)	Manage
1A/Krihnarajapuram, 3 rd street sulur	
Muhammad Taufiq	
20/6/2025 - 25/6/2025	Status: Approved

Figure 4.33 Interface of List of Equipment Request and Rentals (Manage Equipment Requests and Rentals)

The screenshot shows the 'Manage Equipment Requests and Rentals' section of the application. At the top, there is a search bar labeled 'Search product, supplier, order'. To the right of the search bar are a notification icon and a user profile picture. Below the search bar, the title 'Manage Equipment Requests and Rentals' is displayed, followed by the breadcrumb 'Requests and Rentals List > FutureReady'. A sub-section titled 'FutureReady' is shown with a request summary: 'FutureReady (ID123124123123)', address '1A/Krithnarajapuram, 3 rd street sulur', total amount 'RM 75.00', customer 'Muhammad Taufiq', and date range '20/6/2025 - 25/6/2025'. The status is 'Approved'. There are 'Approve' and 'Reject' buttons. Below this, there are two tables: one for 'Camera' and one for 'Camera Lens'. Both tables have columns for No., Brand, Model, Condition, Status, Notes, and Action (with 'Manage' and 'Remove' buttons). The 'Camera' table has three entries, and the 'Camera Lens' table has one entry.

Figure 4.34 Interface of Managing Equipment Request and Rentals (Manage Equipment Requests and Rentals)

The screenshot shows the 'Manage Equipment Returns' section of the application. At the top, there is a search bar labeled 'Search product, supplier, order'. To the right of the search bar are a notification icon and a user profile picture. Below the search bar, the title 'Manage Equipment Returns' is displayed, followed by the breadcrumb 'Equipment Return List'. A sub-section titled 'Manage Equipment Rent Return' is shown with a request summary: 'FutureReady (ID123124123123)', address '1A/Krithnarajapuram, 3 rd street sulur', customer 'Muhammad Taufiq', and date range '20/6/2025 - 25/6/2025'. The status is 'Pending Return'. There is a 'Manage' button. Below this, another sub-section titled 'CONVOCATION - 68 (ID123124123123)' shows similar details with a status of 'Returned' and a 'Manage' button. At the bottom, there is a sub-section titled 'Manage Equipment Request Return' for 'KBV (ID123124123123)' with a status of 'Approved' and a 'Manage' button. On the left side of the main content area, there is a sidebar with navigation links: 'Rent Equipment', 'Manage Equipment', 'Equipment Requests', 'Equipment Returns', 'Orders', 'Settings', and 'Log Out'.

Figure 4.35 Interface of List of Equipment Returns (Manage Equipment Returns)

Manage Equipment Returns

Equipment Return List > FutureReady

FutureReady

FutureReady (ID123124123123)			
1A/Krishnarajapuram, 3 rd street sulur	Muhammad Taufiq	Approve	Reject
20/6/2025 - 25/6/2025	Status: Pending Return		
No.	Brand	Model	Condition
1	Canon	5D Mark IV	Good
2	Canon	5D Mark IV	Good
3	Canon	5D Mark II	Good

Camera							
No.	Brand	Model	Condition	Status	Notes	Action	Filters
1	Canon	5D Mark IV	Good	Available	-	Manage Remove	
2	Canon	5D Mark IV	Good	Available	-	Manage Remove	
3	Canon	5D Mark II	Good	Available	EU	Manage Remove	

Camera Lens							
No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Manage Remove

Settings

Log Out

Figure 4.36 Interface of Managing Equipment Returns (Manage Equipment Returns)

4.5.3 User Interface for Role and Event Management Module

Event List

Event List

List of Events

No.	Name	Description	Location	Status	Start Date	End Date	Action
1	KBV	Video Lesson	SUB	Ongoing	23/5/2025	28/5/2025	Manage Event Manage Member
2	MMM	Minggu Mesra Mahasiswa	DSI	Ongoing	23/5/2025	28/5/2025	Manage Event Manage Member
3	Wedding	Perkahwinan Adam	Pondok	Upcoming	23/6/2025	28/6/2025	Manage Event Manage Member

Settings

Log Out

Figure 4.37 Interface of List of Event

The screenshot shows the 'Manage Member' section of the application. At the top, there is a search bar labeled 'Search product, supplier, order'. To the right of the search bar are a bell icon and a user profile picture. Below the search bar, the title 'Manage Member' is displayed, followed by the breadcrumb 'Event List > Manage Equipment Committee Member'. There are two tables: 'Assign Equipment Committee Member to KBV' and 'Assigned Equipment Committee Member to KBV'. Both tables have columns for No., Name, email, Phone No., Matric No., Roles, and Action. The first table contains three rows with names like Muhammad Taufiq, Muhammad Nabil, and Ahmad Saifudin. The second table contains two rows with names like Muhammad Alif and Ahmad Hazim. Each row in both tables has a 'View Roles' button and a '+ Add' or 'Remove' button.

Figure 4.38 Interface of Manage Club Members

The screenshot shows the 'Event List' section with a modal for 'Add Event Form'. The modal has fields for Event Name (Renter Name), Description (Phone Number), Location (Email), Matric No. (Matric Number), Start Date (dd/mm/yy), and End Date (dd/mm/yy). Below these fields is a section titled 'Assign Equipment Committee Member' with a '+ Add' button. At the bottom of the modal are 'Cancel' and 'Create Event' buttons. In the background, there is a list of events with columns for No., Name, Start Date, End Date, and Action. The list includes entries for KBV, MMM, and Wedding, along with their respective dates and manage buttons.

Figure 4.39 Interface of Add Event Form

4.6 Chapter Summary

In summary, chapter 4 summarise the requirement analysis, project design, and database structure of the Inventory Management System for Kelab Fotokreatif (IFoto). It includes use case description, activity diagrams and sequence diagrams, class diagram and the system architectural style. This chapter explains how 4 main actors which are User, Equipment Executive Committee, Equipment Committee Member and Administrator facilitate various activities such as equipment management, role and event management, equipment requests and rental and authentication and authorization functionalities aiming to streamline the inventory management process for all actors. This chapter also outlines few examples of interface designs for all actor, illustrating their business process. The system's comprehensive design ensures a seamless inventory management process by integrating multiple functionalities and user-friendly interface.

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter summarizes and reflects upon the development of the IFoto: Inventory Management System of Kelab Fotokreatif as a whole. It also assesses the extent of success that has been obtained by the project and gives well-considered suggestions on how it can be improved in the future. The software development cycle took a hybrid Agile-Waterfall methodology that provided a well-organized process of development with the ability to refine over time. This chapter is a conclusive summary of the thesis, with the work done in each of the chapters connected to the end product of the system.

Chapter 1 presented the background of the equipment rental problems that Kelab Fotokreatif is experiencing, as well as the purpose, objectives, scope, and importance of the proposed IFoto system. Chapter 2 has revised the associated systems and the associated technologies such as ReactJS, Spring Boot, and MySQL. Chapter 3 explained the hybrid Agile-Waterfall approach that was employed to drive development. The requirement specifications and system architecture and design with different architectural perspectives were given in Chapter 4. In this concluding chapter the author summarizes all the above work and provides future improvements to the system.

5.2 Achievement of Project Objectives

During the process of this project, it has been able to make a considerable headway towards the realization of the desired goals. The first one, the analysis and determination of the requirements to develop the Inventory Management System of Kelab Fotokreatif (IFoto) was achieved, as the detailed requirement elicitation was conducted. This included interviews, analysis of the existing club operations and a study of related inventory systems. The Software Requirements Specification (SRS) document that was produced gave a clear definition of the functional and non-functional requirements of the system, the use cases, user roles and the constraints of the system.

The second goal of designing a web-based application that should be highly intuitive to users and easy to navigate was achieved by implementing a Model-View-Controller (MVC) architecture. ReactJS was used to create the user interface in a way that was responsive and easy to use and Spring Boot was used on the back end to handle business logic and routing of the API. Stress was made on UI/UX to improve the user experience, especially in important user roles like the Equipment Executive Committee and Event Committee Members. Such features as dynamic equipment booking, visibility of inventory, and access controls based on roles were added so that the software could be used and the work flow could be facilitated.

The two objectives were successfully achieved and the success of these two objectives show that the system is ready to have additional functional extensions and practical application.

5.3 Suggestions for Future Improvement

Although the implemented Inventory Management System of Kelab Fotokreatif (IFoto) has achieved the set primary functional requirements, the system can be improved in the future regarding scalability, performance, and user experience.

In addition, the implementation of a user feedback feature is one of the possible future enhancements that can be made to the system, allowing renters and club members to leave comments or ratings regarding the usability of the system, the state of the equipment, and the overall experience of using the service. Such a feedback system can assist administrators and equipment managers to detect the recurring problems, prioritize maintenance, and improve the quality-of-service delivery.

Also, report generation functions could be added so that administrators would be able to export monthly reports of equipment usage, member activity, or payment totals. This would make it easy to make data-driven decisions and strategic planning of the club.

With these recommendations, IFoto will be able to become a more powerful and all-purpose system that will be able to serve the operational and strategic objectives of Kelab Fotokreatif better.

REFERENCES

- Abramov, D. and Redux documentation. (2021, June 19). Why Use React Redux? | React Redux. React-Redux.js.org. <https://react-redux.js.org/introduction/why-use-react-redux>
- Atlassian. (2024). Jira. Atlassian. <https://www.atlassian.com/software/jira>
- Bigelow, S. J., & Gillis, A. S. (2024, May 10). *What is Rest API (restful API)?* Search App Architecture. <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API>
- Corman, M. (2020, April 16). *An introduction to redux.* Medium. <https://medium.com/@madelinecorman/an-introduction-to-redux-359baccde63d>
- Emon, E. H. (2024, July 4). *Orm, JPA, Spring Data JPA, Spring Data Mongodb: A beginners guide for Java Spring boot.* Medium. <https://medium.com/@emonemrulhasan35/orm-jpa-spring-data-jpa-spring-data-mongodb-a-beginners-guide-for-java-spring-boot-32834c10729d>
- Hayes, A. (2024, June 27). *Inventory management: Definition, how it works, Methods & Examples.* Investopedia. <https://www.investopedia.com/terms/i/inventory-management.asp>
- Heller, M. (2022, July 8). What is Visual Studio Code? Microsoft's extensible code editor. InfoWorld. <https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html>
- Herbert, D. (2022, June 27). What is React.js? (Uses, Examples, & More). Blog.hubspot.com. <https://blog.hubspot.com/website/react-js>

Jackson, B. (2023, November 17). What Is GitHub? A Beginner's Introduction to... Kinsta Managed WordPress Hosting; Kinsta Managed WordPress Hosting.
<https://kinsta.com/knowledgebase/what-is-github/>

Nugroho, A., Sari, D. R., Permana, H. D., & Negara, R. S. (2021). *Rancang Bangun Aplikasi Inventory Berbasis Web dengan Menggunakan Model MVC*. GUEPEDIA. ISBN: 62-39-8329-8

Postman. (2021). Postman | The Collaboration Platform for API Development. Postman. <https://www.postman.com/>

Shinde, V. (2025, May 9). *Waterfall and agile methodology to hybrid model case study*. Waterfall and Agile Methodology to Hybrid Model Case Study. <https://www.softwaretestinghelp.com/agile-waterfall-hybrid-model/>

Sparx Systems Ltd and SparxSystems Software GmbH. (2024). sparx - eu: UML, SysML, BPMN, Togaf, Updm unified in Enterprise Architect von Sparx Systems. <https://www.sparxsystems.eu/>

Walls, C. (2016). *Spring Boot In Action*. Manning Publications.

Westfall, B. (2016, March 28). Leveling Up with React: Redux. CSS-Tricks. <https://css-tricks.com/learning-react-redux/>

Żurawiecki, J. (2024, April 29). *Key differences between waterfall, Agile, and hybrid*. BigPicture. <https://bigpicture.one/blog/waterfall-agile-hybrid-differences/>

Appendix A List of Prompts

1. <https://chatgpt.com/share/685fb51-ce40-8003-b67b-65935822a848>

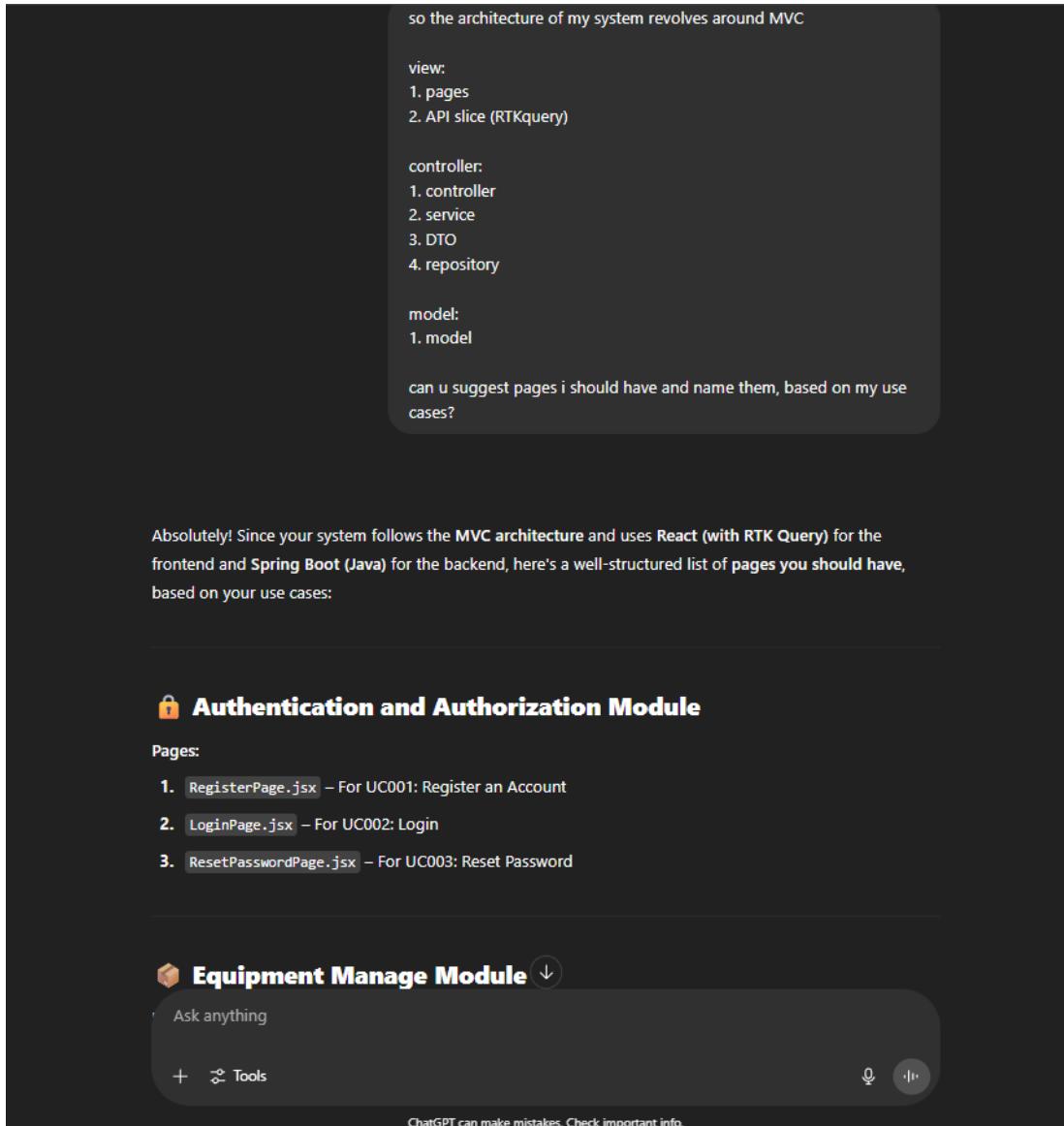


Figure A.1 Example of Prompt Used

Appendix B Software Requirements Specification (SRS)



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECJ 3032: Software Engineering FYP1
Semester 02, 2024/2025

Software Requirements Specification (SRS)

Inventory Management System for Kelab
Fotokreatif

Version 1.0

28/05/2025

Prepared by: Lio Kock Hock

Revision Page

1.1 Overview

The present document includes the introduction of Inventory Management System of Kelab Fotokreatif (IFoto), the general description of the system and the specific requirement that consists of use case diagram, use case description, activity diagram and sequence diagram which were modelled using Unified Modelling Language (UML). This document also includes Non-functional requirements and design constraints.

1.2 Target Audience

The intended audience is Kelab Fotokreatif members, developers, project managers and software testers.

1.3 Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Lio Kock Hock	Completed SRS version 1.0	28/05/2025

Note:

This Software Requirements Specification (SRS) template is adapted from IEEE Recommended Practice for Software Requirements Specification (SRS) (IEEE Std. 830-1998) that is simplified and customized to meet the need of SECJ3203 FYP1 SE course at Faculty of Computing, UTM. Examples of models are from Arlow and Neustadt (2002) and other sources stated accordingly.

Table of Contents

1	Introduction	76
1.1	Purpose.....	76
1.2	Scope.....	76
1.3	Definitions, Acronyms and Abbreviation	77
1.4	References.....	77
1.5	Overview	78
2	Specific Requirements	79
2.1	User Characteristics.....	79
2.1.1	User Role 1: User	79
2.1.2	User Role 2: Equipment Committee Member	79
2.1.3	User Role 3: Equipment Executive Committee	80
2.1.4	User Role 4: Administrator.....	80
2.2	System Features.....	82
2.2.1	Use Case Diagram	82
2.2.2	Class Diagram	84
2.2.3	State Diagram.....	87
2.3	Use Case Details	90
2.3.1	UC001: Register An Account.....	90
2.3.2	UC002: Login	94
2.3.3	UC003: Reset Password.....	97
2.3.4	UC004: View Equipment List	100
2.3.5	UC005: Rent Equipment.....	103
2.3.6	UC006: Return Equipment.....	107
2.3.7	UC007: Make Payment.....	111
2.3.8	UC008: View Generated Invoice and Receipt.....	114
2.3.9	UC009: Request Equipment	117
2.3.10	UC010: Monitor Equipment.....	121
2.3.11	UC011: Update Equipment	124
2.3.12	UC012: Manage Equipment Requests and Rentals	128
2.3.13	UC013: Confirm Equipment Return	134
2.3.14	UC014: Manage Members Roles.....	137
2.3.15	UC015: Manage Event.....	141
2.4	Performance and Other Requirements	145
2.5	Design Constraints.....	146

1 Introduction

1.1 Purpose

The SRS explains how the Inventory Management System of Kelab Fotokreatif (IFoto) was introduced, general description of the processes involved in the development of the system, user story and its description, specific requirements that involve enhanced use case diagram, sequence diagram and activity diagram that were developed using Unified Modelling Language and Enterprise Architect software. The members, developers, project managers and software testers of Kelab Fotokreatif will be the intended audience of this document. The description of this document in detail will be used as a reference to the target audience on the functional and non-functional requirements of the system.

1.2 Scope

Inventory Management System for Kelab Fotokreatif (IFoto) is web-based application to streamline the management of club's equipment, club members role and equipment requests and rentals. The system consists of four different modules which are Equipment Request and Rental Module, Authentication and Authorization Module, Equipment Management Module and Role and Event Management Module.

Firstly, Equipment Requests and Rental Module is responsible to handle the requesting, renting, and returning process of club's equipment. All users of this system have the permission to Rent Equipment. Equipment Committee Member have the permission to Request for Equipment for specific event. This system allow borrower to return the Equipment.

Next, Authentication and Authorization Module manage user registration and login utilizing a built-in Role-Based Access Control (RBAC) feature in Spring Java framework. The user only has access to their respective functionality according to their assigned role.

Besides that, Equipment Management Module allows Equipment Executive Committee members to monitor and manage equipment details, availability, and status. This system also allows Equipment Executive Committee members to approve equipment booking requests and rentals and confirm its return to ensure smooth equipment management experience.

Furthermore, Role and Event Management Module provide functions for administrator to manage club Events and manage club Members Roles. Administrator can assign any club member as Equipment Committee Member to any event.

The benefit of implementing Inventory Management System in the Kelab Fotokreatif (IFoto) is to improve the productivity with fewer errors to address, the club can focus on the providing higher-quality services (Garcia et al., 2024). The key benefits include real-time tracking of equipment availability, reduced administrative workload, minimized risk of double-booking, and better communication among club members. In other words, the objectives of IFoto are to improve efficiency, transparency, and traceability in equipment handling and club operations.

1.3 Definitions, Acronyms and Abbreviation

Definitions of all terms, acronyms and abbreviation used are:

SRS – System Requirements Specifications

UML - Unified Modelling Language

RBAC – Role-Based Access Control

1.4 References

Garcia, R., Herrity, J., Gafner, J., Eads, A., Mendoza, A., & Lagace, L. (2024, August 16). *10 benefits of using an inventory management system | indeed.com*. indeed. <https://www.indeed.com/career-advice/career-development/benefits-of-using-inventory-management-systems>

Pohl, K. & Rupp, C. (2020, May 15). Requirements Engineering Fundamentals. California: Rocky Nook Inc.

1.5 Overview

This SRS has two sections. The first section describes the specific requirements for the Inventory Management System for Kelab Fotokreatif (IFoto). In the specific requirements section, the user characteristics for each user role are stated. Next, it also describes the system features, use case details, performance and other requirements, and design constraint that will affect the system. In use case details subsection, the use cases are explained in three forms which are use case description, sequence diagram and activity diagram. Use case description explains the way users interact with systems to accomplish task or goals. Activity diagram depicts the control flow between activities or actions, and we model it for each respective use case (Pohl, K. & Rupp, C., 2020). Sequence diagram represents the messages sequence passed and the control structures between objects. The second section describes the specific requirements for the IFoto. Besides that, performance and other requirements are referred to non-functional requirements which addresses the characteristic or qualities of the system such as reliability, usability, performance, security, etc. Lastly, design constraints represent any constraint imposed by the Kelab Fotokreatif.

2 Specific Requirements

2.1 User Characteristics

2.1.1 User Role 1: User

- **Position Level:** Club members, external users, students and alumni
- **Technical Skill Level:** User are expected to have basic computer skill to interact with user-friendly user interface such as rental forms, login pages, and payment gateways.
- **System Interaction:** User can browse list of equipment, rent equipment, and make payment to complete transaction via online payment gateway. User also must return it after upon due date.
- **Characteristics:**
 - User only requires minimal computer or technical knowledge.
 - User focuses on simple interface to have access to equipment rental services.
 - User needs a clear user interface to navigate the system.

2.1.2 User Role 2: Equipment Committee Member

- **Role:** Equipment Committee Member was assigned to specific events by Administrator that enables him to request for equipment.
- **Position Level:** Mid-level committee member and Equipment Committee Member for certain event.
- **Technical Skill Level:** Equipment Committee Member basic to mid-level computer skills to perform request equipment, verify equipment condition and status and return equipment.
- **System Interaction:** Equipment Committee Member can view equipment list and request it. Equipment Committee Member also have to return it upon due date.
- **Characteristics:**

- Equipment Committee Member works closely with the Executive Equipment Committee.
- Equipment Committee Member must be organized and responsive to event committee member's needs.
- Equipment Committee Member are active during event planning and execution phases.

2.1.3 User Role 3: Equipment Executive Committee

- **Role:** Equipment Executive Committee manages inventory and equipment availability, and verifies requests and rentals.
- **Position Level:** Equipment Executive Committee
- **Technical Skill Level:** Equipment Committee Member are experienced in equipment management and they are expected to have basic to moderate computer skills.
- **System Interaction:** Equipment Committee Member often updates equipment conditions and status, confirms returns, manages equipment assets and verifies requests and rentals.
- **Characteristics:**
 - Equipment Committee Member are very familiar with club inventory and its details.
 - Equipment Committee Member acts as the connecting bridge between Equipment Committee Members of an event and the club's equipment.
 - Equipment Committee Member are responsible for managing accurate equipment status and conditions.

2.1.4 User Role 4: Administrator

- **Role:** Administrator oversees Equipment status and conditions, club members and events Assigns roles, monitors activities, and ensures smooth operation of events and equipment management.

- **Position Level:** President and Vice President
- **Technical Skill Level:** Administrator are supposed to possess moderate level of computer skills and familiar with general system usage, administrative monitoring tools.
- **System Interaction:** Administrator utilizes user-friendly interface to manage club members roles and events.
- **Characteristics:**
 - Administrators are a decision-makers with strong organizational knowledge.
 - Administrator is responsible for managing club members roles for certain events.
 - Administrator ensures alignment between club goals and system activities.

2.2 System Features

2.2.1 Use Case Diagram

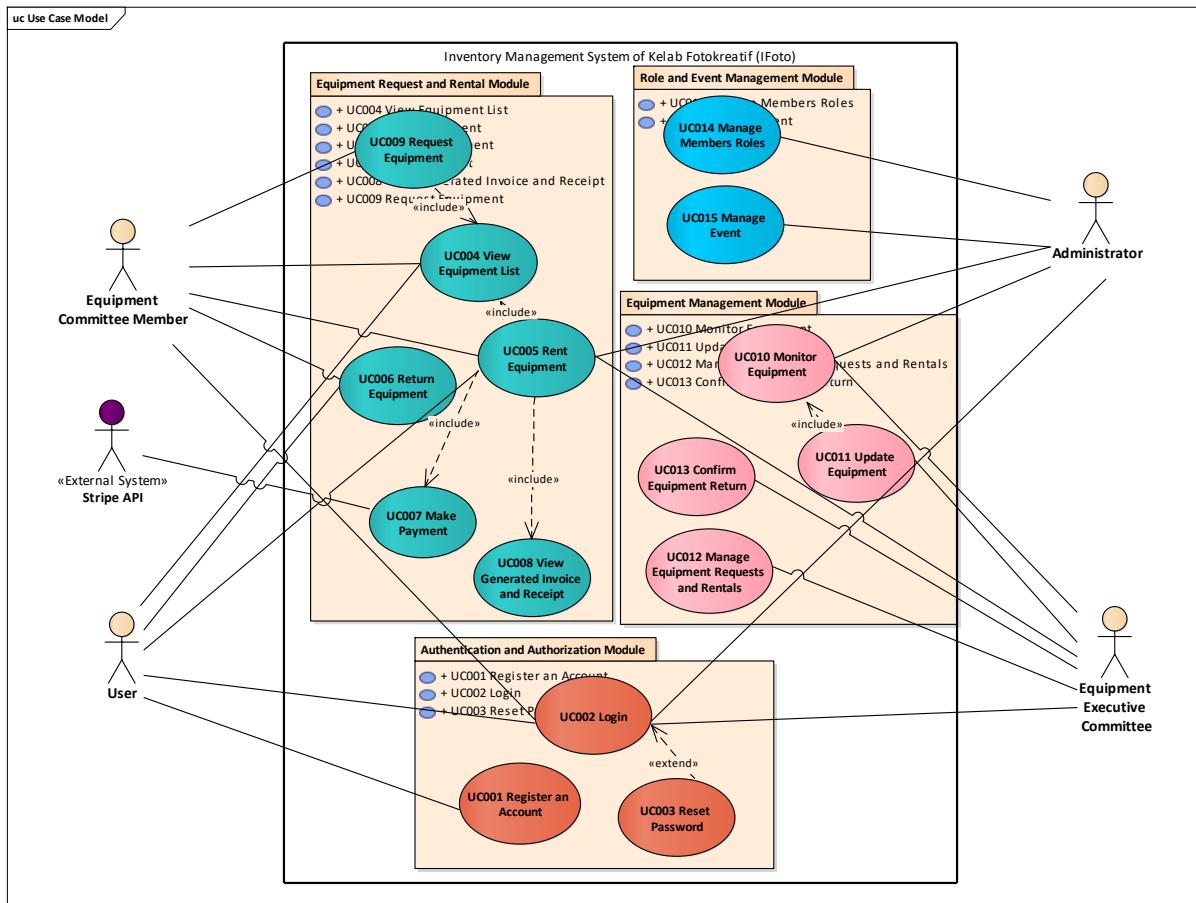


Figure 2.1 Use Case Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)

Table 2.1 Table of Description for each Use Case

Use case	Function	Description
UC001	Register an Account	This use case allows User to sign up as a normal user by providing personal information and credentials.
UC002	Login	This use case enables registered User to log in to the app using their username and password.
UC003	Reset Password	This use case allows User who forgot their password to reset it via email verification.

UC004	View Equipment List	This use case displays a list of all available equipment details such as name, quantity, status, and condition, allowing User and Equipment Committee Member to view it.
UC005	Rent Equipment	This use case allows any user to fill in the Club's Equipment Rental Form digitally to request for equipment formally.
UC006	Return Equipment	This use case facilitates the return process of rented by User and borrowed equipment by the Equipment Committee Member and update the inventory.
UC007	Make Payment	This use case allows User to make payments for equipment rentals through integrated payment gateways.
UC008	View Generated Invoice and Receipt	This use case allows system view the invoice generated after equipment Rent has been approved and view the receipt after payment is made.
UC009	Request Equipment	This use case allows Equipment Committee Member to request equipment in advance for planned events or activities.
UC010	Monitor Equipment	This use case enables the Equipment Executive Committee and Administrator to track equipment availability, status, and conditions.
UC011	Update Equipment	This use case allows Executive Equipment Committee to add, edit, or delete equipment details in the system.
UC012	Manage Equipment Requests and Rentals	This use case enables the Equipment Executive Committee to approve, reject, or manage ongoing rentals and requests.
UC013	Confirm Equipment Requests	This use case allows the Equipment Executive Committee to confirm the return of borrowed equipment after verifies its conditions and completeness.
UC014	Manage Members Roles	This use case allows Administrator to assign or revoke roles of the club members within the system.
UC015	Manage Event	This use case allows Administrator to create, update, and manage events that involve assignment of roles.

2.2.2 Class Diagram

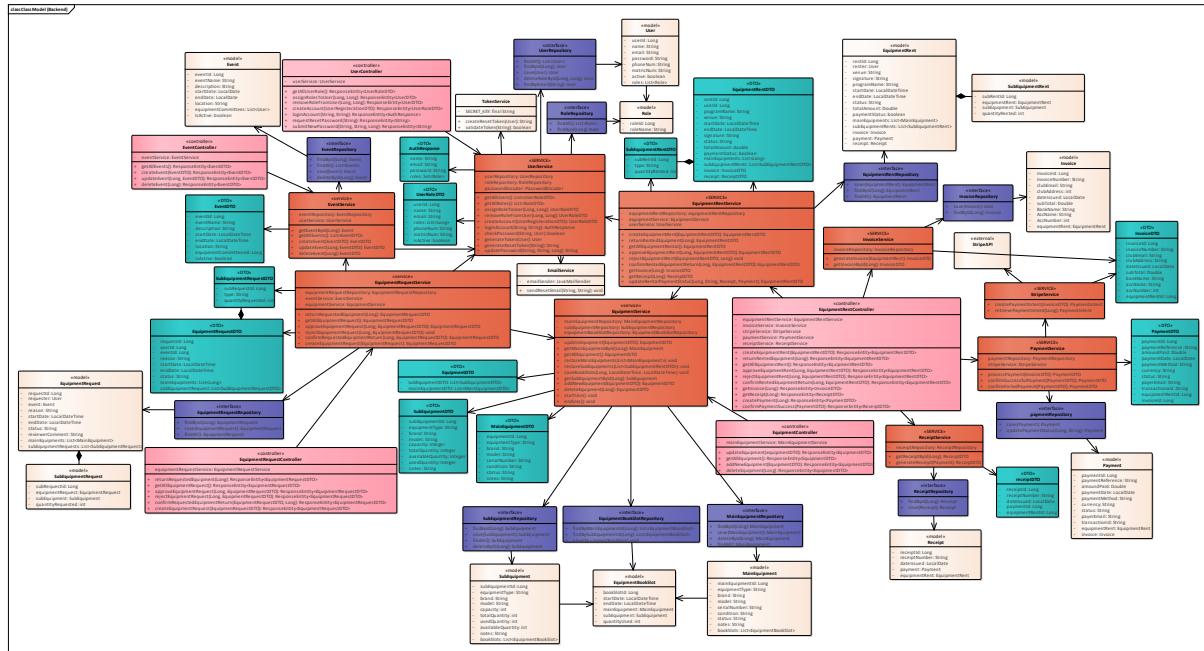


Figure 2.2 Class Diagram Inventory Management System for Kelab Fotokreatif (IFoto)

Table 2.2 Table of Description for each Class

Stereotype	Class Name	Description
<<model>>	Event	Core business entity representing events with properties like eventId, eventName, description, dates, location, and equipment committees
	User	Core business entity representing system users with userId, name, email, password, phone, matriculation number, and roles
	Role	Core business entity representing user roles with roleId, roleName, and associated users
	Equipment	Core business entity representing equipment items with rental, renter, venue, signature, program name, dates, and status information
	MainEquipment	Core business entity representing main equipment with equipmentId, type, brand, model, serial number, condition, and status
	SubEquipment	Core business entity representing sub-equipment with subEquipmentId, equipmentType, brand, model, and capacity
	EquipmentRequest	Core business entity representing equipment requests with requestId, requester, event, reason, dates, status, and sub-equipment requests
	SubEquipmentRequest	Core business entity representing sub-equipment requests with subRequestId, equipment request, sub-equipment, and quantity requested

<<DTO>>	EventDTO	Data Transfer Object for Event entity with eventId, eventName, startDate, endDate, venue, and description
	UserRoleDTO	Data Transfer Object for User roles with userId, fullName, email, and roles information
	AuthResponse	Data Transfer Object for user login with name, email, password, and roles
	EquipmentRentDTO	Data Transfer Object for equipment rental with rentId, userId, program name, venue, dates, and equipment details
	SubEquipmentRentDTO	Data Transfer Object for sub-equipment rental with rental details and equipment information
	EquipmentRequestDTO	Data Transfer Object for equipment requests with requestId, userId, eventId, reason, dates, and status
	SubEquipmentRequestsDTO	Data Transfer Object containing list of sub-equipment requests
	SubEquipmentDTO	Data Transfer Object for sub-equipment with detailed specifications and availability information
	EquipmentDTO	Data Transfer Object for equipment with sub-equipment DTOs and main equipment DTOs
	MainEquipmentDTO	Data Transfer Object for main equipment with equipmentId, type, brand, model, serial number, condition, and status
<<Repository>>	EventRepository	Data access interface for Event entities defining CRUD operations
	UserRepository	Data access interface for User entities defining CRUD operations
	RoleRepository	Data access interface for Role entities defining CRUD operations
	EquipmentRentRepository	Data access interface for Equipment rental operations
	EquipmentRequestRepository	Data access interface for Equipment requests operations
	EquipmentRepository	Data access interface for Equipment entities with findById, save, and other CRUD operations
<<service>>	EventService	Business logic service for event operations using EventRepository
	UserService	Business logic service for user operations with UserRepository, RoleRepository, and PasswordEncoder
	EquipmentRentService	Business logic service for equipment rental operations using EquipmentRentRepository
	EquipmentRequestService	Business logic service for equipment requests operations using EquipmentRequestRepository
	EquipmentService	Business logic service for equipment operations with MainEquipmentRepository and SubEquipmentRepository

	PaymentService	Business logic service for Payment Operations with PaymentRepository
	InvoiceService	Business logic service for generating Invoice Operations with InvoiceRepository
	ReceiptService	Business logic service for generating Receipt Operations with ReceiptRepository
	StripeService	Business logic service Payment Operations with PaymentRepository
<<controller>>	EventController	REST API controller handling event-related HTTP requests using EventService
	UserController	REST API controller handling user-related HTTP requests using UserService
	EquipmentRentController	REST API controller handling equipment rental requests using EquipmentRentService
	EquipmentRequestController	REST API controller handling equipment requests operations using EquipmentRequestService
	EquipmentController	REST API controller handling equipment operations with MainEquipmentService and updateEquipment methods

2.2.3 State Diagram

2.2.3.1 State Diagram EquipmentRequest class

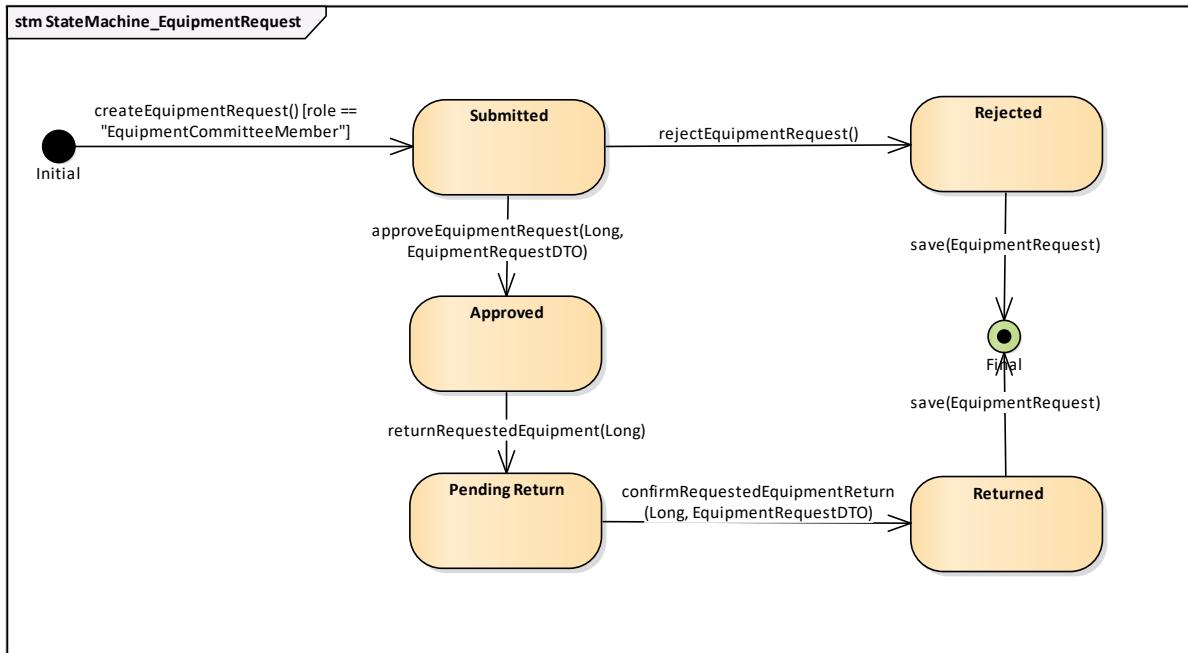


Figure 4.3 State Diagram for EquipmentRequest class

2.2.3.2 State Diagram EquipmentRent class

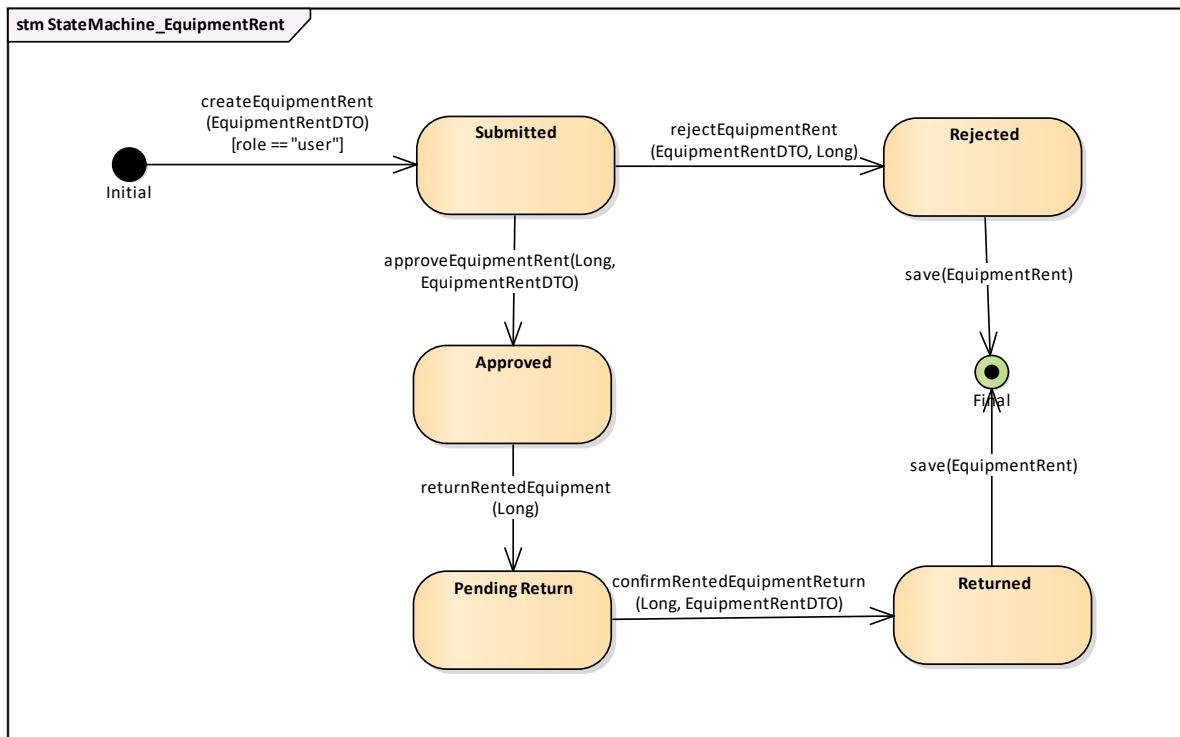


Figure 4.4 State Diagram for EquipmentRent class

2.2.3.3 State Diagram Payment class

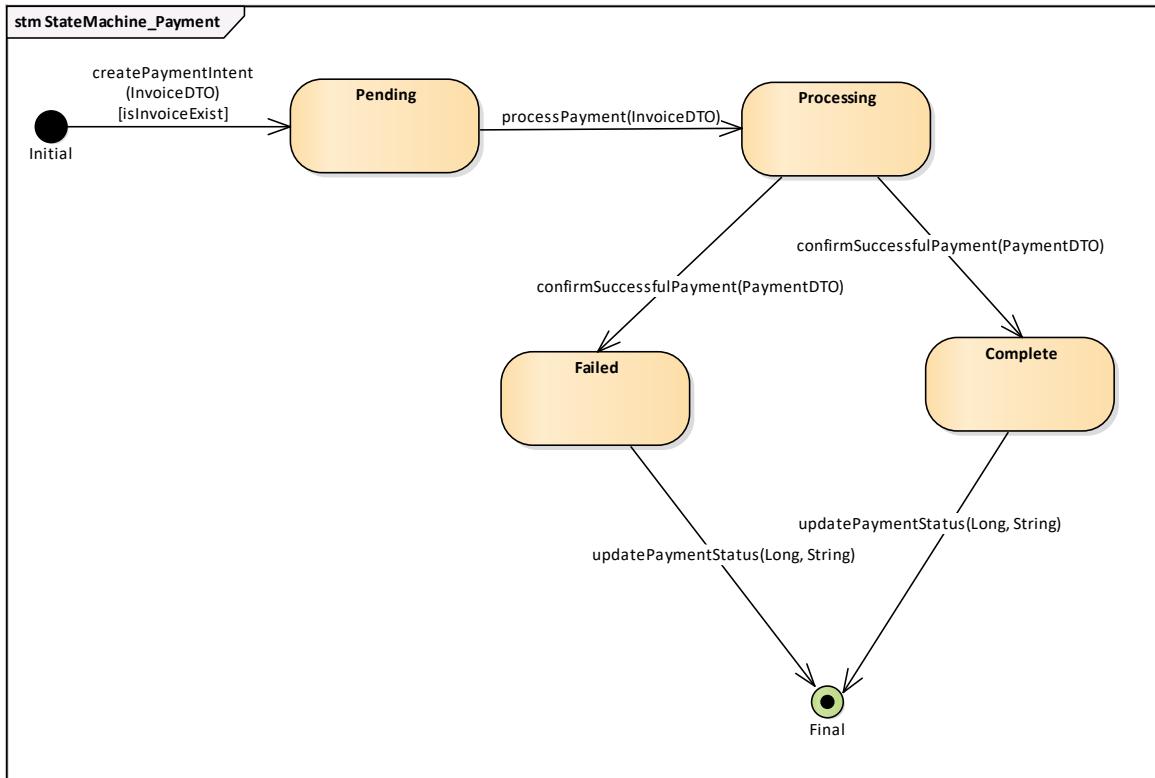


Figure 4.5 State Diagram for Payment class

2.2.3.4 State Diagram MainEquipment class

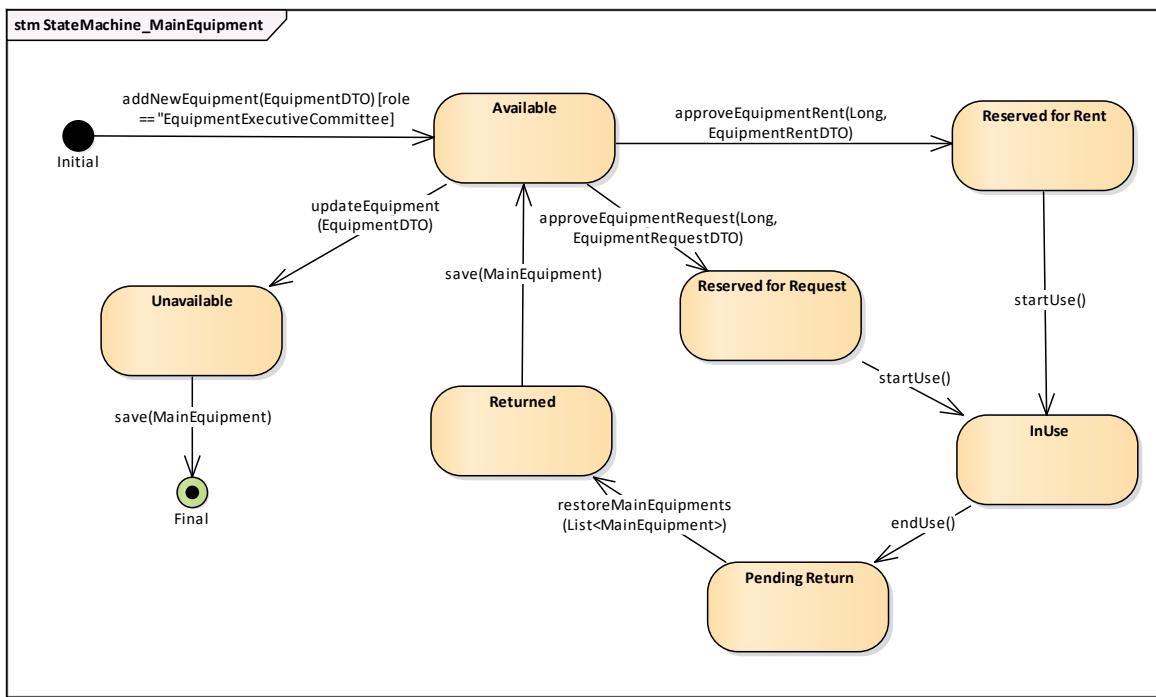


Figure 4.6 State Diagram for MainEquipment class

2.3 Use Case Details

2.3.1 UC001: Register An Account

2.3.1.1 Use Case Specification of UC001

Table 2.3 Table of Use Case Description for US001 Register Account

Use Case ID	UC001
Use Case Name	Register an Account
Description	This use case allows User to sign up as a normal user by providing personal information and credentials.
Actor(s)	User
Pre-condition(s)	Unauthenticated users can access registration page.
Normal Flow(s)- NF	<ol style="list-style-type: none">1. The user navigates to the registration page by clicking the "Register" button on the system's login screen.2. The user enters all their information and credentials in the required field to complete the registration.3. The user clicks the "Submit" button to send their registration details to the system.4. The system validates all inputs.5. The system sends an email notification to the user's email for verification.6. The system display "Check your email to verify your account."
Alternative Flow(s) - AF	<p>AF1. Email Exists</p> <ol style="list-style-type: none">1. System display "This email is already registered. Try logging in or use a different email."2. The user re-enter information by using different email. <p>AF2. Password Strength Fail</p> <ol style="list-style-type: none">1. System display "Password must contain 8+ characters with 1 special symbol (!@#\$%^&*)."2. The user re-enter new password.

Exception Flow(s) - EF	<p>EF1. Server Error</p> <ol style="list-style-type: none"> 1. System Display "Registration temporarily unavailable. Please try again later." 2. The user reloads the page. <p>EF1. Incomplete Form Submission</p> <ol style="list-style-type: none"> 1. System Show "Please fill all required fields (*)." 2. The user enters every required fields.
Post-condition(s)	Account is created with Pending Approval status.

2.3.1.2

Activity Diagram of UC001

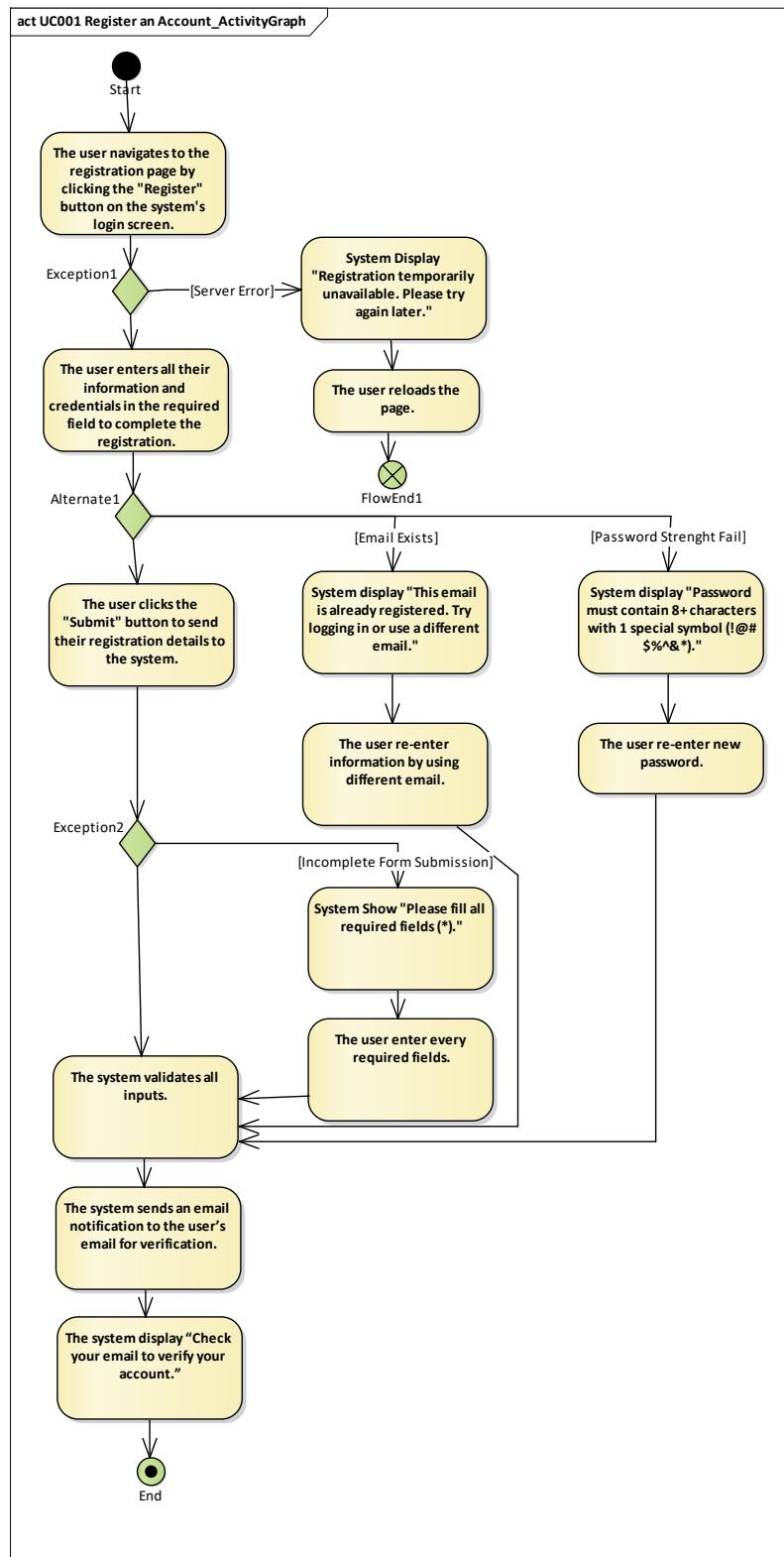


Figure 2.7 Activity Diagram for UC001 Register an Account

2.3.1.3 Sequence Diagram of UC001

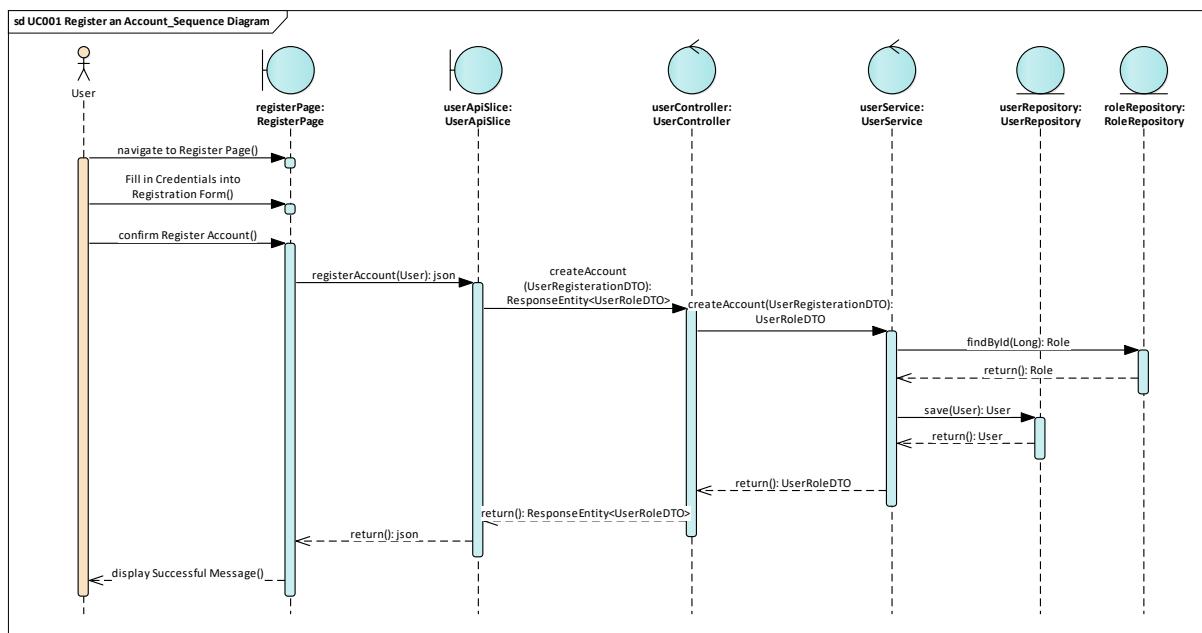


Figure 2.8 Sequence Diagram for UC001 Register an Account

2.3.2 UC002: Login

2.3.2.1 Use Case Description of UC002

Table 2.4 Table of Use Case Description for UC002 Login

Use Case ID	UC002
Use Case Name	Login
Description	This use case enables registered users to log in to the system using their username and password to access their respective features based on their role.
Actor(s)	User, Administrator, Equipment Executive Committee and Equipment Committee Member.
Pre-condition(s)	User has an active account.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The user navigates to the login page by clicking "Log In" button. 2. The user enters their credentials in the input fields. 3. The user clicks the "Login" button to submit credentials. 4. The system validates and verifies the email and the password. 5. The system redirects the user to their role-specific dashboard.
Alternative Flow(s) - AF	<p>AF1. Remember Me Option</p> <ol style="list-style-type: none"> 1. System shall extend session cookie expiry to 7 days. <p>AF2. Password Visibility Toggle</p> <ol style="list-style-type: none"> 1. System shall reveal the password in plaintext.
Exception Flow(s) - EF	<p>EF1. Invalid Credentials</p> <ol style="list-style-type: none"> 1. System shall display "Invalid email or password." 2. The user re-enter credentials in the input field. <p>EF1. Inactive Account</p> <ol style="list-style-type: none"> 1. System displays "Account has not been verified."
Post-condition(s)	User gains access to role-specific dashboard or interface.

2.3.2.2 Activity Diagram of UC002

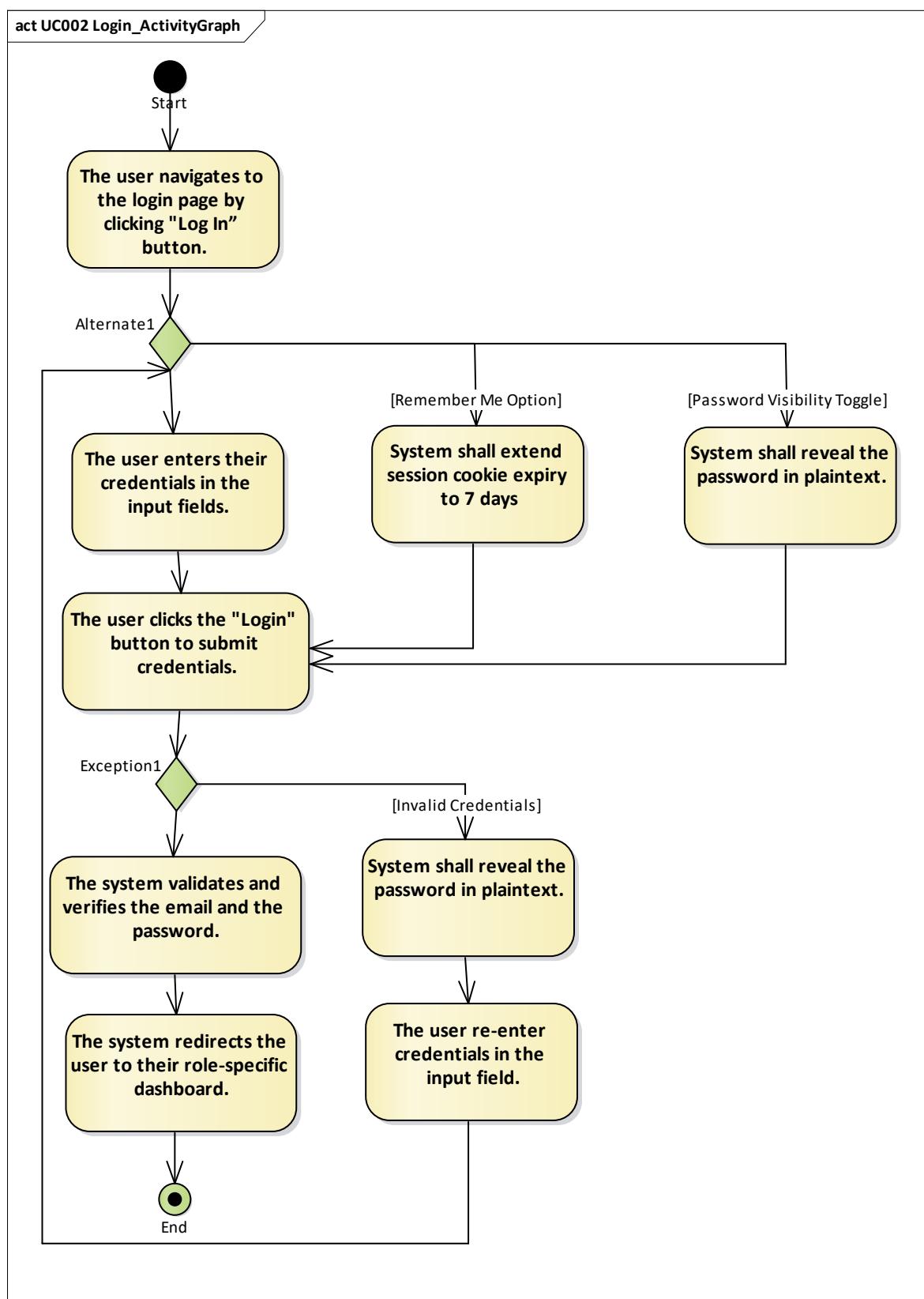


Figure 2.9 Activity Diagram for UC002 Login

2.3.2.3 Sequence Diagram of UC002

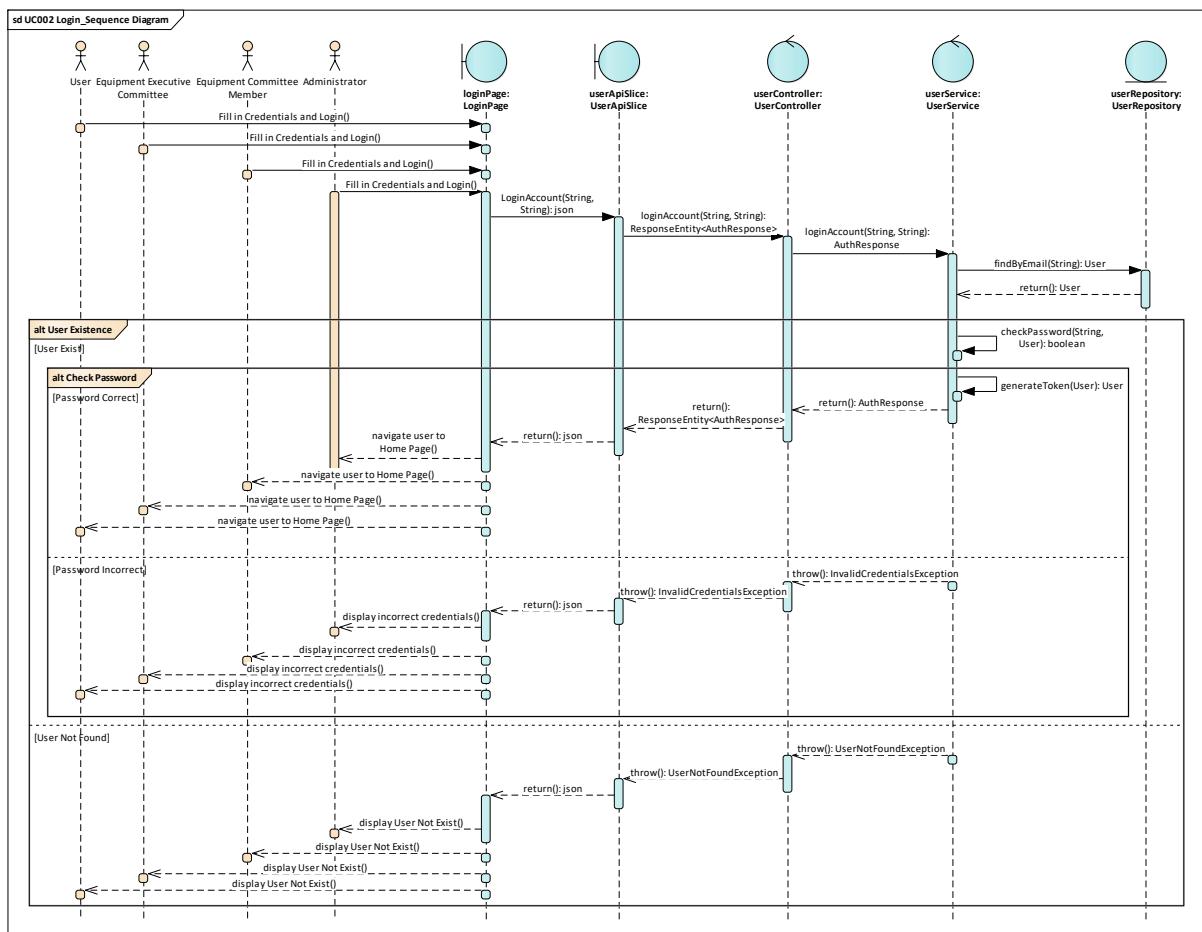


Figure 2.10 Sequence Diagram for UC002 Register an Account

2.3.3 UC003: Reset Password

2.3.3.1 Use Case Description of UC003

Table 2.5 Table of Use Case Description for UC003 Reset Password

Use Case ID	US003
Use Case Name	Reset Password
Description	This use case allows User who forgot their password to reset it via email verification.
Actor(s)	User, Administrator, Equipment Executive Committee and Equipment Committee Member.
Pre-condition(s)	User has a registered email or account in the system.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The user clicks "Forgot Password?" button on the login page. 2. The user enters their registered email address and clicks "Send Reset Link." Button. 3. The system validates the email format and checks for account existence. 4. The system sends an email for password reset. 5. The user opens the email and enters the new password. 6. The system confirms password reset and display "Password updated. Please log in."
Alternative Flow(s) - AF	AF1. Resend Password Reset Email <ol style="list-style-type: none"> 1. User clicks "Resend Email." 2. System sends a new email for password reset.
Exception Flow(s) - EF	EF1. Non-existent Email <ol style="list-style-type: none"> 1. System shall display "Invalid email or password." 2. The user re-enter credentials in the input field. EF2. Inactive Account <ol style="list-style-type: none"> 1. System displays "Account has not been verified."
Post-condition(s)	Password is updated. All active sessions are terminated.

2.3.3.2

Activity Diagram of UC003

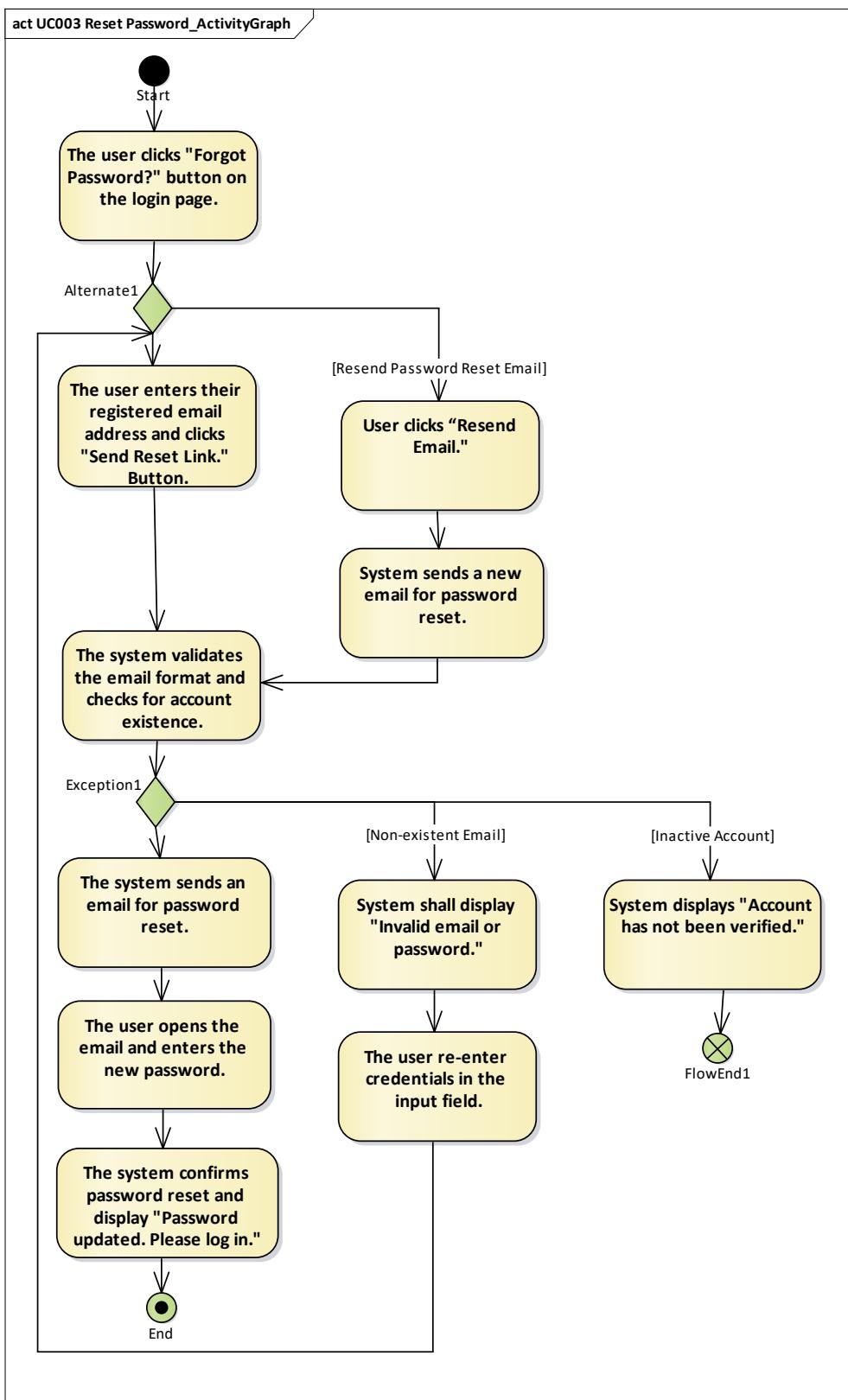


Figure 2.11 Activity Diagram for UC003 Reset Password

2.3.3.3 Sequence Diagram of UC003

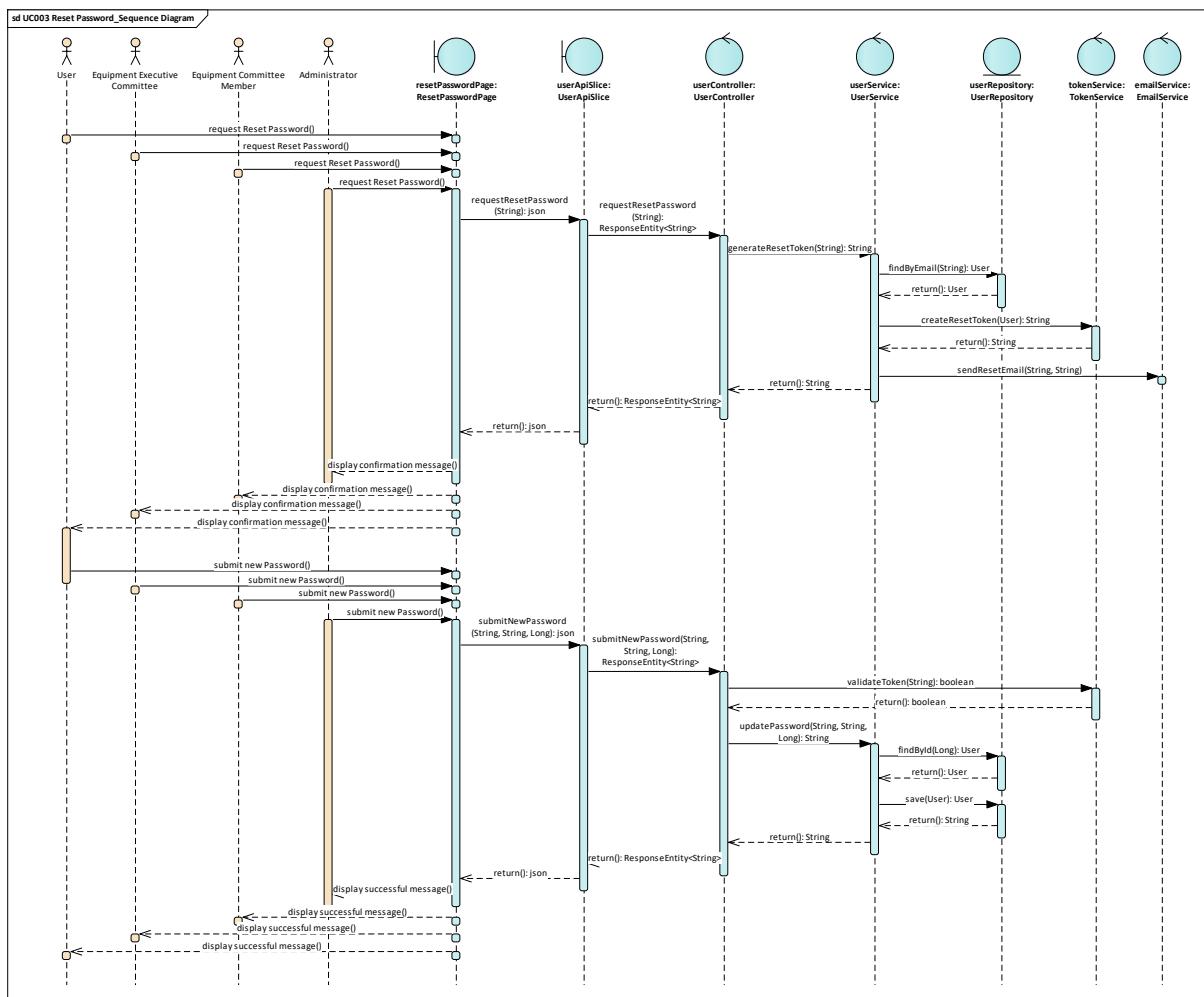


Figure 2.12 Activity Diagram for UC003 Reset Password

2.3.4 UC004: View Equipment List

2.3.4.1 Use Case Description of UC004

Table 2.6 Table of Use Case Description for UC004 View Equipment List

Use Case ID	UC004
Use Case Name	View Equipment List
Description	This use case displays a list of all available equipment details such as name, quantity, status, and condition, allowing User and Equipment Committee Member to view it.
Actor(s)	User and Equipment Committee Member.
Pre-condition(s)	User is logged into an account.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The actor navigates to the "Equipment List" page via the main menu. 2. The system fetches equipment data from the database. 3. The system displays the equipment list along with its details such as name, category, quantity, status and renting price. 4. The system displays action button "Rent Now." 5. The actor clicks "View Details" on an item to see more details.
Alternative Flow(s) - AF	AF1. Filter Equipment by Categories <ol style="list-style-type: none"> 1. The user applies filters. 2. The system updates the list of equipment based on filters.
Exception Flow(s) - EF	EF1. Error Fetching Data <ol style="list-style-type: none"> 1. System displays warning message, "Error Fetching Equipment Data." 2. The actor refreshes the page EF2. Server Error <ol style="list-style-type: none"> 1. System Display "Server Error. Please refresh this page." 2. The actor refreshes the page.
Post-condition(s)	Equipment list is displayed with name, real-time status, quantity and condition.

2.3.4.2 Activity Diagram of UC004

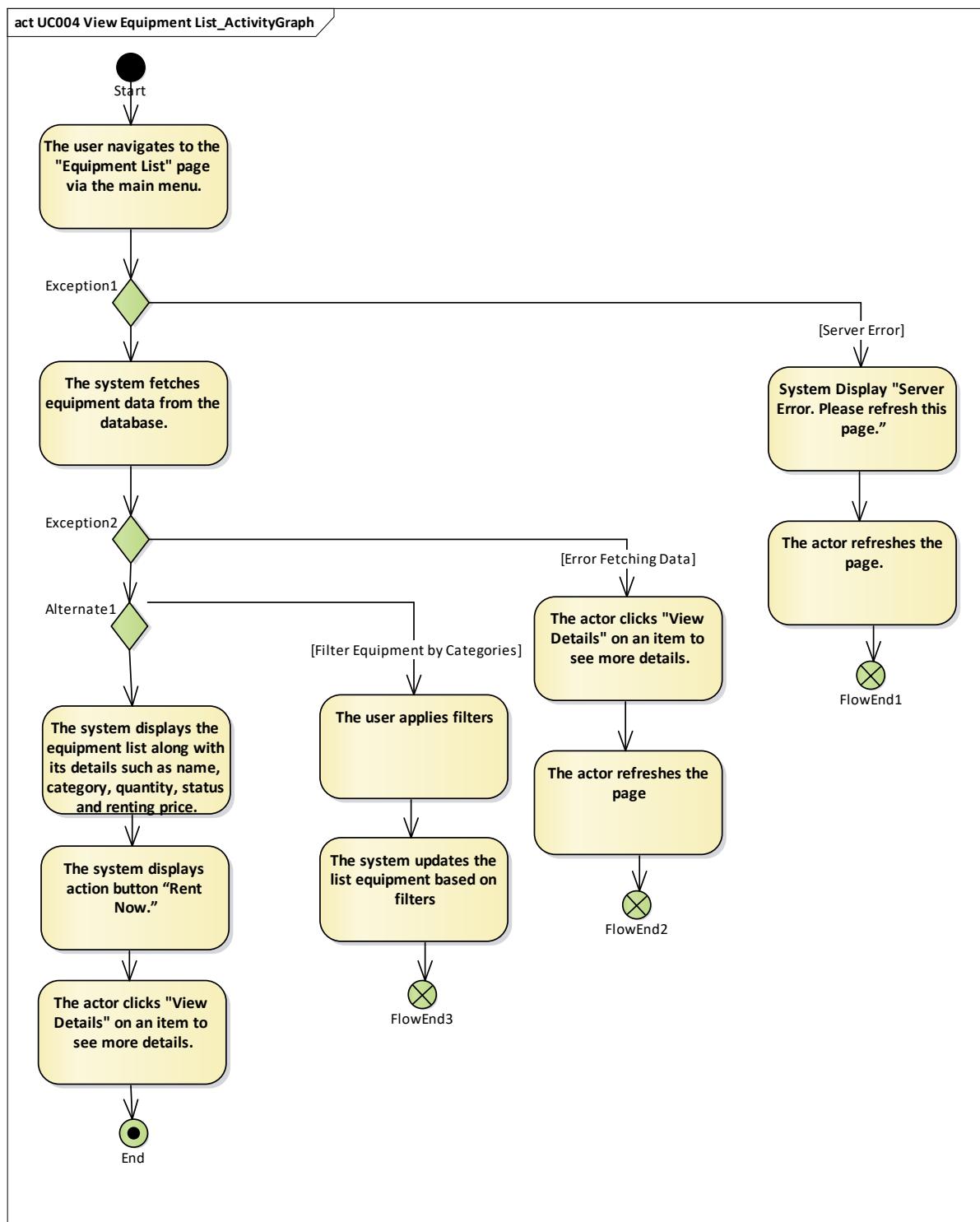


Figure 2.13 Activity Diagram for UC004 View Equipment List

2.3.4.3 Sequence Diagram of UC004

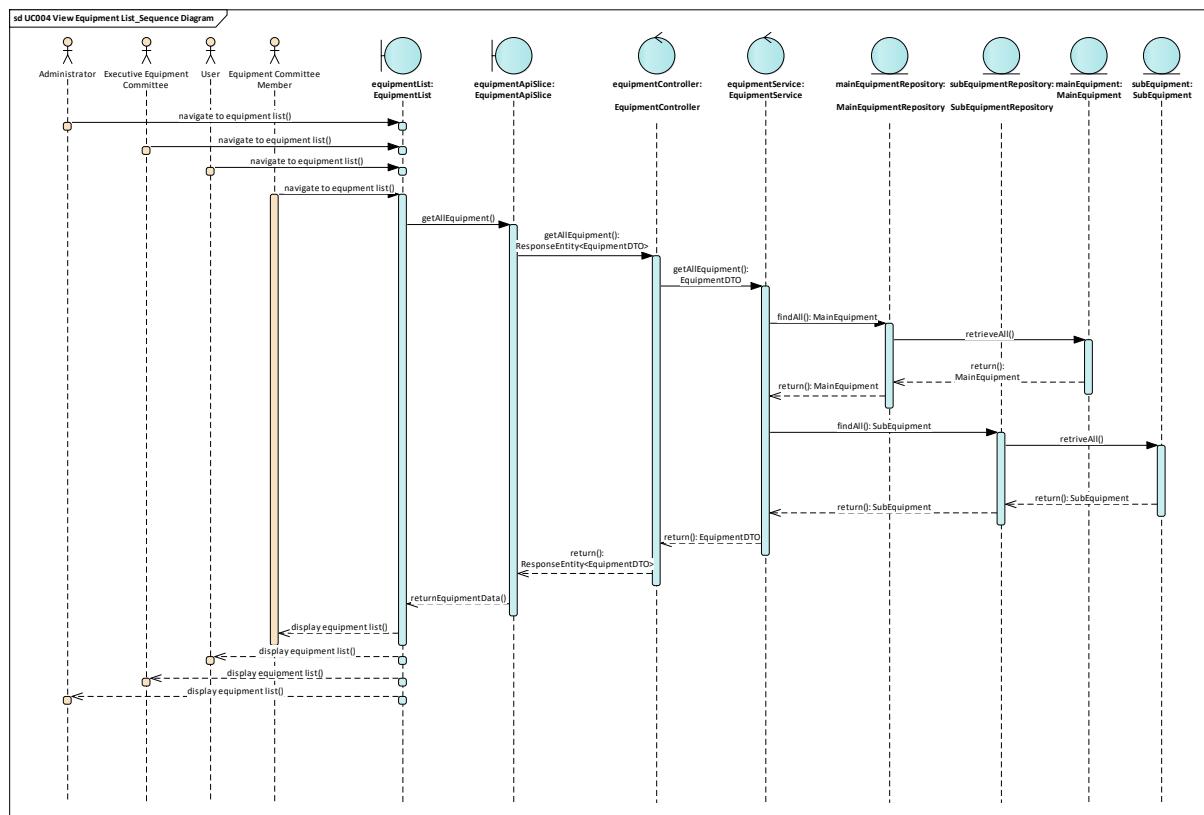


Figure 2.14 Sequence Diagram for UC004 View Equipment List

2.3.5 UC005: Rent Equipment

2.3.5.1 Use Case Description of UC005

Table 2.7 Table of Use Case Description for UC005 Rent Equipment

Use Case ID	UC005
Use Case Name	Rent Equipment
Description	This use case allows actor to rent equipment from Kelab Fotokreatif by filling out digital Club's Equipment Rental Form.
Actor(s)	User, Equipment Executive, Equipment Administrator and Equipment Committee Member.
Pre-condition(s)	<ul style="list-style-type: none"> 1. The actor is logged into an account. 2. The actor must have access to the “Rent Equipment” section.
Normal Flow(s)- NF	<ul style="list-style-type: none"> 1. The actor navigates to “Rent Equipment” section. 2. System displays a rental form. 3. The actor fills in Program name, venue, selects available start date and available end date from calendar and click “Next.” 4. System displays list of equipment (UC004). 5. The actor selects the desired Equipment and click “Next.” 6. The actor read the term and conditions and check the checkbox. 7. The actor uploads a picture of signature and click “Next.” 8. System displays formal Club's Equipment Rental Form that summarise entered details. 9. The actor is required to view and fill in required rest of the personal details. 10. The actor submit the rental form to Equipment Executive Committee for further review (UC013). 11. System saves the rental form details. 12. After the Club's Equipment Rental Form has been approved, the actor is required to make payment (UC007).
Alternative Flow(s) - AF	AF1: Incomplete Form Submission <ul style="list-style-type: none"> 1. If the actor skips any required fields or fails to upload a signature.

	<p>2. The system prompts the actor to complete all required steps before proceeding.</p> <p>AF2: Actor Cancels Request</p> <ol style="list-style-type: none"> 1. The actor can cancel the process. 2. The actor is redirected back to the main page.
Exception Flow(s) - EF	<p>EF1. Error Fetching Data</p> <ol style="list-style-type: none"> 1. System displays warning message, "Error Fetching Equipment Data." 2. The actor refreshes the page.
Post-condition(s)	<ol style="list-style-type: none"> 1. The rental form is submitted and stored in the system. 2. The rental form status will be set to "Pending Approval" 3. Upon approval, the actor will proceed to make payment as per UC007.

2.3.5.2

Activity Diagram of UC005

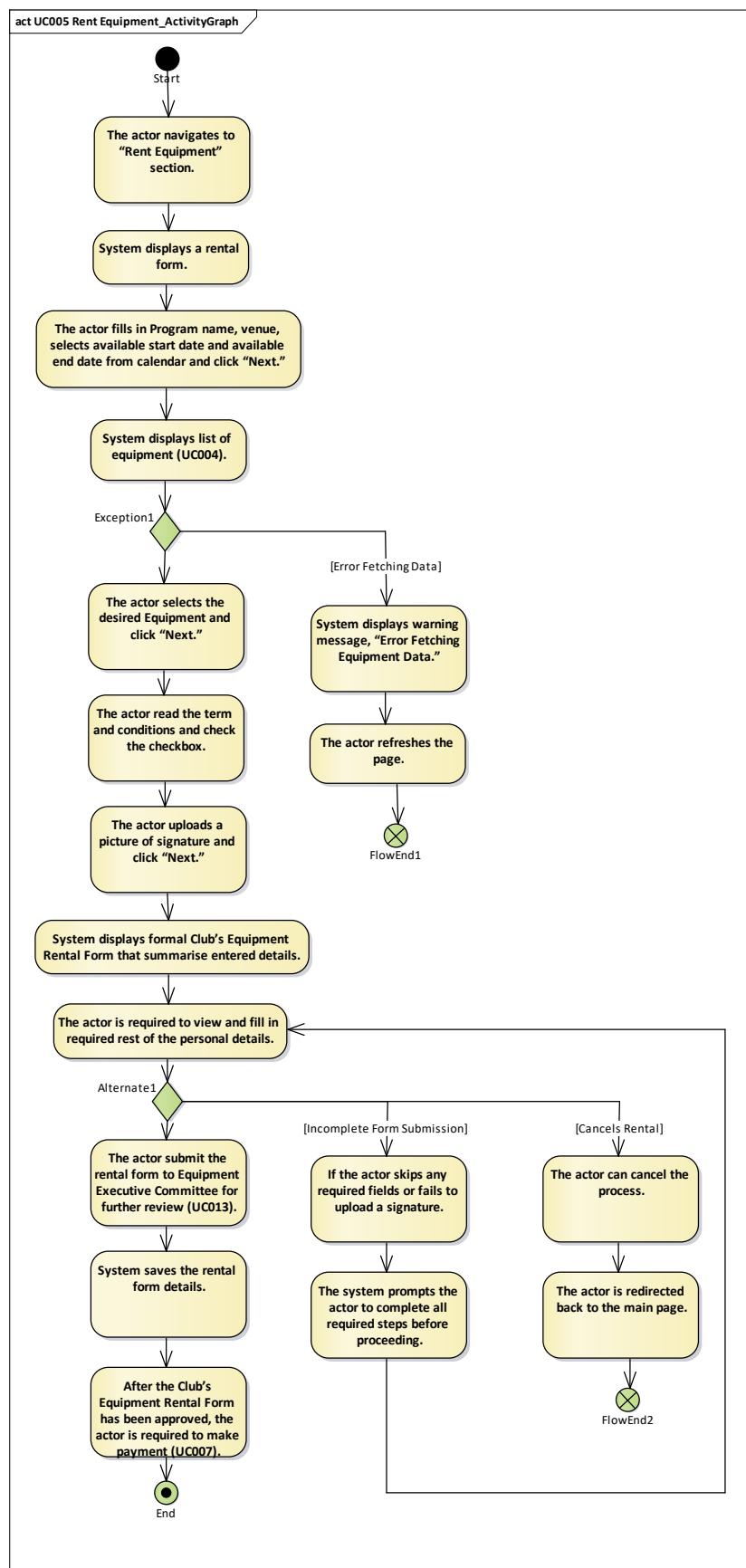


Figure 2.15 Activity Diagram for UC005 Rent Equipment

2.3.5.3 Sequence Diagram of UC005

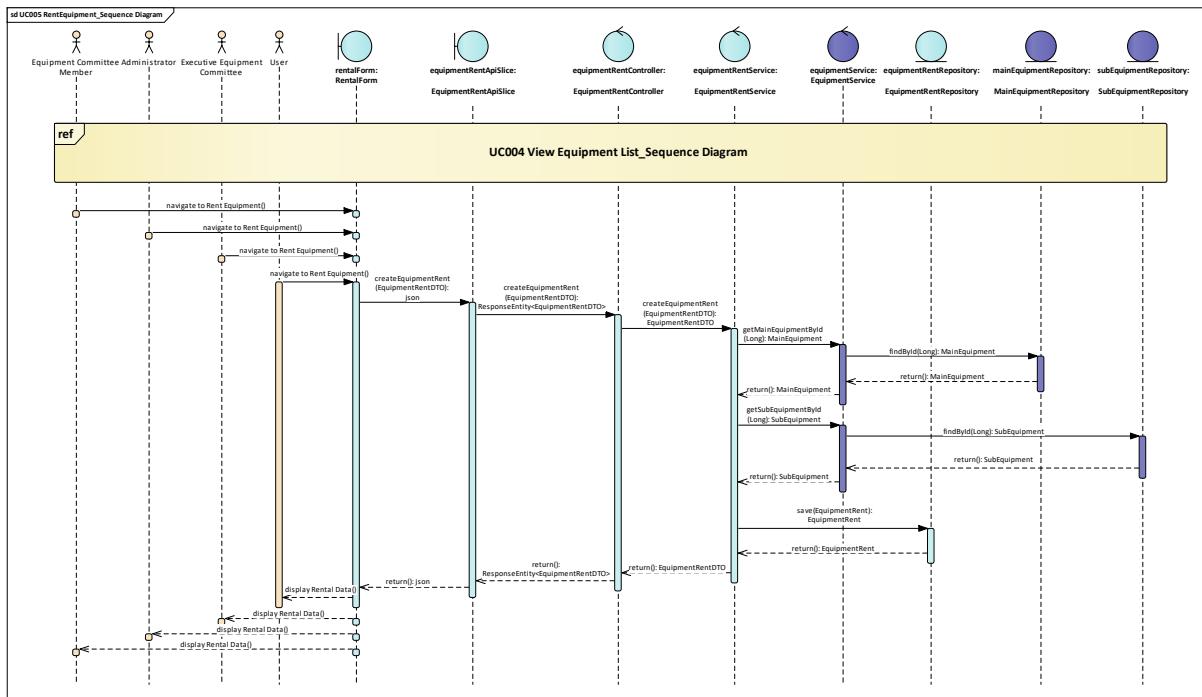


Figure 2.16 Sequence Diagram for UC005 Rent Equipment

2.3.6 UC006: Return Equipment

2.3.6.1 Use Case Description of UC006

Table 2.8 Table of Use Case Description for US006 Return Equipment

Use Case ID	UC006
Use Case Name	Return Equipment
Description	This use case facilitates the return process of rented by User and borrowed equipment by the Equipment Committee Member and update the inventory.
Actor(s)	User, Equipment Executive Equipment, Administrator and Equipment Committee Member.
Pre-condition(s)	<ul style="list-style-type: none"> 1. Actor has active equipment rental and borrowed.
Normal Flow(s)- NF	<ul style="list-style-type: none"> 1. The actor navigates to "My Rentals" and selects the active rental. 2. The actor clicks "Return Equipment" and upload picture of the returning equipment. 3. The actor confirms the return of each equipment through checkboxes. 4. The actor can leave some optional notes and submit the returns. 5. The system verifies the return date. 6. The system shall send the return details to the Equipment Executive Committee for review (UC014).
Alternative Flow(s) - AF	AF1. Late Return <ul style="list-style-type: none"> 1. The actor returns the equipment after return date. 2. The system will calculate extra fee as punishment. 3. The system displays extra fees to be paid and redirect actor to payment gateway (UC007).
Exception Flow(s) - EF	EF1. Error Fetching Rental Data <ul style="list-style-type: none"> 1. System displays warning message, "Error Fetching Rental Data." 2. The actor refreshes the page.

Post-condition(s)	Rental marked as "Returned" in the system. Equipment status updated to "Available" or "Unavailable" depending on the condition.
--------------------------	---------------------------------------------------------------------------------------------------------------------------------

2.3.6.2

Activity Diagram of UC006

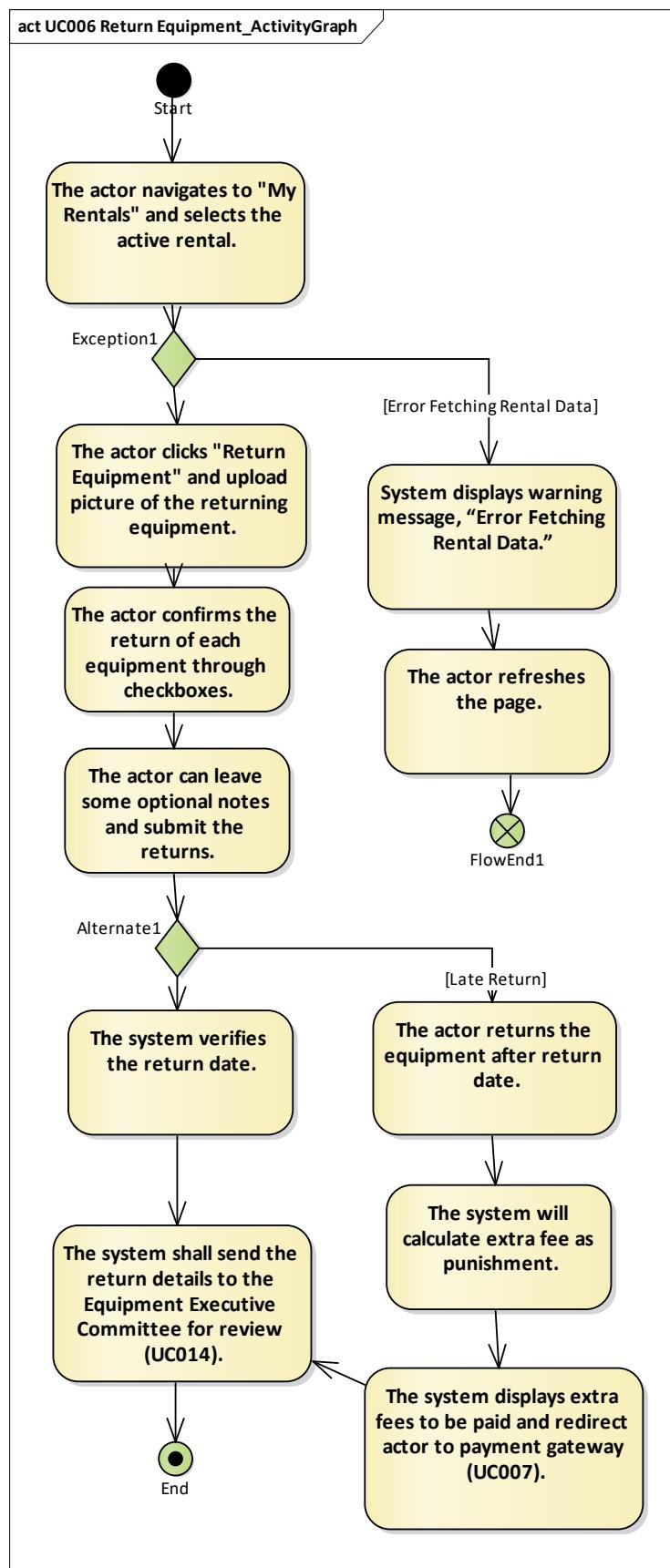


Figure 2.17 Activity Diagram for UC006 Return Equipment

2.3.6.3 Sequence Diagram of UC006

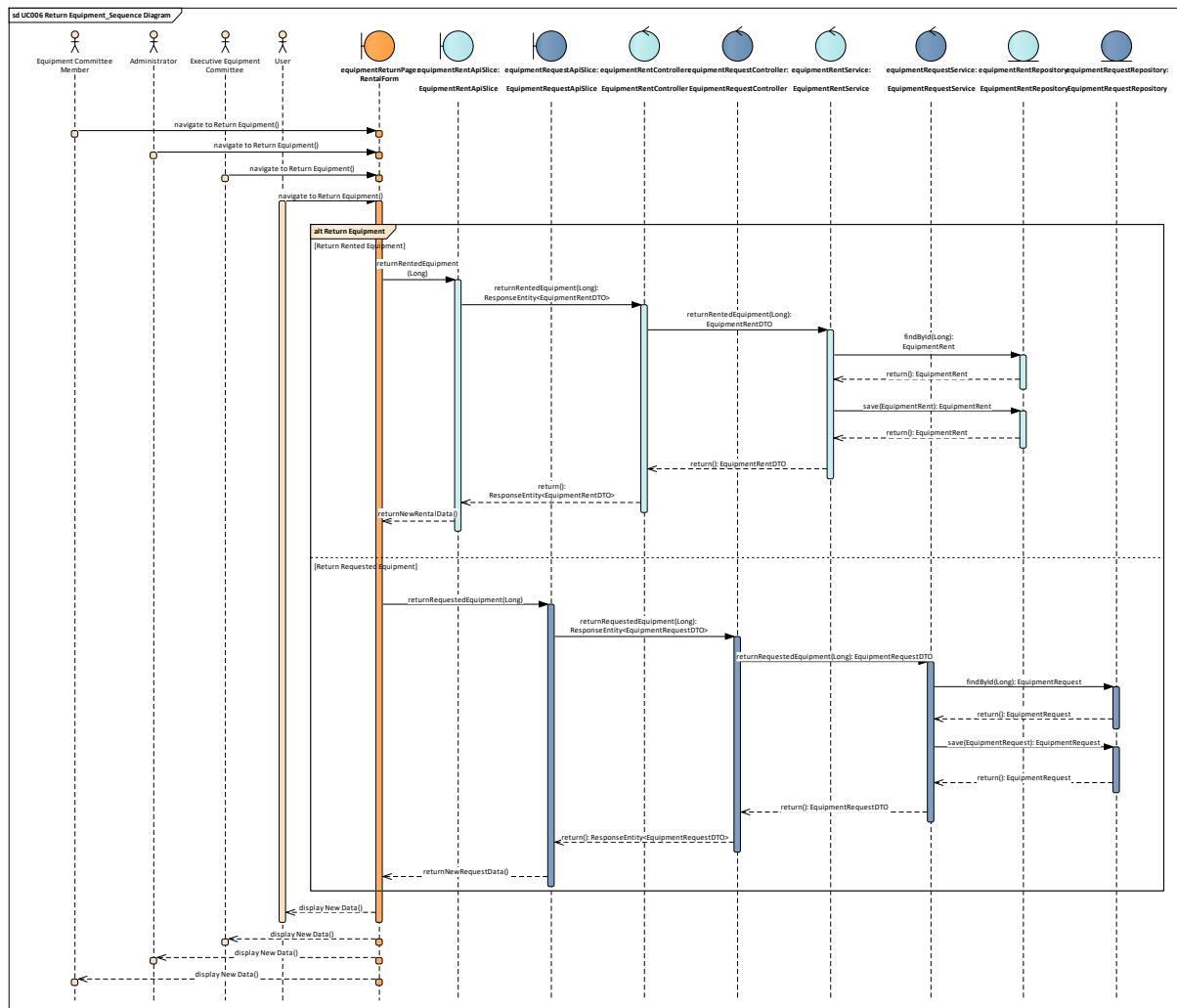


Figure 2.18 Sequence Diagram for UC006 Return Equipment

2.3.7 UC007: Make Payment

2.3.7.1 Use Case Description of UC007

Table 2.9 Table of Use Case Description for UC007 Make Payment

Use Case ID	UC007
Use Case Name	Make Payment
Description	This use case allows User to make payments for equipment rentals through integrated payment gateways.
Actor(s)	User, Equipment Committee Member, Equipment Executive Member, and Administrator.
Pre-condition(s)	<ol style="list-style-type: none"> 1. The actor has reached checkout with valid rental selection. 2. System has calculated total fees.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The actor proceeds to the payment page after selecting equipment (UC005) and reached checkout page. 2. System displays total rental cost and payment options. 3. The actor selects a payment method such as credit/debit card, online banking, or e-wallet. 4. The system redirects the user to payment page. 5. The actor enters required payment details 6. The actor click confirm to initiate the payment process. 7. System displays payment confirmation and generate invoice (UC008).
Exception Flow(s) - EF	<p>EF1. Payment declined</p> <ol style="list-style-type: none"> 1. If Payment fails due to insufficient funds. 2. System displays error message and prompt user to try again. <p>EF2. Invalid payment information</p> <ol style="list-style-type: none"> 1. If user enters incorrect card number or CVV 2. System prompts user to re-enter card details.
Post-condition(s)	Equipment status changes to "Rented" and user paid for the equipment rental.

2.3.7.2

Activity Diagram of UC007

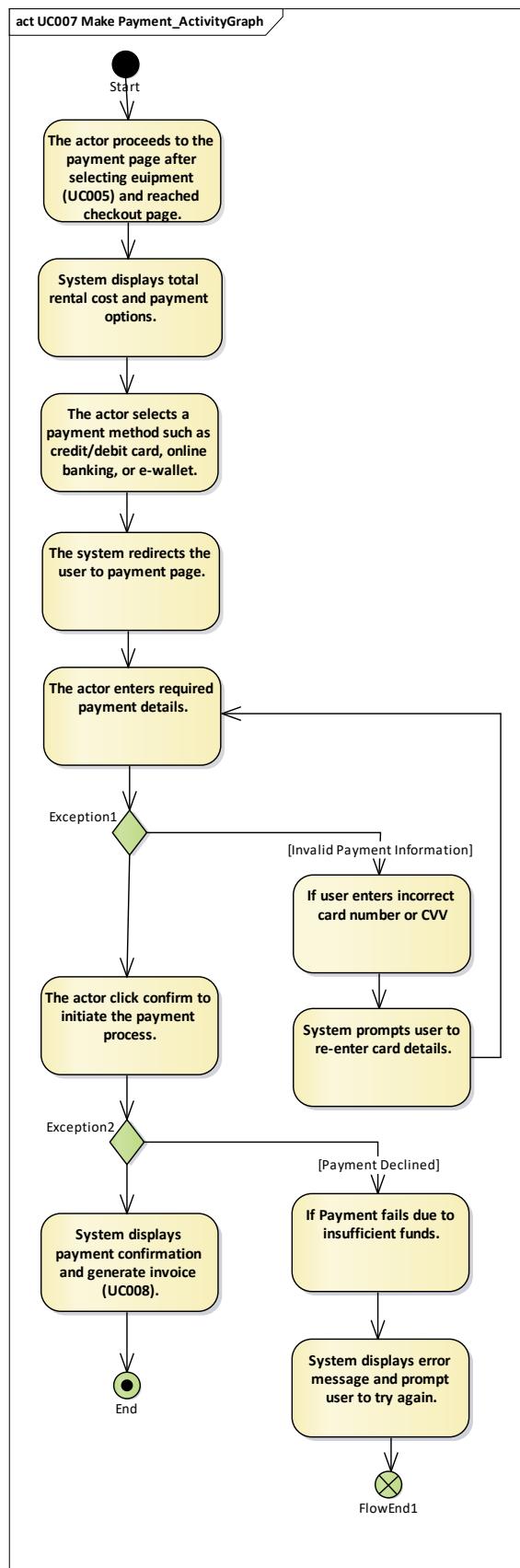


Figure 2.19 Activity Diagram for UC007 Make Payment

2.3.7.3 Sequence Diagram of UC007

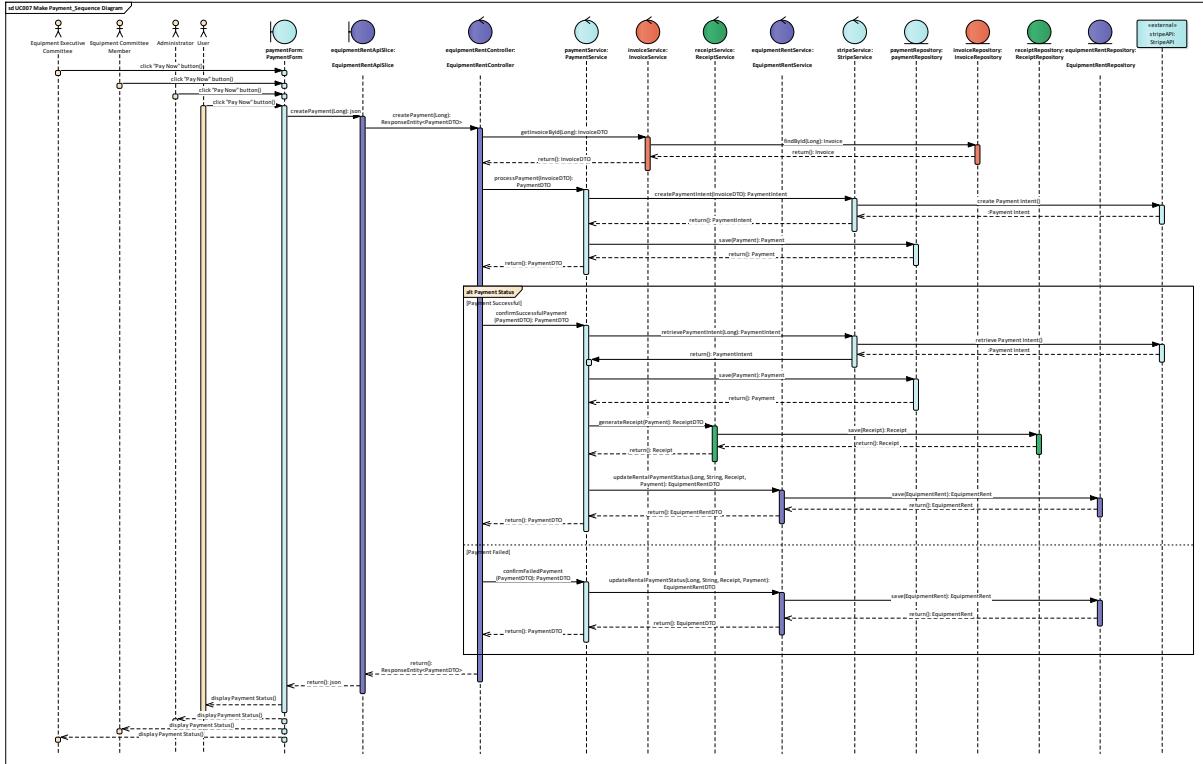


Figure 2.20 Sequence Diagram for UC007 Make Payment

2.3.8 UC008: View Generated Invoice and Receipt

2.3.8.1 Use Case Description of UC008

Table 2.10 Table of Use Case Description for UC008 View Generated Invoice and Receipt

Use Case ID	UC008
Use Case Name	View Generated Invoice and Receipt
Description	This use case allows system view the invoice generated after Rent Equipment (UC005) has been approved (UC012) and view the receipt after make payment (UC007) .
Actor(s)	User, Equipment Committee Member, Equipment Executive Member, and Administrator.
Pre-condition(s)	<ol style="list-style-type: none"> 1. The actor has rent equipment (UC005) has been approved (UC012). 2. The actor has successfully paid for the equipment rental. (UC008).
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The actor navigates to the Rent Equipment section. 2. The system displays a list of completed Equipment Rent transactions with invoice and receipt links. 3. The actor selects to view Invoice or receipt for a specific Equipment Rent. 4. The system retrieves and displays the full details of the invoice.
Alternative Flow(s) - AF	AF1. Invoice or Receipt Not Found <ol style="list-style-type: none"> 1. The system cannot find an invoice or receipt for the selected Equipment Rent. 2. The system displays an error message: "Invoice or receipt is not yet available for this rental."
Exception Flow(s) - EF	EF1. Network failure or server error while retrieving data. <ol style="list-style-type: none"> 1. The system failed to retrieve invoice or receipt for the selected Equipment Rent. 2. The system displays an error message: "Server Error. Please Try Again."

Post-condition(s)	<ol style="list-style-type: none"> 1. The Renter successfully views the associated invoice and receipt details for a rental. 2. No data is modified during this process.
--------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.3.8.2 Activity Diagram of UC008

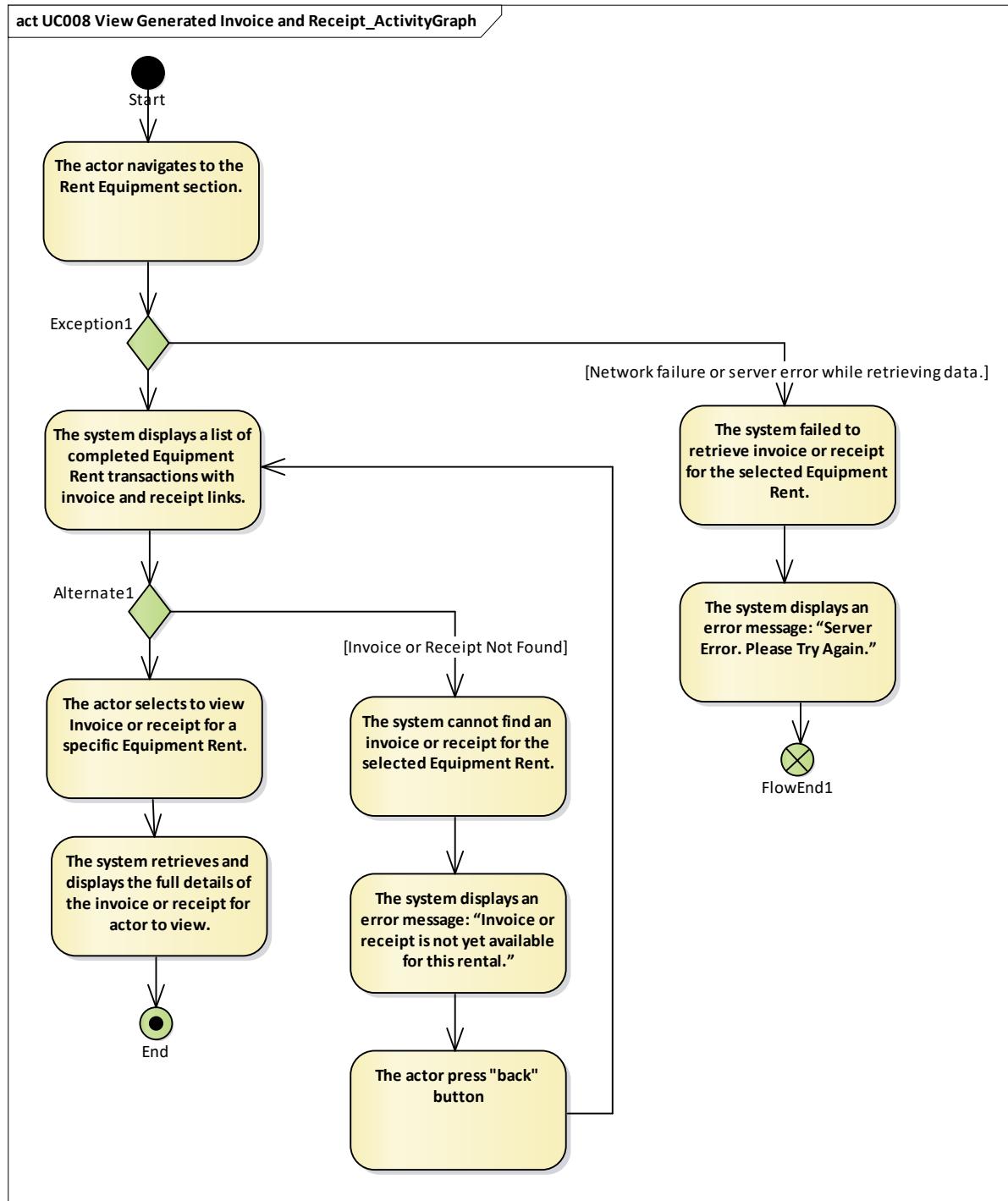


Figure 2.21 Activity Diagram for UC008 Generate Invoice

2.3.8.3 Sequence Diagram of UC008

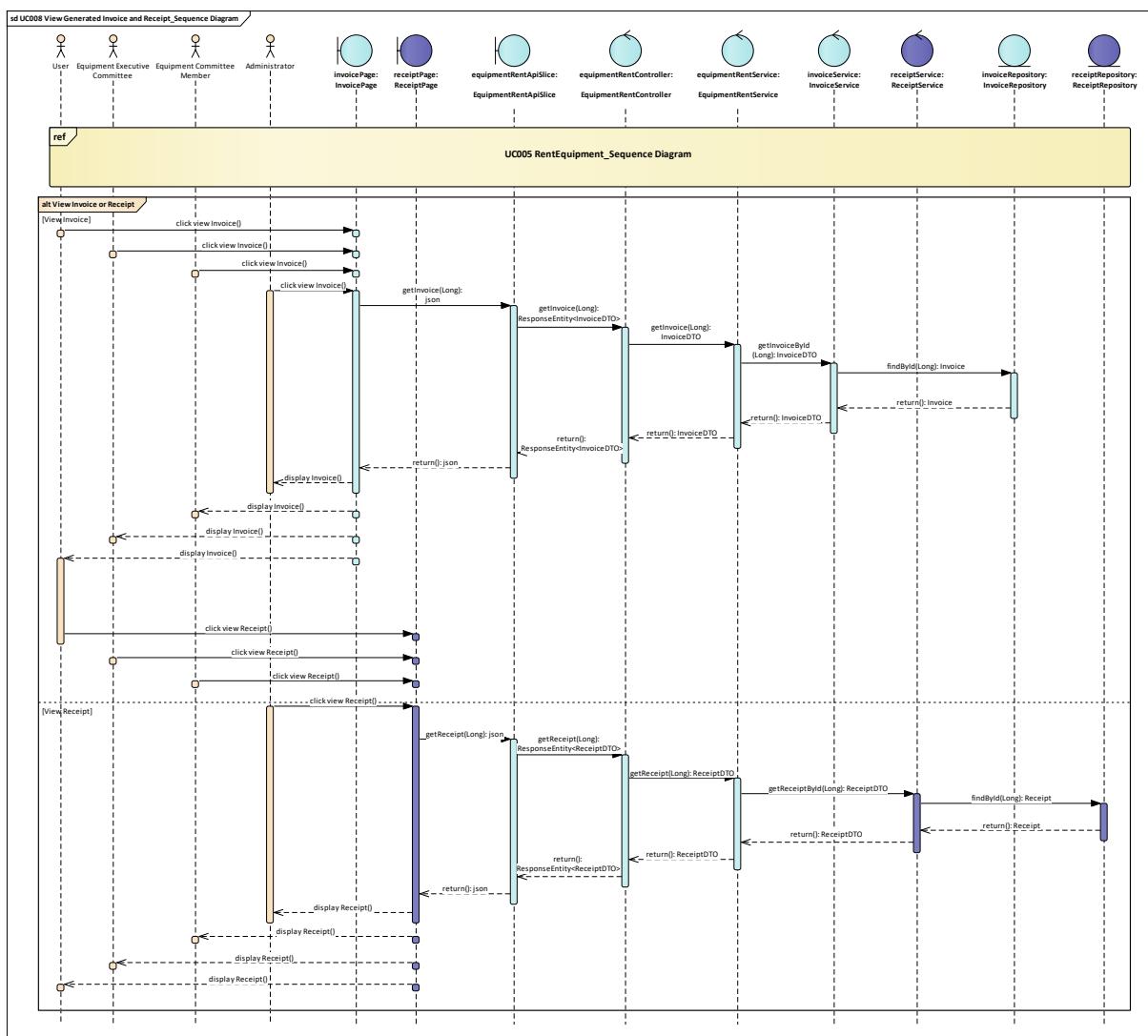


Figure 2.22 Sequence Diagram for UC008 Generate Invoice

2.3.9 UC009: Request Equipment

2.3.9.1 Use Case Description of UC009

Table 2.11 Table of Use Case Description for UC009 Request Equipment

Use Case ID	UC009
Use Case Name	Request Equipment
Description	This use case allows Equipment Committee Member to request equipment in advance for planned events or activities.
Actor(s)	Equipment Committee Member
Pre-condition(s)	<ol style="list-style-type: none"> 1. The Equipment Committee Member must be logged into the system. 2. The event created by Administrator in the system (UC016). 3. The event already assigned to the specified Equipment Committee Member (UC017).
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. Equipment Committee Member navigates to “Request Equipment” section. 2. System displays a list of events. 3. Equipment Committee Member select the assigned event. 4. System displays an equipment request form that prompts Equipment Committee Member to fill in information. 5. Equipment Committee Member selects start date and end date from calendar and click “Next.” 6. System displays list of equipment (UC004). 7. Equipment Committee Member selects the desired Equipment and click “Next.” 8. System displays formal Club’s Equipment Request Form that summarise entered details. 9. Equipment Committee Member can to view or edit the required rest of the personal details. 10. Equipment Committee Member submit the request form to Equipment Executive Committee for further review (UC013). 11. System saves the request form details.
Alternative Flow(s) - AF	AF1: Incomplete Form Submission

	<ol style="list-style-type: none"> 1. If Equipment Committee Member skips any required fields or fails to upload a signature. 2. The system prompts the actor to complete all required steps before proceeding. <p>AF2: Cancels Request</p> <ol style="list-style-type: none"> 1. Equipment Committee Member can cancel the process. 2. Equipment Committee Member is redirected back to the main page.
Exception Flow(s) - EF	<p>EF1. Error Fetching Data</p> <ol style="list-style-type: none"> 1. System displays warning message, "Error Fetching Equipment Data." 2. Equipment Committee Member refreshes the page.
Post-condition(s)	An equipment request form is submitted and wait for Equipment Executive Committee approval.

2.3.9.2

Activity Diagram of UC009

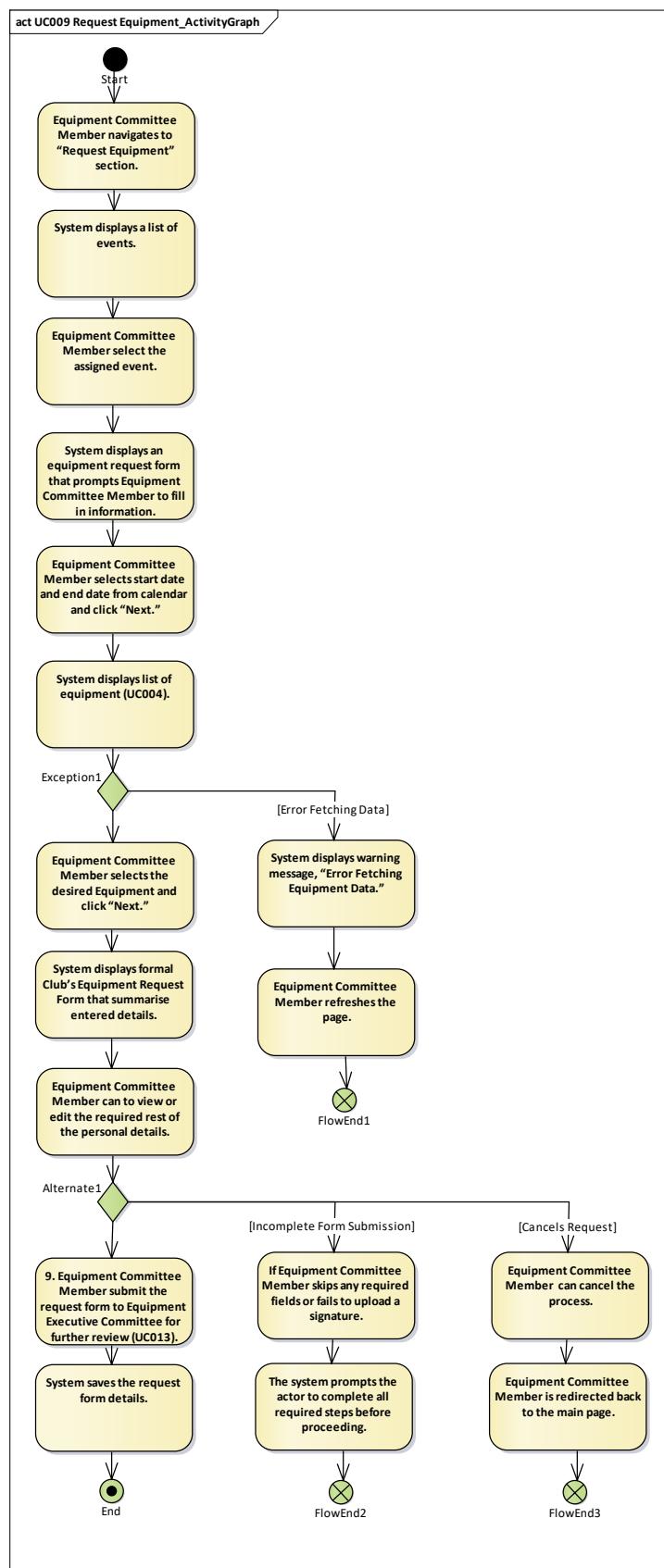


Figure 2.23 Activity Diagram for UC009 Request Equipment

2.3.9.3 Sequence Diagram of UC010

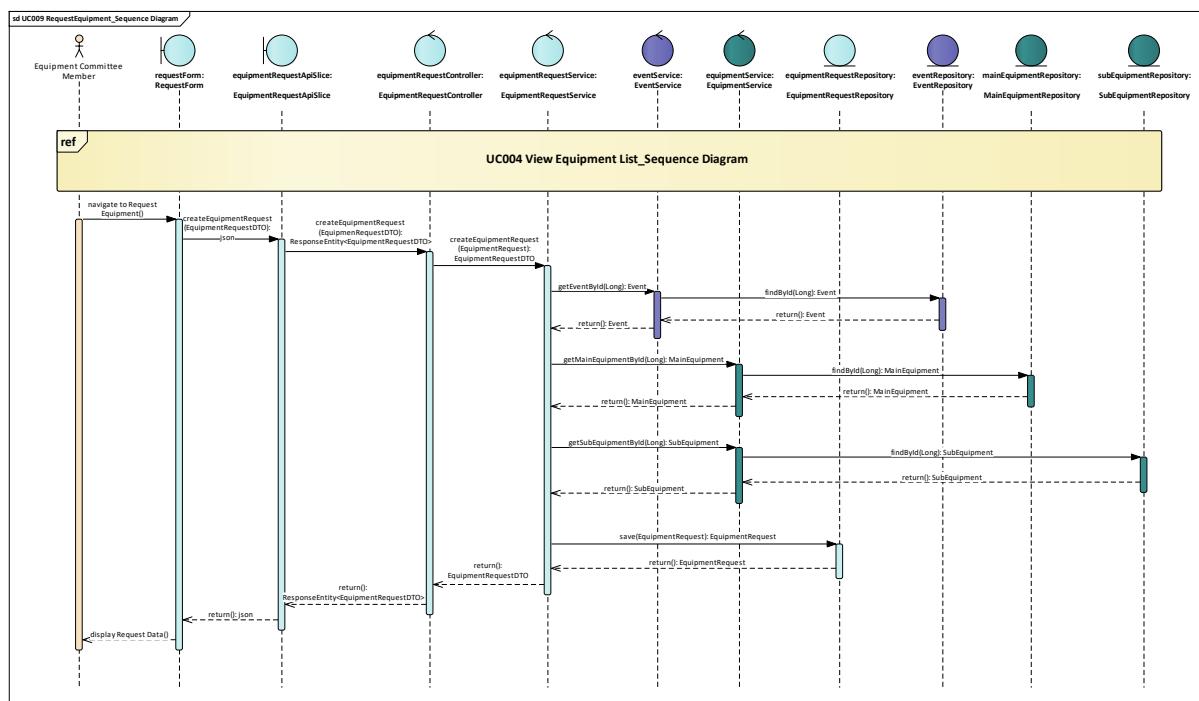


Figure 2.24 Sequence Diagram for UC009 Request Equipment

2.3.10 UC010: Monitor Equipment

2.3.10.1 Use Case Description of UC010

Table 2.12 Table of Use Case Description for UC010 Monitor Equipment

Use Case ID	UC010
Use Case Name	Monitor Equipment
Description	This use case enables the Equipment Executive Committee and Administrator to track equipment availability, status, and conditions.
Actor(s)	Equipment Executive Committee and Administrator
Pre-condition(s)	<ol style="list-style-type: none"> 1. The actor has logged into the system. 2. Equipment data exists in the system.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The actor navigates to the “Monitor Equipment” dashboard. 2. The system displays a list of equipment items with its details such as equipment ID, name, category, quantity, status and conditions, last rented date, return date, assigned event/user and history button. 3. The actor selects a specific item to view detailed information.
Alternative Flow(s) - AF	<p>AF1. Filter Choices</p> <ol style="list-style-type: none"> 1. The actor uses filters to find desired categories of the equipment. <p>AF2. Search Keywords</p> <ol style="list-style-type: none"> 1. The actor enters search keywords to search the desired equipment from the list.
Exception Flow(s) - EF	<p>EF1. Error Fetching Data</p> <ol style="list-style-type: none"> 1. The system displays an error message, “Failed to Fetch Data.” 2. The system prompts actor to reload page.
Post-condition(s)	The actor successfully views list of detailed equipment status and conditions.

2.3.10.2 Activity Diagram of UC010

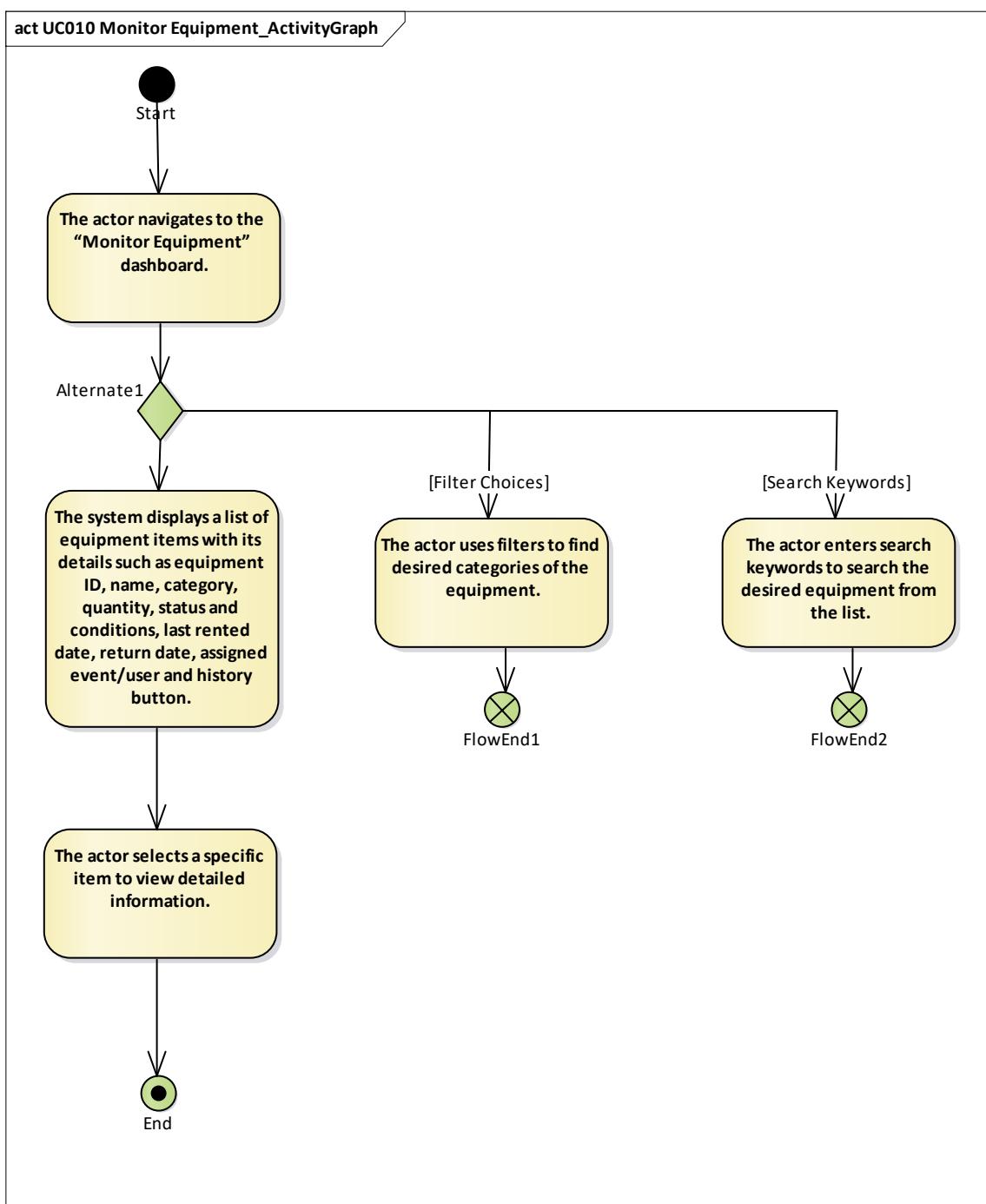


Figure 2.25 Activity Diagram for UC010 Monitor Equipment

2.3.10.3 Sequence Diagram of UC010

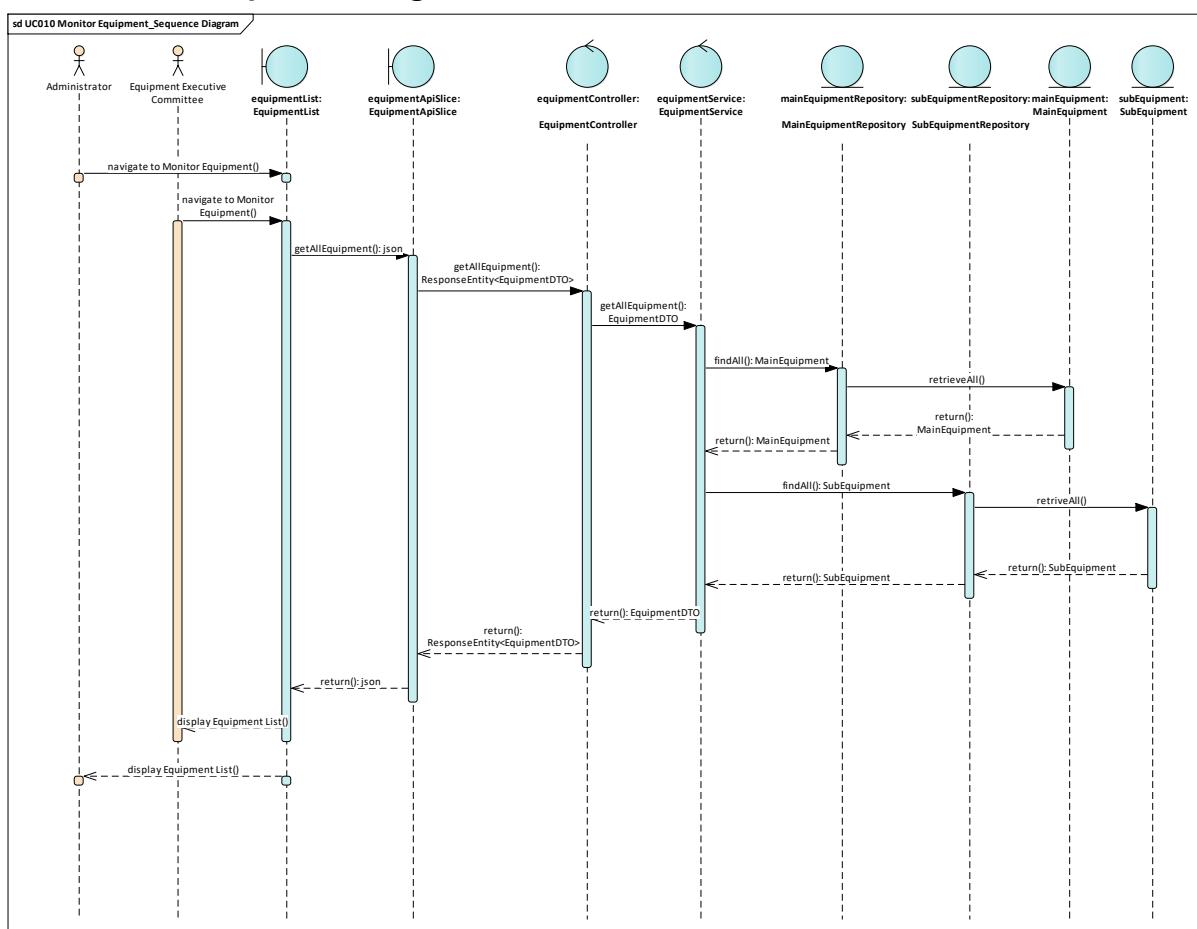


Figure 2.26 Sequence Diagram for UC010 Monitor Equipment

2.3.11 UC011: Update Equipment

2.3.11.1 Use Case Description of UC011

Table 2.13 Table of Use Case Description for UC011 Update Equipment

Use Case ID	UC011
Use Case Name	Update Equipment
Description	This use case allows Equipment Executive Committee to add, edit, or delete equipment details in the system.
Actor(s)	Equipment Executive Committee.
Pre-condition(s)	<ol style="list-style-type: none"> 1. The Equipment Executive Committee is logged into the system. 2. The user has been assigned as Equipment Executive Committee by Administrator.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. The Equipment Executive Committee navigates to the “Equipment Management” page. 2. The system displays the list of all equipment (UC010). 3. The Equipment Executive Committee can choose to add, edit, and delete equipment. <ul style="list-style-type: none"> a. Add New Equipment (AF1) b. Edit Existing Equipment (AF2) c. Delete Equipment (AF3) 4. The system validates and applies the changes. 5. The system displays “Update Successful” message and updates the list accordingly.
Alternative Flow(s) - AF	<p>AF1. Add New Equipment</p> <ol style="list-style-type: none"> 1. Equipment Executive Committee selects the “Add Equipment” option. 2. The system displays a form for entering equipment details (name, category, quantity, condition, image, etc.). 3. The Equipment Executive Committee fills in the form and submits. <p>AF2. Edit Existing Equipment</p>

	<ol style="list-style-type: none"> 1. The Equipment Executive Committee selects the “Edit” option for a specific equipment item. 2. The system displays the existing details in an editable form. 3. The Equipment Executive Committee modifies the necessary fields and submits. <p>AF3. Delete Equipment</p> <ol style="list-style-type: none"> 1. The Equipment Executive Committee selects the “Delete” option for a specific equipment item. 2. The system asks for confirmation. 3. The Equipment Executive Committee click “Confirm.”
Exception Flow(s) - EF	<p>EF1. Deletion of Assigned Equipment</p> <ol style="list-style-type: none"> 1. The system displays warning message, “This item cannot be deleted until it is returned.”
Post-condition(s)	Equipment information is updated in the system database.

2.3.11.2 Activity Diagram of UC011

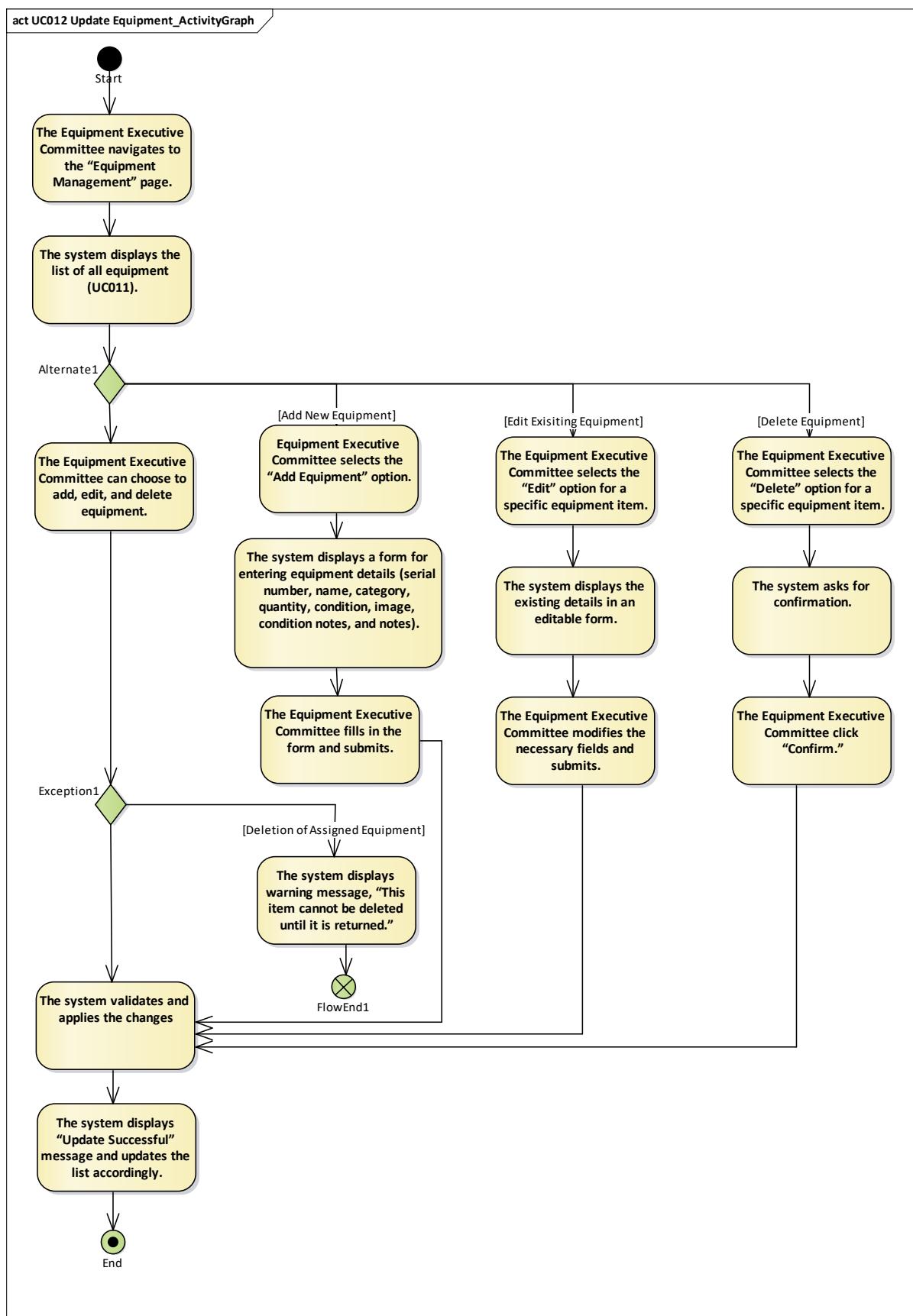


Figure 2.27 Activity Diagram for UC011 Update Equipment

2.3.11.3 Sequence Diagram of UC011

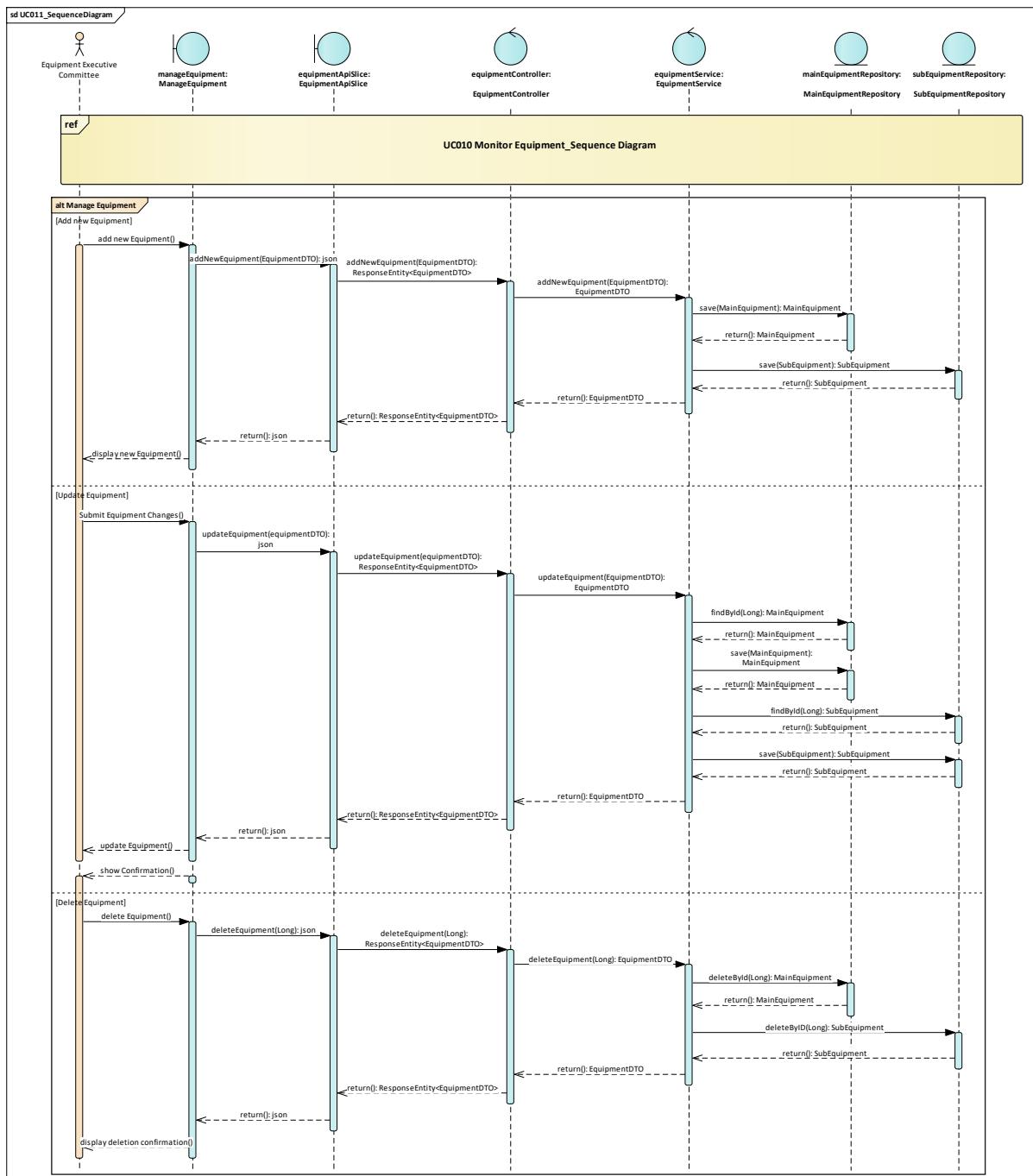


Figure 2.28 Sequence Diagram for UC011 Update Equipment

2.3.12 UC012: Manage Equipment Requests and Rentals

2.3.12.1 Use Case Description of UC012

Table 2.14 Table of Use Case Description for UC012 Manage Equipment Requests and Rentals

Use Case ID	UC012
Use Case Name	Manage Equipment Requests and Rentals
Description	This use case enables the Equipment Executive Committee to approve, reject, or manage incoming equipment request for events and rentals.
Actor(s)	Equipment Executive Committee.
Pre-condition(s)	<ol style="list-style-type: none"> 1. The Equipment Executive Committee is logged into the system. 2. The user has been assigned as Equipment Executive Committee by Administrator.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. Equipment Executive Committee navigates to the “Equipment Requests and Rentals” section. 2. Equipment Executive Committee views a list of pending equipment requests and rentals. 3. Equipment Executive Committee selects a request or rental to view its detailed information for review. 4. Equipment Executive Committee choose specific equipment based on the desired equipment list. 5. Equipment Executive Committee approves or approves with changes or rejects the equipment request or rental. 6. The system updates the request or rental status accordingly. 7. System sends notification to the Equipment Committee Member who requested equipment for a specific event or any user that rent the equipment.
Alternative Flow(s) - AF	<p>AF1. Filter Equipment Requests</p> <ol style="list-style-type: none"> 1. The Equipment Executive Committee uses filters to find desired categories of the equipment. <p>AF2. Search Equipment Requests by keywords</p> <ol style="list-style-type: none"> 1. The actor Equipment Executive Committee search keywords to search the event name, date, requester, or the equipment itself.

	<p>AF3. Selects Equipment Committee Member Equipment Request</p> <ol style="list-style-type: none"> 1. Equipment Executive Committee selects a request to view its detailed information for review. 2. Equipment Executive Committee monitor a list of equipment (UC010). 3. Equipment Executive Committee can edit or change specific equipment requested by Equipment Committee Member. 4. Equipment Executive Committee approves or rejects the equipment request. 5. The system updates the request status accordingly. 6. System sends notification to the Equipment Committee Member who requested equipment for a specific event.
Exception Flow(s) - EF	<p>EF1. Failed Data Retrieval</p> <ol style="list-style-type: none"> 1. The system displays an error message, “Fail to Retrieve Data” 2. The system prompts to refresh page.
Post-condition(s)	<ol style="list-style-type: none"> 1. The request or rental is either approved or rejected. 2. If the request or rental has been approved, the equipment status will be updated to reserve for the specified event.

2.3.12.2 Activity Diagram of UC012

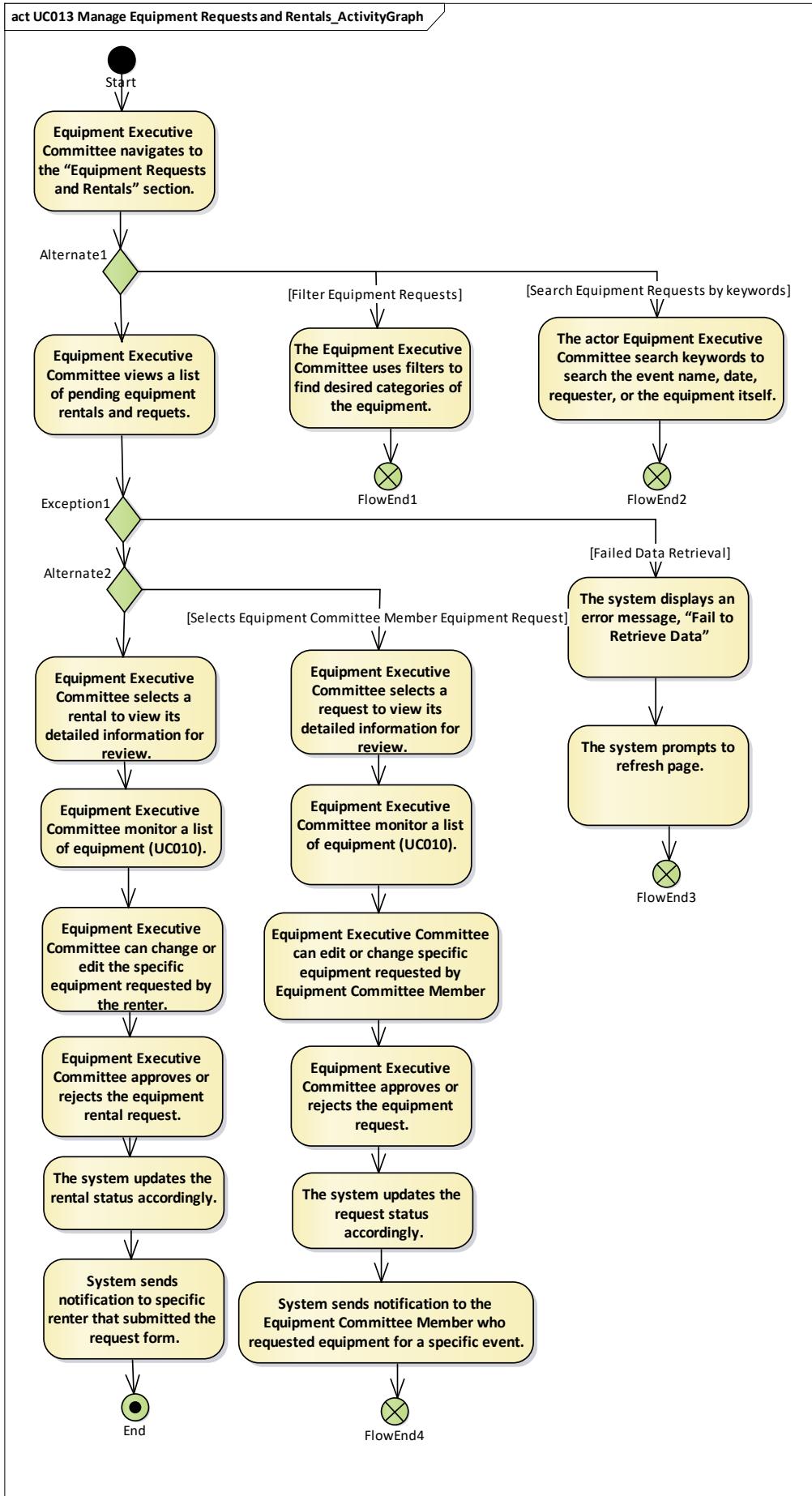


Figure 2.29 Activity Diagram for UC012 Manage Equipment Requests and Rentals

2.3.12.3 Sequence Diagram of UC012

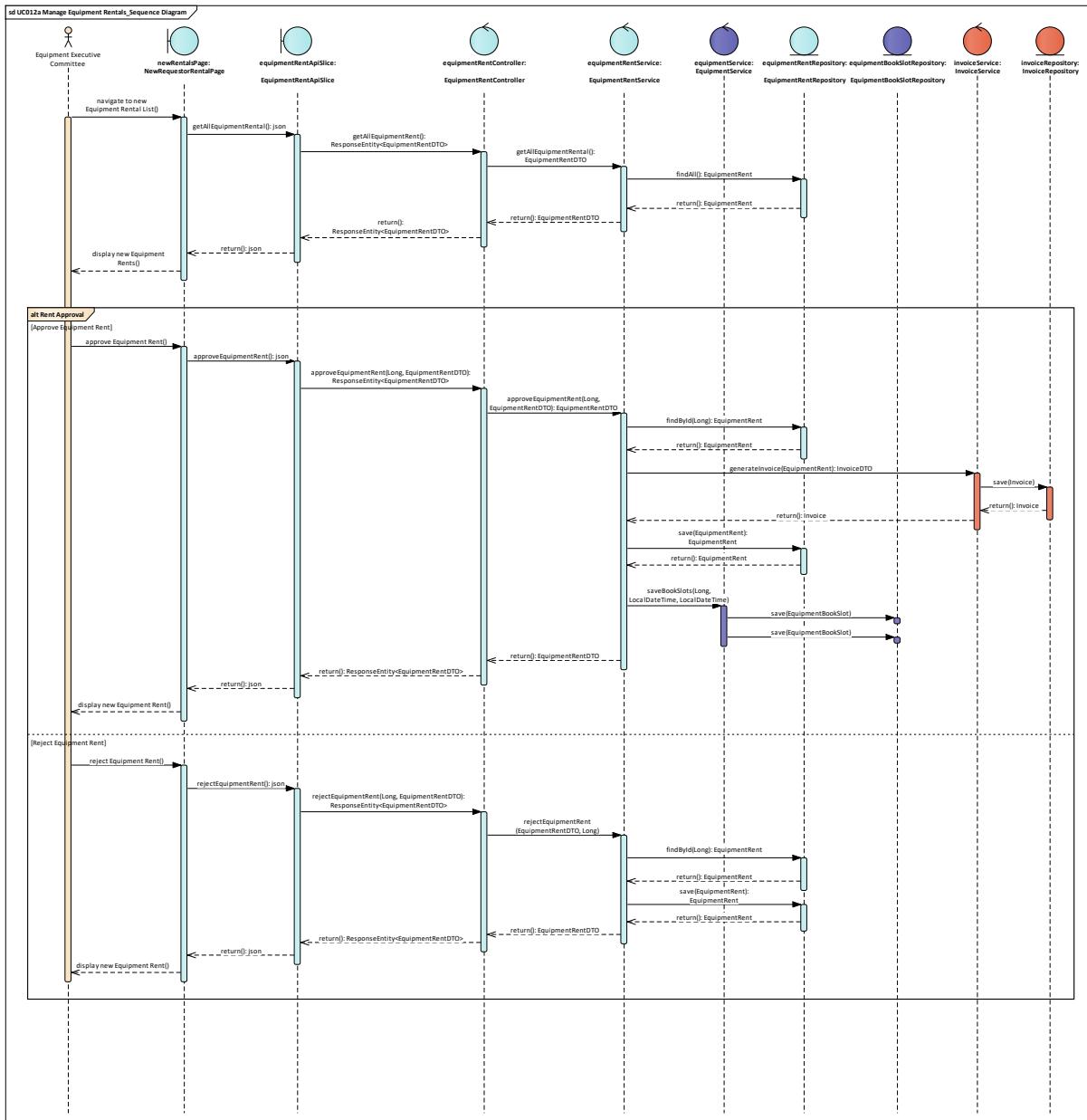


Figure 2.30 Sequence Diagram for UC012a Manage Equipment Requests and Rentals

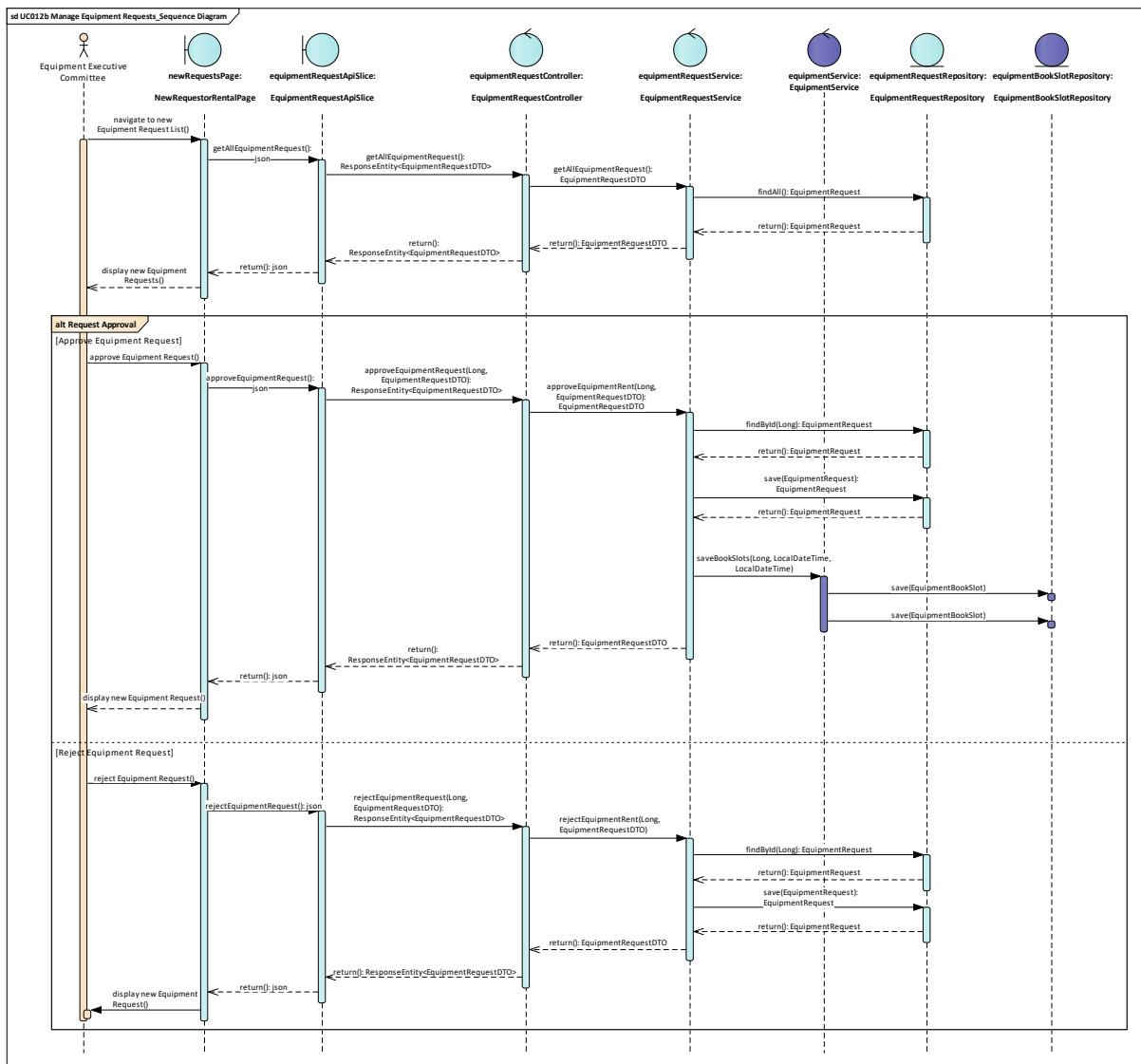


Figure 2.31 Sequence Diagram for UC012b Manage Equipment Requests and Rentals

2.3.13 UC013: Confirm Equipment Return

2.3.13.1 Use Case Description of UC013

Table 2.15 Table of Use Case Description for UC013 Confirm Equipment Return

Use Case ID	UC013
Use Case Name	Confirm Equipment Return
Description	This use case allows the Equipment Executive Committee to confirm the return of borrowed equipment after verifies its conditions and completeness.
Actor(s)	Equipment Executive Committee.
Pre-condition(s)	<ol style="list-style-type: none"> 1. The borrowed or rented equipment has been returned by the Equipment Committee Member or any User (UC007). 2. Return status is pending in the system.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. Equipment Executive Committee navigates to the "Returned Equipment" section. 2. Equipment Executive Committee selects a pending return equipment section. 3. Equipment Executive Committee verifies the equipment checklist and notes. 4. Equipment Executive Committee confirms the equipment return in the system. 5. System updates equipment status, rental status and logs the return.
Alternative Flow(s) - AF	AF1: Partial or Incomplete Return <ol style="list-style-type: none"> 1. Equipment Executive Committee marks some items as missing or damaged. 2. System notify Equipment Executive Committee to follow up with the Renter or Equipment Committee Member. 3. System records the issue and flags equipment for further review.
Exception Flow(s) - EF	EF1. No Equipment Returned Past Due Date <ol style="list-style-type: none"> 1. The system will notify the Equipment Executive Committee to follow up with the borrower or renter.

Post-condition(s)	<ol style="list-style-type: none"> Equipment status notes are updated to "Available" or "Damaged/Missing". Equipment returns has been recorded in the system.
--------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

2.3.13.2 Activity Diagram of UC013

Figure 2.32 Activity Diagram for UC013 Confirm Equipment Return

2.3.13.3 Sequence Diagram of UC013

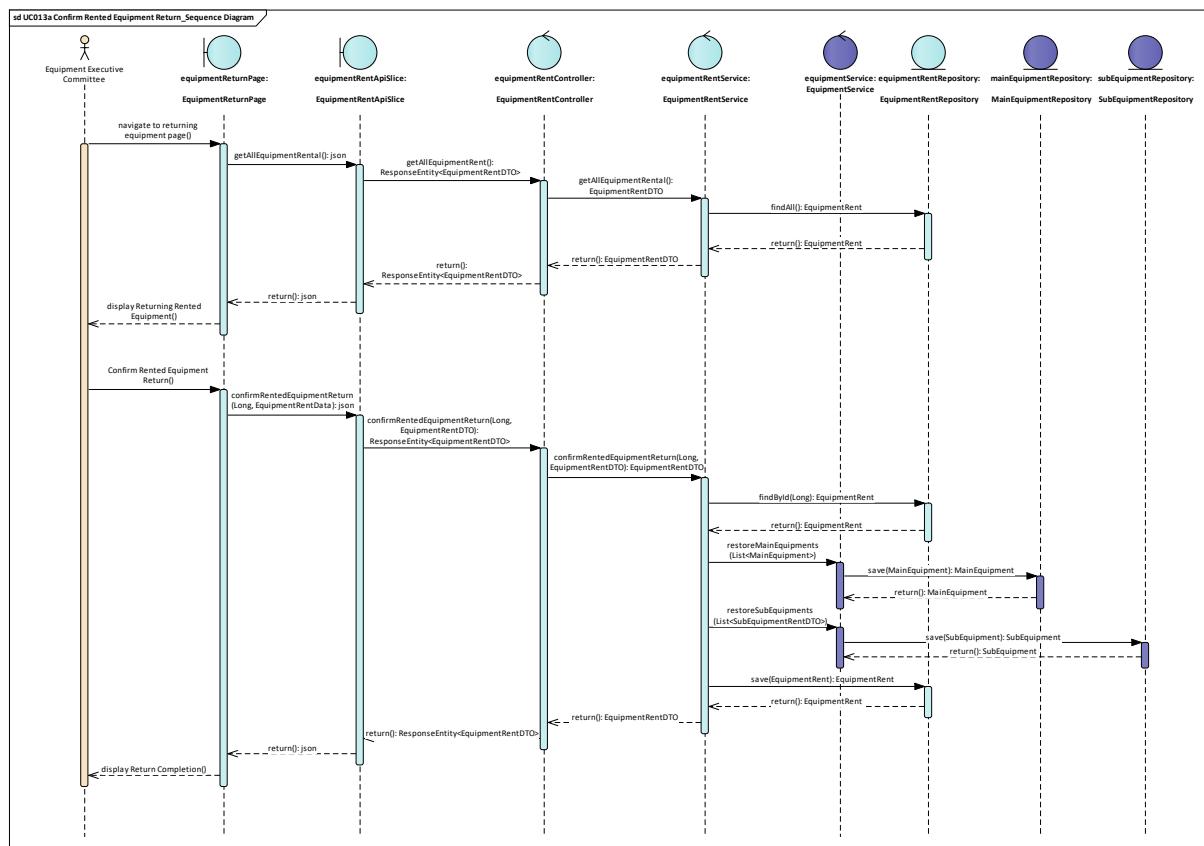


Figure 2.33 Sequence Diagram for UC013a Confirm Equipment Return

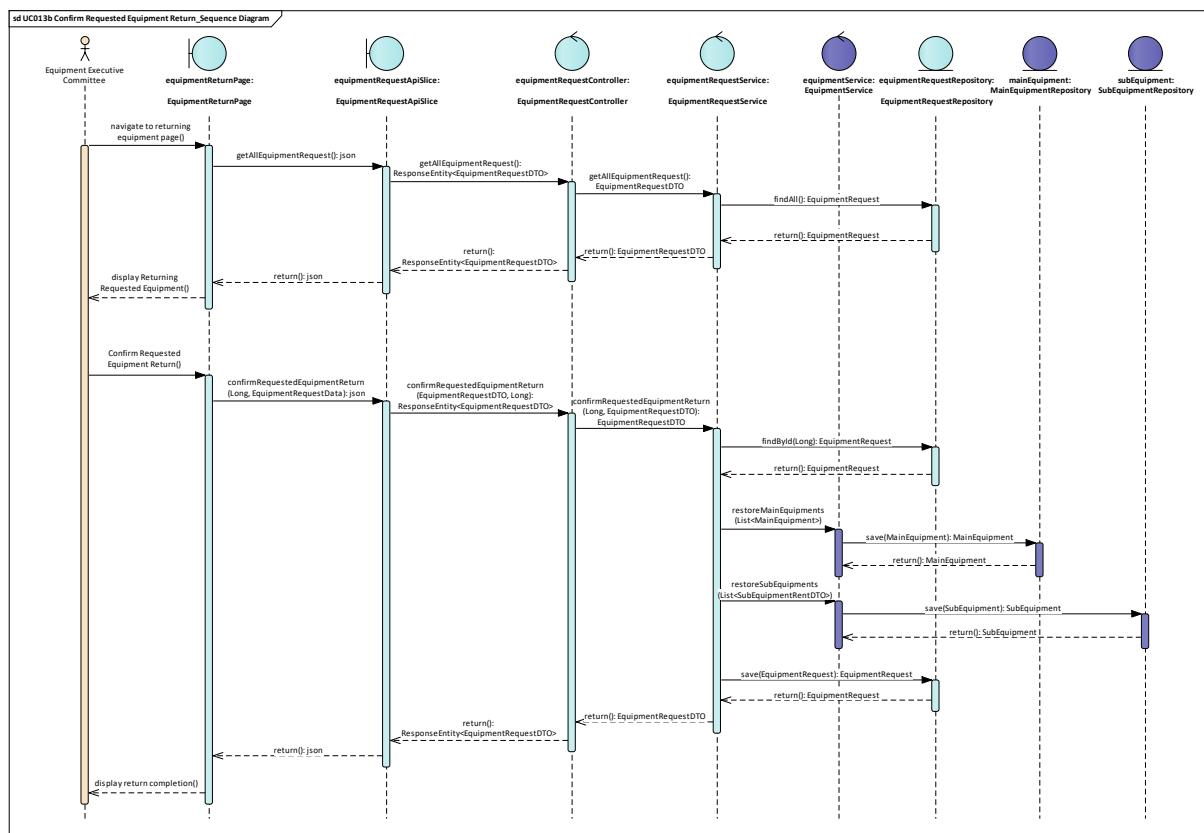


Figure 2.34 Sequence Diagram for UC013b Confirm Equipment Return

2.3.14 UC014: Manage Members Roles

2.3.14.1 Use Case Description of UC014

Table 2.16 Table of Use Case Description for UC014 Manage Members Roles

Use Case ID	UC014
Use Case Name	Manage Members Roles
Description	This use case allows Administrator to assign, change or revoke roles of the club members within the system based on their responsibilities in the club.
Actor(s)	Administrator
Pre-condition(s)	<ol style="list-style-type: none"> 1. The Administrator is authenticated and logged into the system. 2. The member to be assigned has a registered account in the system. 3. A specific event must already exist (UC017) before assigning a member to Equipment Committee Member.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. Administrator navigates to the “Manage Roles” section. 2. Administrator selects the member and chooses a new role or revokes existing roles. 3. Administrator confirms the role changes. 4. System updates the member’s role and access rights.
Alternative Flow(s) - AF	<p>AF1. Search User</p> <ol style="list-style-type: none"> 1. The Administrator enters search keywords to search the user’s name in the system. <p>AF2. Update User Role</p> <ol style="list-style-type: none"> 1. The Administrator can change the member role. 2. The system will save the new user role. <p>AF3. Revoke User Role</p> <ol style="list-style-type: none"> 1. The Administrator can revoke the user role to remove the user access rights. 2. The system saves remove the role for specified user.

	<p>AF4. Assign Equipment Committee Member to Specific Event</p> <ol style="list-style-type: none"> 1. The Administrator can assign a user as Equipment Committee Member to a specific event. 2. The Administrator choose specific event.
Exception Flow(s) - EF	<p>EF1. Failed Data Retrieval</p> <ol style="list-style-type: none"> 1. The system displays an error message, "Failed to Retrieve Data." 2. The system prompts administrator to reload page. <p>EF2. Events Does Not Exist Yet</p> <ol style="list-style-type: none"> 1. The system displays an error message, "There is no existing Event to assign the user as "Equipment Committee Member." 2. The system prompt Administrator to create an event (UC016).
Post-condition(s)	<ol style="list-style-type: none"> 1. Member role is updated in the system. 2. System permissions and access rights are given to the member according to the new role assigned.

2.3.14.2 Activity Diagram of UC014

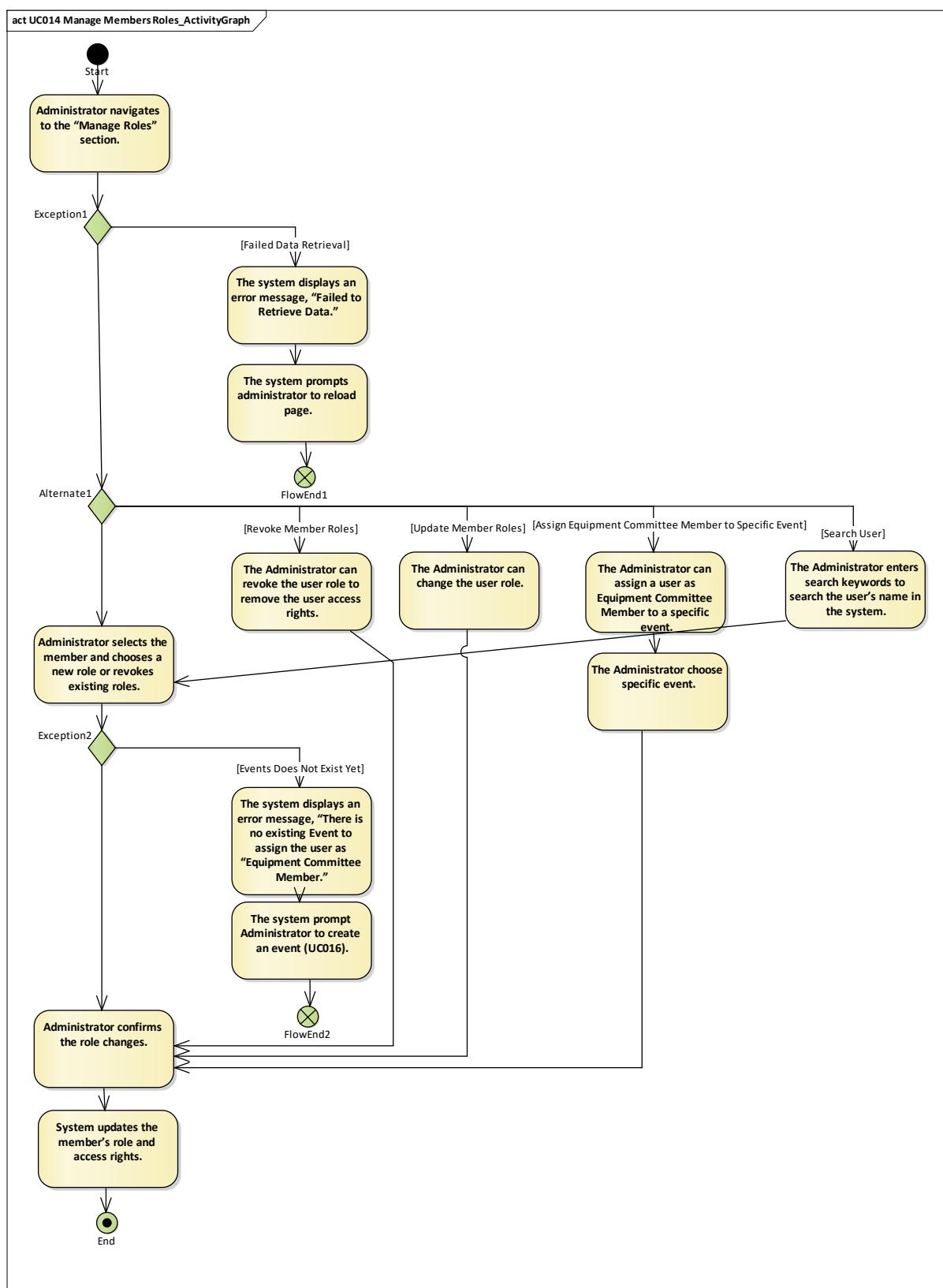


Figure 2.35 Activity Diagram for UC014 Manage Member Roles

2.3.14.3 Sequence Diagram of UC014

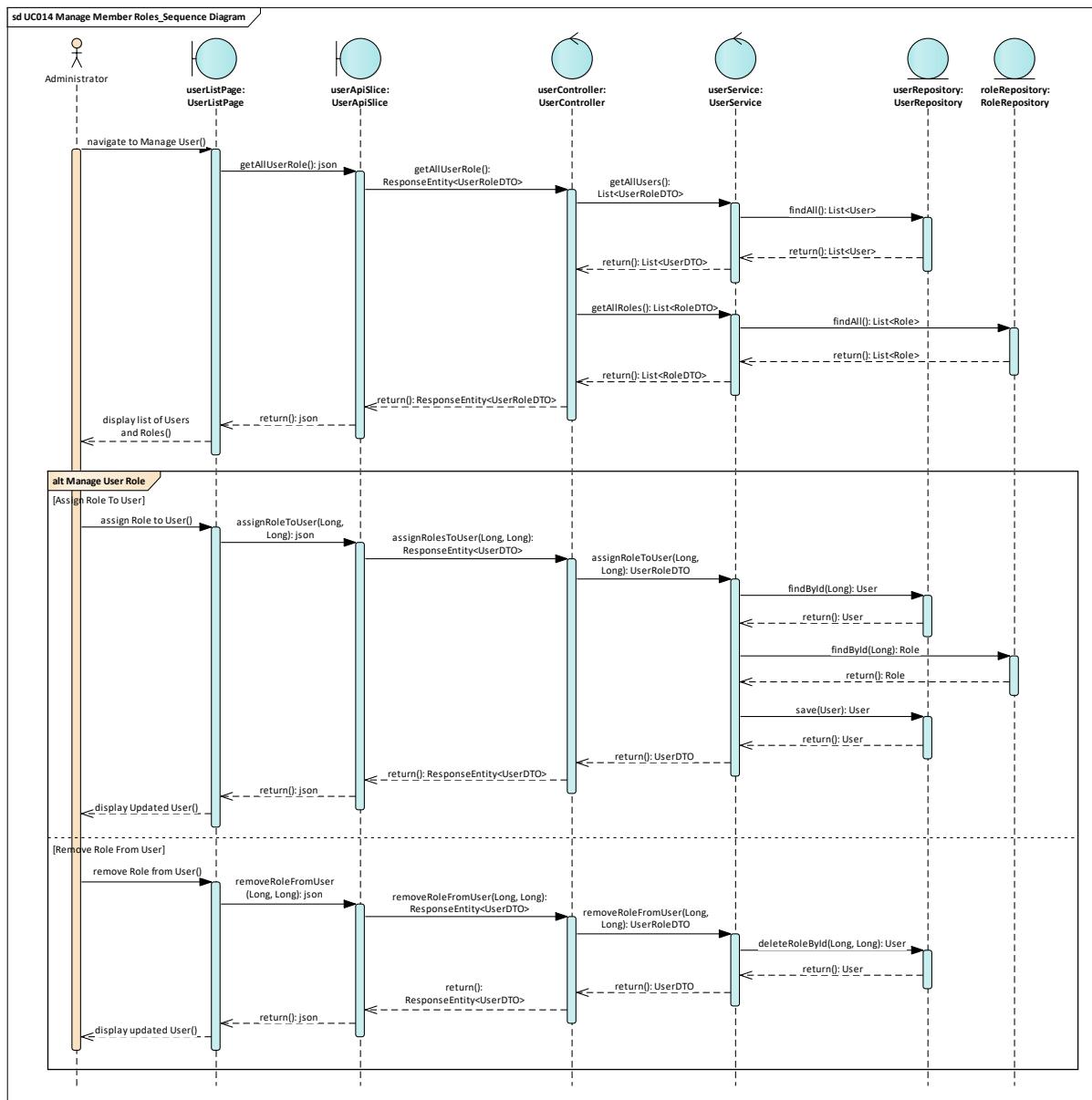


Figure 2.36 Sequence Diagram for UC014 Manage Member Roles

2.3.15 UC015: Manage Event

2.3.15.1 Use Case Description of UC015

Table 2.17 Table of Use Case Description for UC015 Manage Event

Use Case ID	UC015
Use Case Name	Manage Event
Description	This use case allows the Administrator to create, update, and manage events conducted by the club.
Actor(s)	Administrator
Pre-condition(s)	<ol style="list-style-type: none"> 1. The Administrator must be logged into the system.
Normal Flow(s)- NF	<ol style="list-style-type: none"> 1. Administrator navigates to the event management interface. 2. Administrator view the list of Events created. 3. Administrator selects "Create Event". 4. Administrator inputs event details such as name, description, dates, etc. 5. Administrator submits the form. 6. System validates and saves the event details.
Alternative Flow(s) - AF	<p>AF1: Edit Existing Event</p> <ol style="list-style-type: none"> 1. Administrator click "Edit" button on existing event. 2. Administrator edit the necessary field. <p>AF2: Delete Specific Event</p> <ol style="list-style-type: none"> 1. Administrator click "Delete" button on existing event. 2. System prompts confirmation message to administrator. 3. Administrator click "Confirm." <p>AF3: Assign Equipment Committee Member</p> <ol style="list-style-type: none"> 1. Administrator can select user from list and assign them as Equipment Committee Member or skip it.
Exception Flow(s) - EF	<p>EF1. Failed Data Retrieval</p> <ol style="list-style-type: none"> 1. The system displays an error message, "Failed to Retrieve Data." 2. The system prompts administrator to reload page.

Post-condition(s)	The event is created and stored in the system.
--------------------------	------------------------------------------------

2.3.15.2 Activity Diagram of UC015

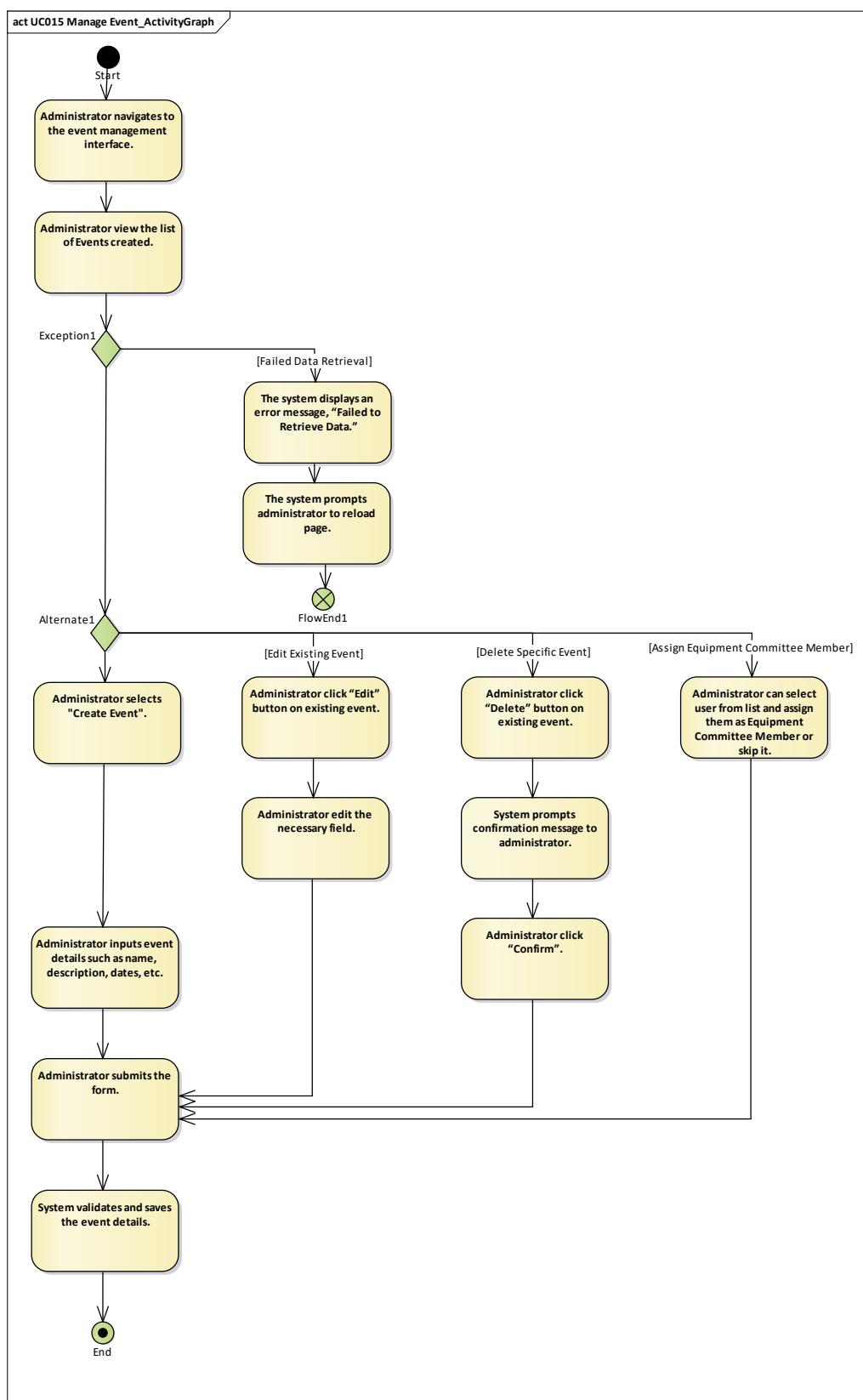


Figure 2.37 Activity Diagram for UC015 Manage Event

2.3.15.3 Sequence Diagram of UC015

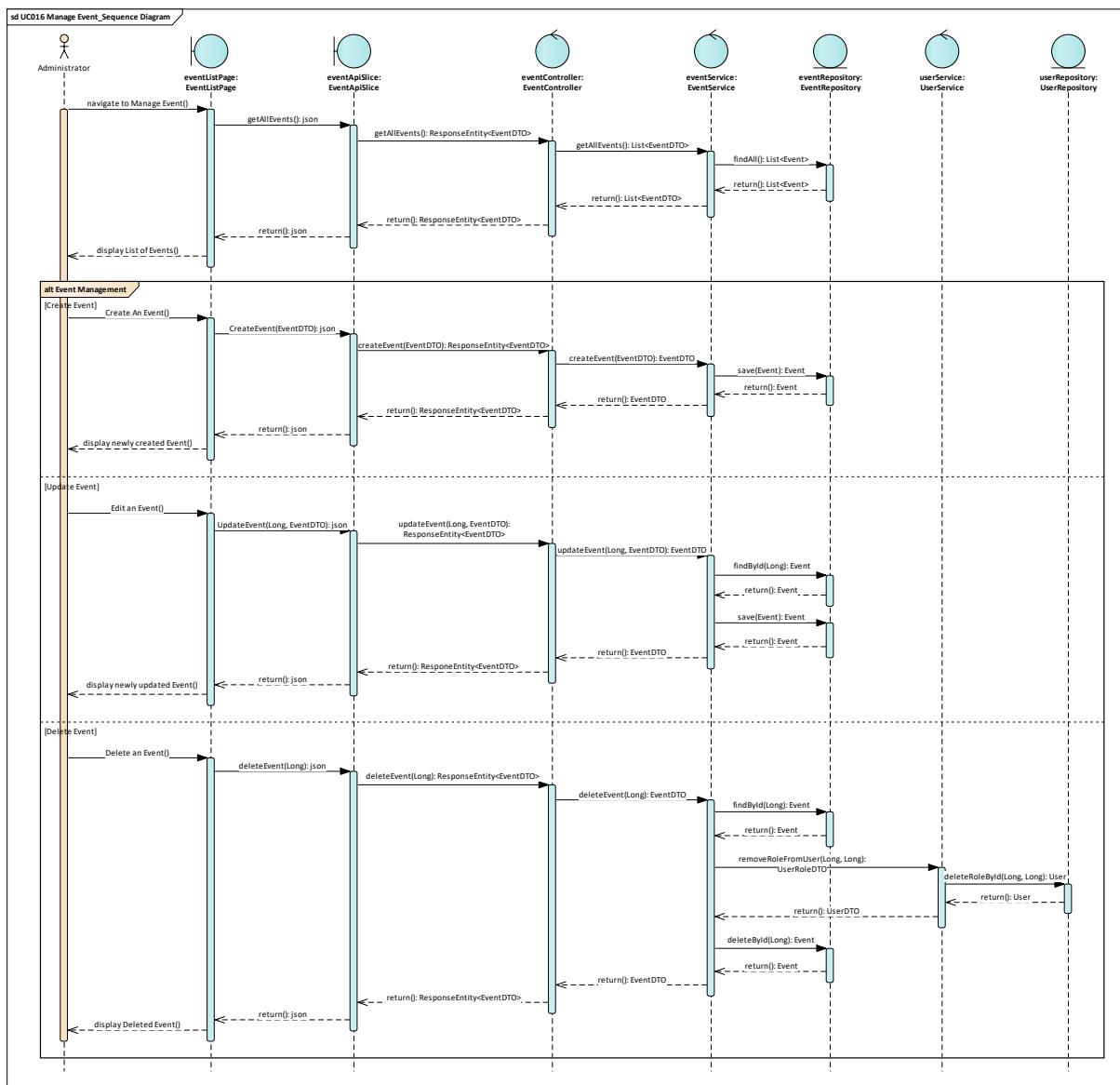


Figure 2.38 Sequence Diagram for UC015 Manage Event

2.4 Performance and Other Requirements

Performance Requirements

The requirements stated below ensure the IFoto works properly within range of expected workloads:

- **Response Time:** The system should be able to respond to user interactions within 2 seconds under normal load conditions.
- **Throughput:** The system should be able to process at least 20 concurrent user requests per minute without long delay.
- **Capacity:** The system must support up to 80 registered users and handle at least 40 concurrent active users.
- **Availability:** The system should maintain 99% uptime during operational hours (8:00 AM to 10:00 PM, daily).

Software System Requirements

The Software System Requirements stated below defines the non-functional requirements aspects of IFoto:

- **Usability:** The system interface shall be user-friendly for new users to navigate through and perform basic functions with minimal to no training.
- **Reliability:** The system must perform consistently with minimal error within long period of time.
- **Maintainability:** The system should be modular and follow the Object-Oriented Programming to allow extension or bug fixes to be implemented within 2 days for minor issues.
- **Portability:** The web application shall be accessible on major browsers such as Google Chrome, Firefox, Edge, and work on mobile device as well via responsive design.
- **Security:** All user data shall be encrypted in transit and at rest and sensitive data will be filtered out before send to frontend for user. Role-based access control (RBAC) will be implemented to authorize user based on their respective role.

2.5 Design Constraints

Access Devices

- The system must be accessible to common user devices such as PCs, laptops, smartphones, and tablets used by club members and external user by implementing responsive web design.

Browser Compatibility

- The system must function properly and compatible with common browsers such as Chrome, Firefox, Microsoft Edge, and Safari.

Integration Standards

- RESTful APIs must be used for frontend and backend communication in JSON format.

Stakeholder Demands

- IFoto functionalities are highly dependent to the stakeholder's business logic, demands and operational workflows which defines IFoto core logic such as equipment management, booking, approval, and role management.

Appendix C Software Design Document (SDD)



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECJ 3032: Software Engineering FYP1
Semester 02, 2024/2025

Software Design Document (SDD)

Inventory Management System for Kelab
Fotokreatif

Version 1.0

17/06/2025

Prepared by: Lio Kock Hock

Revision Page

1.1 Overview

This current document contains the introduction of Inventory Management System for Kelab Fotokreatif (IFoto), overall description of the system, and specific requirement that include Context Viewpoint, Composition Viewpoint, Logical Viewpoint, Information Viewpoint, Interface Viewpoint, Structure Viewpoint, Interaction Viewpoint, State Dynamic Viewpoint and Algorithm Viewpoint that were modelled using Unified Modelling Language (UML).

1.2 Target Audience

The target audience are Kelab Fotokreatif members, developers, project managers and software testers.

1.3 Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Lio Kock Hock	Completed SDD version 1.0	17/06/2025

Note:

This Software Design Descriptions (SDD template is adapted from IEEE Recommended Practice based on Software Design Descriptions (SDD) (IEEE Std. 1016-1998 1), that are simplified and customized to meet the need of SECJ3203 FYP1 SE course at Faculty of Computing, UTM. Examples of models are from Arlow and Neustadt (2002) and other sources stated accordingly.

Table of Contents

1	Introduction	152
1.1	Purpose.....	152
1.2	Scope.....	152
1.3	Overview	153
2	Reference	154
3	Glossary	155
4	Design Body.....	156
4.1	Design Stakeholders and Their Concerns	156
4.2	Context Viewpoint	158
4.2.1	Design Concerns	158
4.2.2	Design View (Use Case Diagram)	159
4.3	Composition Viewpoint	161
4.3.1	Design Concerns	161
4.3.2	Design View (Architectural Style and & Pattern Diagram)	162
4.4	Logical Viewpoint	164
4.4.1	Design Concerns	164
4.4.2	Design View (Class Diagram)	170
4.5	Information Viewpoint	176
4.5.1	Design Concerns	176
4.5.2	Design View (ERD Diagram)	178
4.6	Interface Viewpoint	183
4.6.1	Design Concerns	183
4.6.2	Design View (Interfaces).....	185
4.7	Structure Viewpoint.....	201
4.7.1	Design Concerns	201
4.7.2	Design View (Package Diagram)	202
4.8	Interaction Viewpoint.....	208
4.8.1	Design Concerns	208
4.8.2	Design View (Sequence Diagram).....	209
4.9	State Dynamic Viewpoint	225
4.9.1	Design Concerns	225
4.9.2	Design View (State Machine Diagram)	226
4.10	Algorithm Viewpoint	229
4.10.1	Design Concerns	229
4.10.2	Design View (Activity Diagram).....	230

5	User Interface Design	245
5.1	User Interface for Equipment Request and Rentals Module	245
5.2	User Interface for Equipment Management Module.....	253
5.3	User Interface for Role and Event Management Module	257

1 Introduction

1.1 Purpose

The Inventory Management System for Kelab Fotokreatif (IFoto), a centralised online platform for inventory management, is designed and documented in this Software Design Document (SDD). The purpose of this SDD is to give all the information required for this web-based system's database design, user interface design, detailed design, and system architecture. Most of the criteria and information in this publication are taken from the Software Requirement Specification (SRS), which it is a continuation of. SDD primarily focusses on the design's planning, analysis, implementation, and decision-making processes. This helps developers better understand the structure and behaviour of the system, which facilitates the conversion to code.

1.2 Scope

The extent of this Software Design Document (SDD) is to provide the software architecture and design requirements of the Inventory Management System of Kelab Fotokreatif (IFoto). The internal structure and design considerations that govern the development of the system including how it fulfils the requirements as stipulated in the Software Requirements Specification (SRS) are outlined in this document. The SDD is a guide to software developers, testers and system integrators who work on the development and support of IFoto.

The IFoto system will assist and automate the process of renting and returning of equipment, assigning user roles, scheduling events and payment. It is organized into four major modules: Equipment Management, Equipment Request and Rental, Role and Event Management and Authentication and Authorization. The Model-View-Controller (MVC) is an architectural pattern that ensures modules separation of concerns, enhanced maintainability, and module development (Gamma et al., 1994).

The SDD expounds on the different software design perspectives including the Logical, Composition, Interface, Algorithm and Context perspectives as suggested in IEEE 42010:2011 standards. Some of the technologies used in the implementation are ReactJS on the frontend, Spring Boot on the backend, and MySQL as the relational database layer. The system uses RESTful APIs to communicate between client and server parts and incorporates DTOs to transport data efficiently and abstract (Fowler, 2002).

This document deals with the technical part of the system only and omits the non-technical issues related to the project scheduling, resources allocation or budgeting. It is mainly used as a technical guide to make sure that the system design is achieved with the functional requirements and non-functional quality characteristics like scalability, usability, and performance (Pressman & Maxim, 2020).

1.3 Overview

This Software Design Document (SDD) gives a detailed account of the architecture and design decisions to be followed to develop the Inventory Management System of Kelab Fotokreatif (IFoto). It is a blueprint that helps the developers, testers, and stakeholders to get a picture of how the system is organized, how the components work with each other and how the system satisfies the functional and non-functional requirements stipulated in Software Requirements Specification (SRS).

The SDD is structured into several viewpoints such as Context Viewpoint, Composition Viewpoint, Logical Viewpoint, Interface Viewpoint, and Algorithm Viewpoint. Both perspectives support various design issues and introduce models like class diagrams, sequence diagrams, activity diagrams, and architectural diagrams to demonstrate behaviour and structure of the system.

2 Reference

Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.

IEEE. (2011). *ISO/IEC/IEEE 42010:2011 systems and software engineering—Architecture description*. Institute of Electrical and Electronics Engineers.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

3 Glossary

Definitions of all terms, acronyms and abbreviation used are:

SDD – System Design Document

UML – Unified Modelling Language

RBAC – Role-Based Access Control

BIGINT – a variable to store big Integer

VARCHAR – a variable to store String

BOOLEAN – a variable to store the value of True or False

ERD – Entity Relationship Diagram

4 Design Body

4.1 Design Stakeholders and Their Concerns

This system is designed with a number key stakeholders whose role is different in making sure that this system achieves both functional and operational demands. They affect the architectural and design process of development directly with their concerns.

System Developers

Implementation of the system is the responsibility of the Developers following design. Their main issues are a clear interaction of components, presence of reusable services and good documentation of algorithms, interfaces, and data structures.

Integrators and Testers

Testers ensure that the design enables successful unit testing and integration and concentrate on checking that there is consistency among modules. They are worried about the fact whether interfaces between modules are clearly defined and that error-handling mechanisms are traceable.

Administrator

The administrators are charged with the task of administrating the total system including assigning duties, making events as well as tracking the use of equipment. Security of the system, willingness of the system to permit role management, ability to audit what people are doing and that all the modules are glued together is the major issues among them.

Equipment Executive Committee

They check and accept request and rental of equipment. These are their fears to have a clear interface to check pending approvals, a correct rental history, ability to verify returns, and a possibility to produce invoices and collect payments.

Equipment Committee Member

They get events assigned to them by Administrator and they can submit requests of equipment they want to use and make sure equipment are returned accordingly. They are also concerned with the ease of equipment request creation as well as where their requests are and equipment availability pertaining to the events assigned to them.

User (Club Member or Renter)

General users operate the system to make an equipment rental or to see their past rentals. They are also concerned with an easy-to-use interface, real-time information about the availability of equipment, clear payment and creation of the invoice system, and the ability to track their status of being rented and time limits of returning the equipment.

4.2 Context Viewpoint

4.2.1 Design Concerns

The purpose of the Context viewpoint is to identify a design subject's offered services, its actors (User, Equipment Committee Member, Equipment Executive Committee and Administrator), to establish the system boundary and to effectively delineate the design subject's scope of use and operation.

The system features described in the Context Viewpoint involves key services and functionalities of the system to its users and other actors. These are the main functionalities of the system to are supposed to deliver which was derived from stakeholder requirements and demands. For example, in an Inventory Management System, features may include user registration and authentication, equipment booking, return processing, payment processing, and equipment management. These features are presented as externally visible behaviours and interactive to user which emphasize what the system does for its actors rather than how it is internally implemented.

4.2.2 Design View (Use Case Diagram)

Figure 4.1 represent the system features and its respective actor that perform the functionalities. Table 4.1 shows each use case and its description.

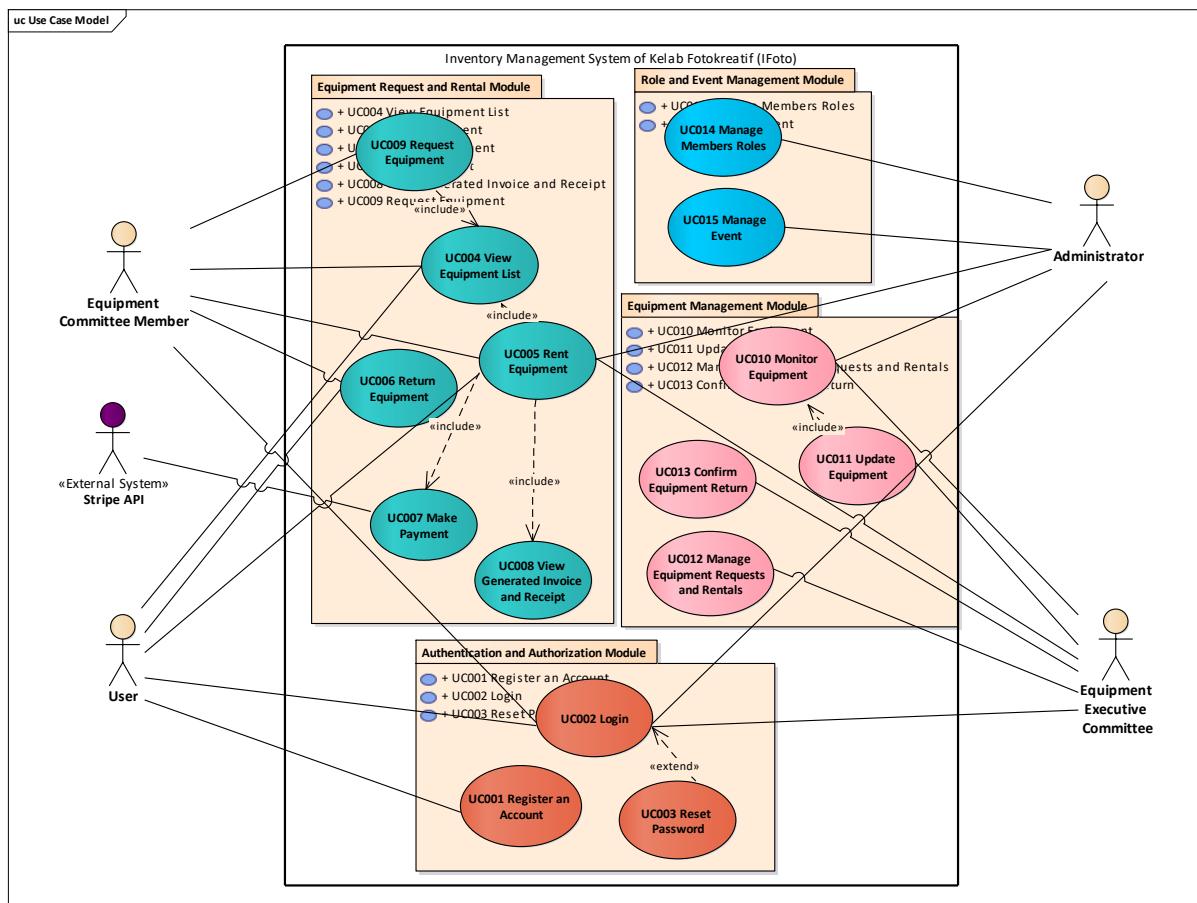


Figure 4.1 Use Case Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)

Table 4.1 Table of Description for each Use Case

Use case	Function	Description
UC001	Register an Account	This use case allows User to sign up as a normal user by providing personal information and credentials.
UC002	Login	This use case enables registered User to log in to the app using their username and password.
UC003	Reset Password	This use case allows User who forgot their password to reset it via email verification.

UC004	View Equipment List	This use case displays a list of all available equipment details such as name, quantity, status, and condition, allowing User and Equipment Committee Member to view it.
UC005	Rent Equipment	This use case allows any user to fill in the Club's Equipment Rental Form digitally to request for equipment formally.
UC006	Return Equipment	This use case facilitates the return process of rented by User and borrowed equipment by the Equipment Committee Member and update the inventory.
UC007	Make Payment	This use case allows User to make payments for equipment rentals through integrated payment gateways.
UC008	View Generated Invoice and Receipt	This use case allows system view the invoice generated after equipment Rent has been approved and view the receipt after payment is made.
UC009	Request Equipment	This use case allows Equipment Committee Member to request equipment in advance for planned events or activities.
UC010	Monitor Equipment	This use case enables the Equipment Executive Committee and Administrator to track equipment availability, status, and conditions.
UC011	Update Equipment	This use case allows Executive Equipment Committee to add, edit, or delete equipment details in the system.
UC012	Manage Equipment Requests and Rentals	This use case enables the Equipment Executive Committee to approve, reject, or manage ongoing rentals and requests.
UC013	Confirm Equipment Requests	This use case allows the Equipment Executive Committee to confirm the return of borrowed equipment after verifies its conditions and completeness.
UC014	Manage Members Roles	This use case allows Administrator to assign or revoke roles of the club members within the system.
UC015	Manage Event	This use case allows Administrator to create, update, and manage events that involve assignment of roles.

4.3 Composition Viewpoint

4.3.1 Design Concerns

Composition Viewpoint is meant to clarify the way in which IFoto is composed of smaller modules or subsystems, as well as clarify respective tasks. This practice of separation of concerns gives structural perspective of the system and assist in making system more modular, maintainable, and scalable.

In the case of IFoto, the system consists of some important modules, which are Authentication and Authorization Module, Equipment Management Module, Equipment Request and Rental Module and Role and Event Management Module. All these modules are developed as subsystems that entrap certain services, data models, and controllers. By way of example, the Equipment Management Module is used to track inventory and makes updates and the Authentication Module manages user access and security roles. Such a failure encourages the modularity, enabling changes or improvements to a section of the system without affecting the rest of it significantly.

In addition, clear structure of the composition leads to scalability. With the system design of the IFoto system evolving, features/modules may be added as soon as the system evolves rather than redesigning the whole system. It additionally increases its maintainability through the individualisation of faults or bugs in certain modules and thus easier to debug and test. All the modules can be tested separately and then incorporated in the entire system. The Composition Viewpoint, thus, not only exhibits the internal structure of the system, yet also enables the long-term qualitative properties such as maintainability, scalability, and adaptability.

This level of modularity of the system allows the Composition Viewpoint to act as a cross between the conceptual architecture of the system and the detailed technical implementation, providing a clear and flexible boundary between the present and future development effort.

4.3.2 Design View (Architectural Style and & Pattern Diagram)

4.3.2.1 Architectural Style & Pattern Diagram

The architecture pattern and style adopted to the Inventory Management System of the Kelab Fotokreatif (IFoto) is Model-View-Controller (MVC) based on 3-tier architecture. This architecture is meant to divide the system into three levels. According to Figure 4.2, the View that is represented by the User Interface layer (React frontend), this layer oversees the user interface rendering and processing user interactions. It sends and receives data to and from the backend using API calls. Then, the Controller depicted by Business Logic layer consists of such components as Controllers, DTOs, Services, Repositories, and Models. This layer accepts the requests of the user via the REST endpoints, enforces the business rules, and formats the response in terms of DTOs to transfer the data. Lastly, the Model that is denoted by the Data Access Layer (MySQL database) stores and retrieve data. JPA Repository manages database and interacts with the Business Logic Layer via the JPA Entities to offer data persistence. These layers offer clear separation of concerns, leading to maintainability, scalability, and flexibility. Package Diagram in Figure 4.3 shows a much more details for each component in the 3-Tier Architecture where Package Frontend consist of User Interface Layer while Package Backend consist of Domain Layer and Data Access Layer.

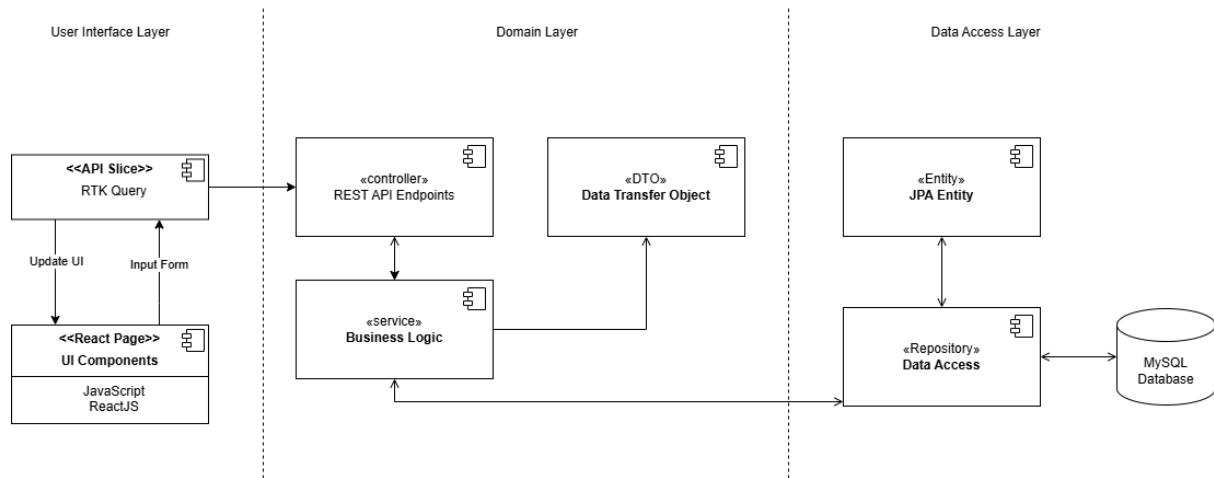


Figure 4.2 Architecture Model of 3-Tier Architecture in Inventory Management System for Kelab Fotokreatif (IFoto)

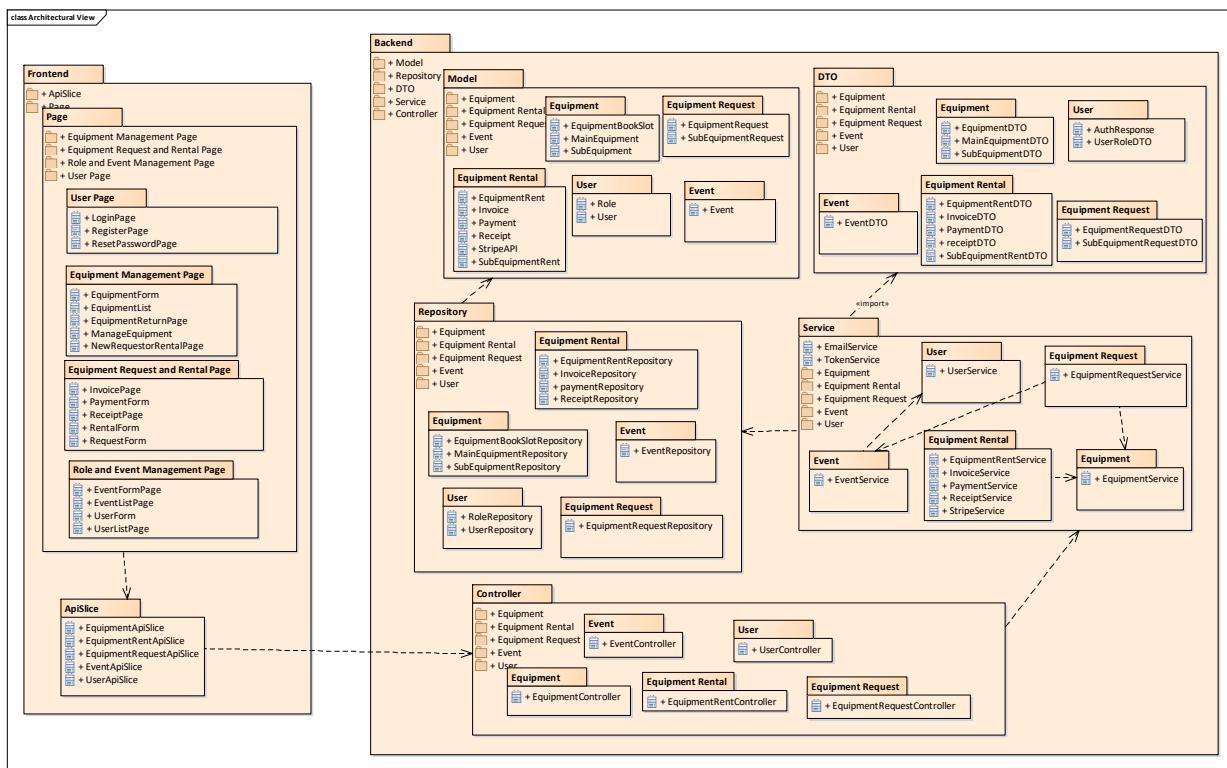


Figure 4.3 Package Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)

4.4 Logical Viewpoint

4.4.1 Design Concerns

The Logical Viewpoint states more about the corpus of the system in the sense of the installation of its construction with classes and interfaces, and their connection. It establishes how the types are realized as well as how they interact with each other to achieve the functionality of the system. This perspective is concerned with the blueprint level details and does not worry about physical deployment of the service or the wiring aspects.

Table 4.2 Table of Description for each Class

Stereotype	Class Name	Description
<<model>>	Event	Core business entity representing events with properties like eventId, eventName, description, dates, location, and equipment committees
	User	Core business entity representing system users with userId, name, email, password, phone, matriculation number, and roles
	Role	Core business entity representing user roles with roleId, roleName, and associated users
	Equipment	Core business entity representing equipment items with rental, renter, venue, signature, program name, dates, and status information
	MainEquipment	Core business entity representing main equipment with equipmentId, type, brand, model, serial number, condition, and status
	SubEquipment	Core business entity representing sub-equipment with subEquipmentId, equipmentType, brand, model, and capacity
	EquipmentRequest	Core business entity representing equipment requests with requestId, requester, event, reason, dates, status, and sub-equipment requests
	SubEquipmentRequest	Core business entity representing sub-equipment requests with subRequestId, equipment request, sub-equipment, and quantity requested
<<DTO>>	EventDTO	Data Transfer Object for Event entity with eventId, eventName, startDate, endDate, venue, and description
	UserRoleDTO	Data Transfer Object for User roles with userId, fullName, email, and roles information
	AuthResponse	Data Transfer Object for user login with name, email, password, and roles

	EquipmentRentDTO	Data Transfer Object for equipment rental with rentId, userId, program name, venue, dates, and equipment details
	SubEquipmentRent DTO	Data Transfer Object for sub-equipment rental with rental details and equipment information
	EquipmentRequest DTO	Data Transfer Object for equipment requests with requestId, userId, eventId, reason, dates, and status
	SubEquipmentRequestsDTO	Data Transfer Object containing list of sub-equipment requests
	SubEquipmentDTO	Data Transfer Object for sub-equipment with detailed specifications and availability information
	EquipmentDTO	Data Transfer Object for equipment with sub-equipment DTOs and main equipment DTOs
	MainEquipmentDTO	Data Transfer Object for main equipment with equipmentId, type, brand, model, serial number, condition, and status
<<Repository>>	EventRepository	Data access interface for Event entities defining CRUD operations
<<Repository>>	UserRepository	Data access interface for User entities defining CRUD operations
<<Repository>>	RoleRepository	Data access interface for Role entities defining CRUD operations
<<Repository>>	EquipmentRentRepository	Data access interface for Equipment rental operations
<<Repository>>	EquipmentRequestRepository	Data access interface for Equipment requests operations
<<Repository>>	EquipmentRepository	Data access interface for Equipment entities with findById, save, and other CRUD operations
<<service>>	EventService	Business logic service for event operations using EventRepository
<<service>>	UserService	Business logic service for user operations with UserRepository, RoleRepository, and PasswordEncoder
<<service>>	EquipmentRentService	Business logic service for equipment rental operations using EquipmentRentRepository
<<service>>	EquipmentRequestService	Business logic service for equipment requests operations using EquipmentRequestRepository
<<service>>	EquipmentService	Business logic service for equipment operations with MainEquipmentRepository and SubEquipmentRepository
<<service>>	PaymentService	Business logic service for Payment Operations with PaymentRepository
<<service>>	InvoiceService	Business logic service for generating Invoice Operations with InvoiceRepository

	ReceiptService	Business logic service for generating Receipt Operations with ReceiptRepository
	StripeService	Business logic service Payment Operations with PaymentRepository
<<controller>>	EventController	REST API controller handling event-related HTTP requests using EventService
	UserController	REST API controller handling user-related HTTP requests using UserService
	EquipmentRentController	REST API controller handling equipment rental requests using EquipmentRentService
	EquipmentRequestController	REST API controller handling equipment requests operations using EquipmentRequestService
	EquipmentController	REST API controller handling equipment operations with MainEquipmentService and updateEquipment methods

Relationships between Entities:

Table 4.3 Table of Relationships Between Entities

Relationship Type	Class A	Class B	Description
@ManyToMany	User	Role	A user can have multiple roles; a role can be assigned to multiple users.
@ManyToOne	EquipmentRent	User	Each rental is requested by a single user (renter).
@ManyToMany	EquipmentRent	MainEquipment	A rental can include many main equipment items, and equipment can be reused.
@OneToMany	EquipmentRent	SubEquipmentRent	A rental can contain many sub-equipment rentals.
@ManyToOne	SubEquipmentRent	SubEquipment	Each sub-equipment rent refers to one specific sub-equipment.

@OneToMany	MainEquipment	EquipmentBookSlot	Each main equipment can have many booking slots.
@OneToMany	SubEquipment	EquipmentBookSlot	Each sub-equipment can also have many booking slots.
@OneToOne	EquipmentRent	Invoice	Each rental will generate exactly one invoice.
@OneToOne	Invoice	Payment	An invoice will lead to one payment.
@OneToOne	Payment	Receipt	A receipt is issued for each payment.
@ManyToOne	EquipmentRequest	Event	Each equipment request is linked to one event.
@ManyToOne	Event	User (optional)	Some users may organize or be assigned to events.

Controller-Service Relationships:

Table 4.4 Table of Controller-Service Relationships

Controller	Service	UML Relationship Type	Explanation
EquipmentRentController	EquipmentRentService	Association	Uses the service to create, approve, return rentals. Controller doesn't own it.
UserController	UserService	Association	Delegates login/register/role operations to the service.
InvoiceController	InvoiceService	Association	Triggers invoice generation after approval.
PaymentController	PaymentService	Association	Handles Stripe payment operations.
ReceiptController	ReceiptService	Association	Triggers receipt generation post-payment.
EquipmentController	EquipmentService	Association	Performs equipment CRUD, availability checks.
EventController	EventService	Association	Retrieves and manages event data.
EquipmentRequestController	EquipmentRequestService	Association	For committee-based equipment requesting.

Service-Service Relationships:

Table 4.5 Table of Service-Service Relationships

Service A	Service B	UML Type	Explanation
EquipmentRentService	EquipmentService	Association	To fetch and update quantities, manage book slots during rental.
StripeService	InvoiceService	Association	To retrieve invoice before payment.
ReceiptService	PaymentService	Association	To retrieve payment info for receipt generation.
EquipmentRequestService	EventService	Association	To fetch event data and assign equipment request to certain event
EquipmentRequestService	EquipmentService	Association	To fetch and update quantities, manage book slots during rental.
EventService	UserService	Association	To fetch user data and assign them to certain event.

4.4.2 Design View (Class Diagram)

4.4.2.1 Complete Class Diagram

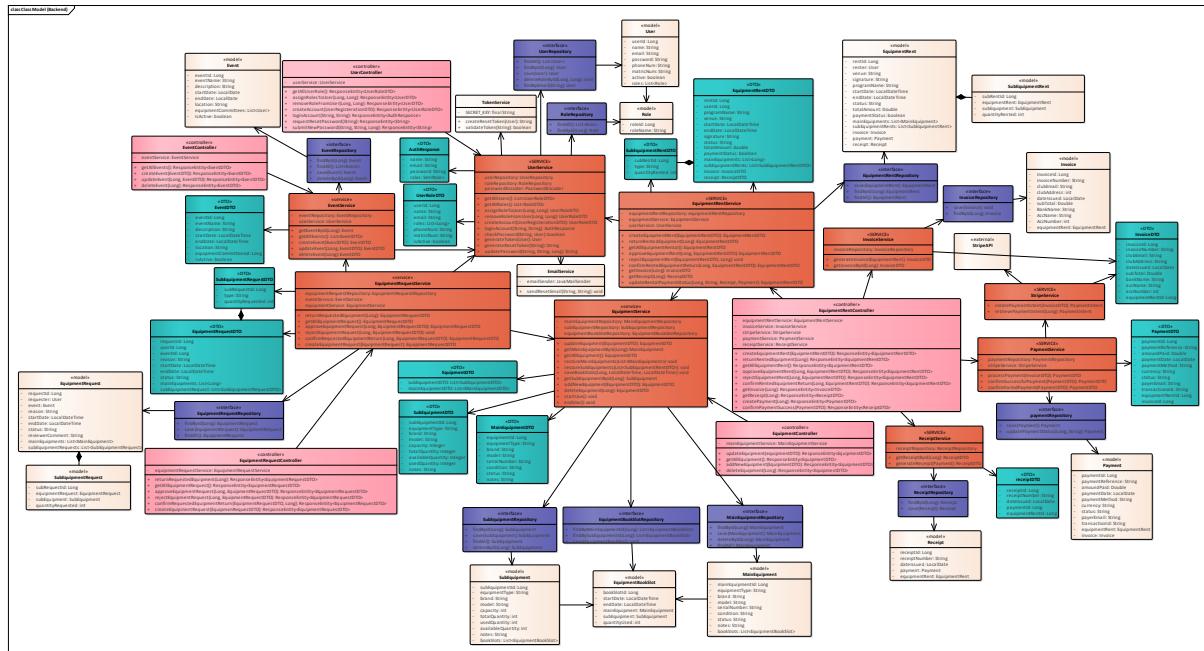


Figure 4.4 Complete Class Diagram of Backend of Inventory Management System for Kelab Fotokreatif (IFoto)

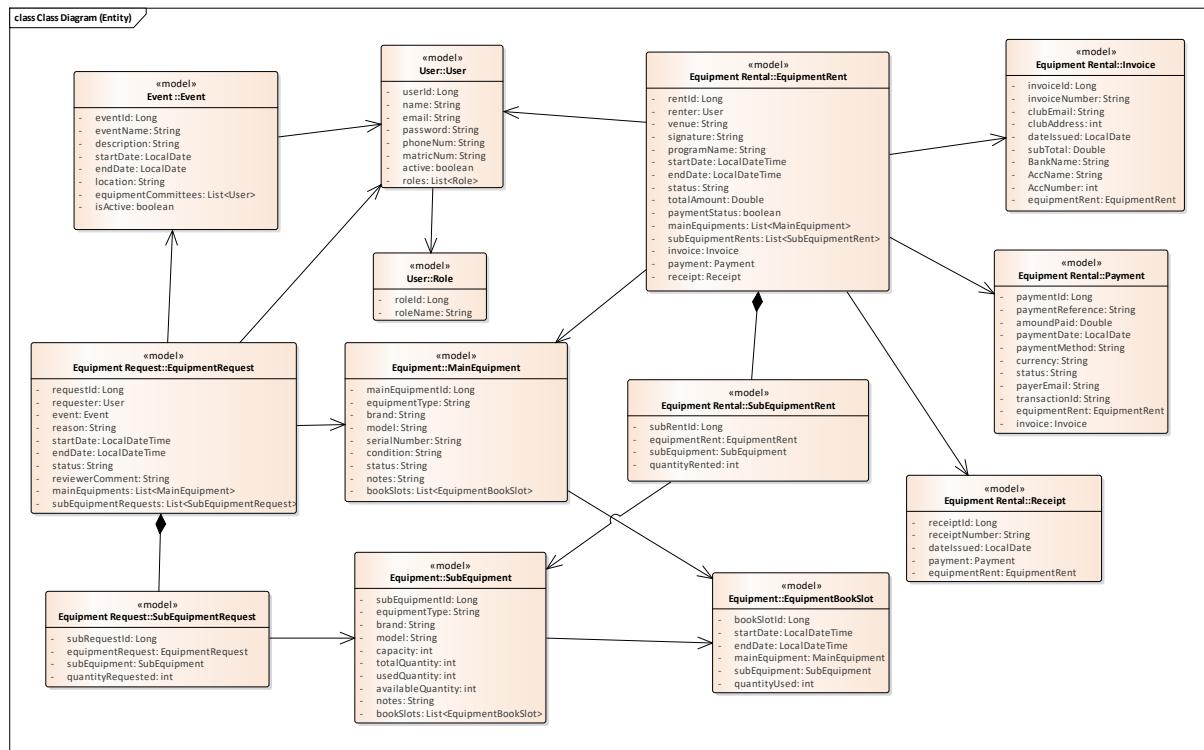


Figure 4.5 Class Diagram of Models and Relationship of Inventory Management System for Kelab Fotokreatif (IFoto)

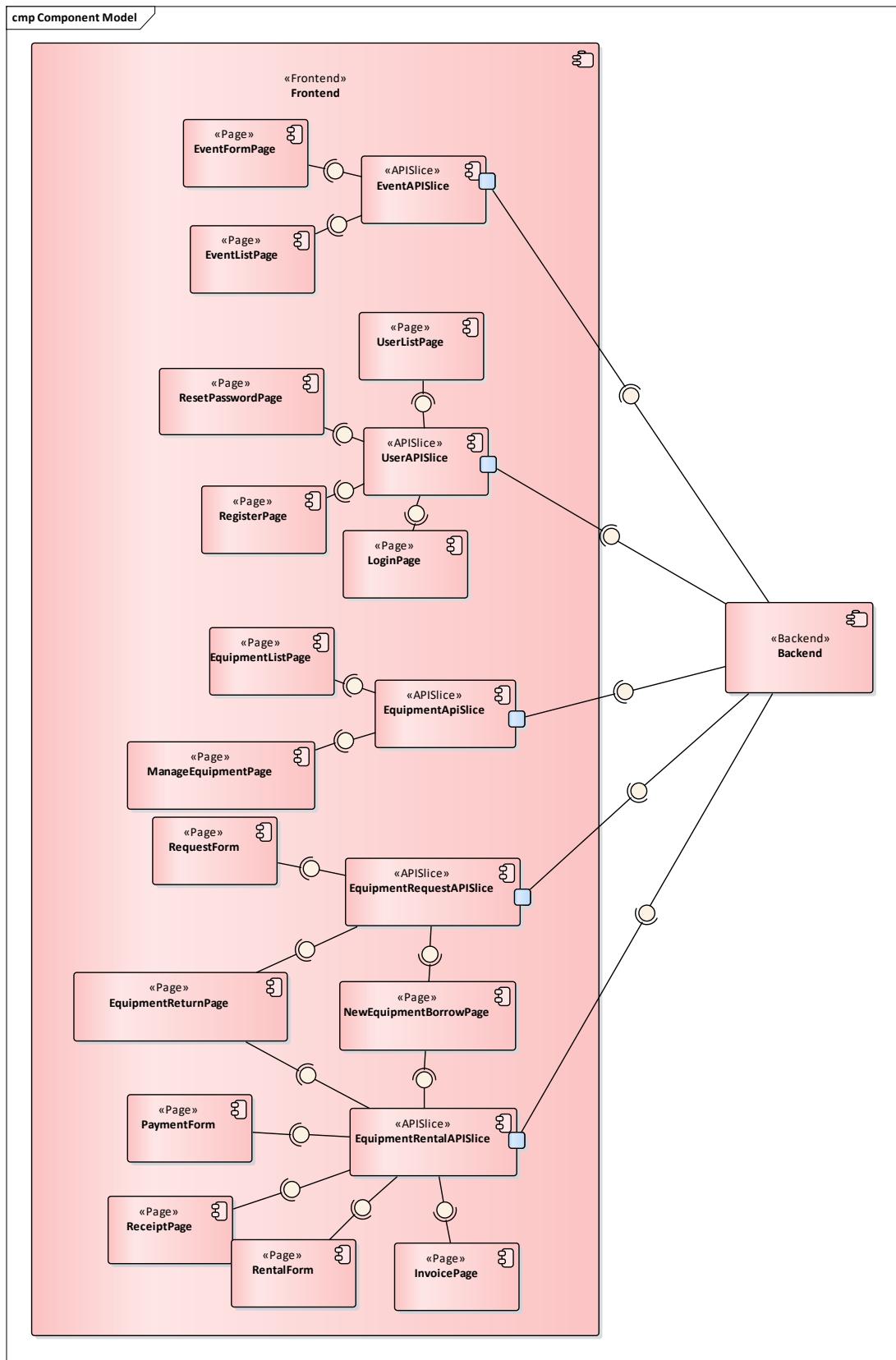


Figure 4.6 Frontend Component Diagram for Inventory Management System for Kelab Fotokreatif (IFoto)

4.4.2.2 Authentication and Authorization Module Class Diagram

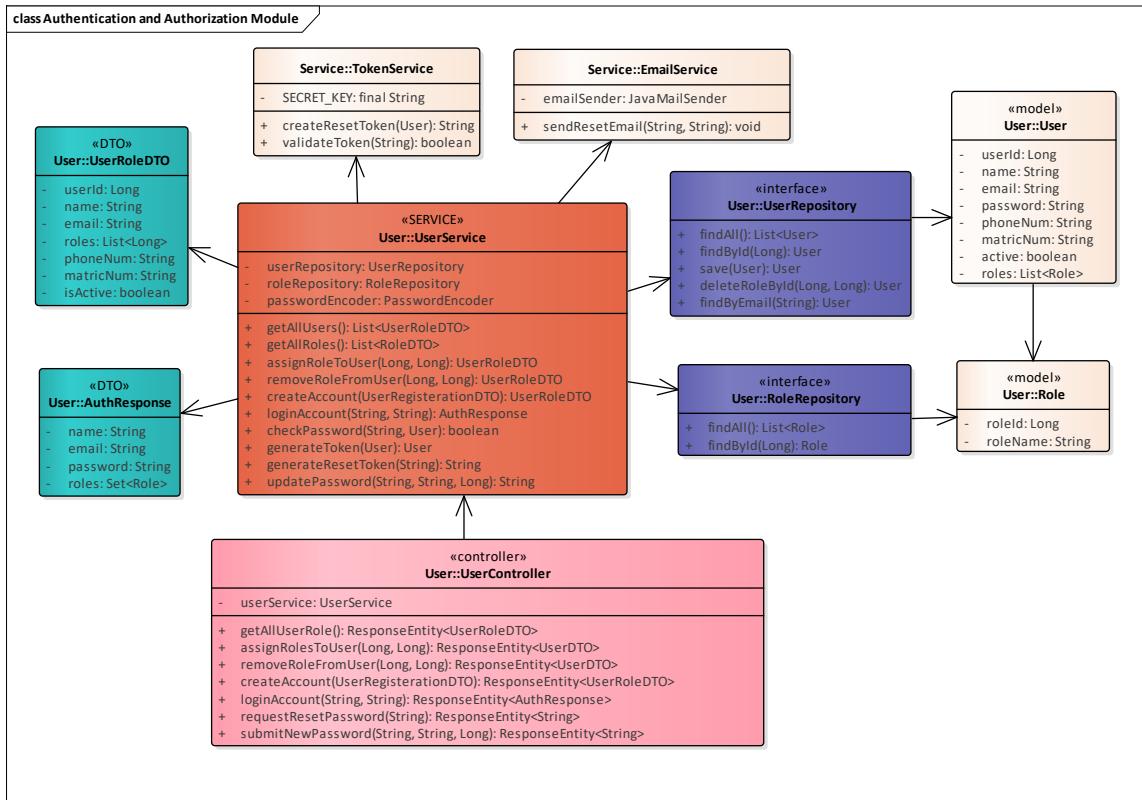


Figure 4.7 Class Diagram for Authentication and Authorization Module.

4.4.2.3 Equipment Request and Rental Module Class Diagram

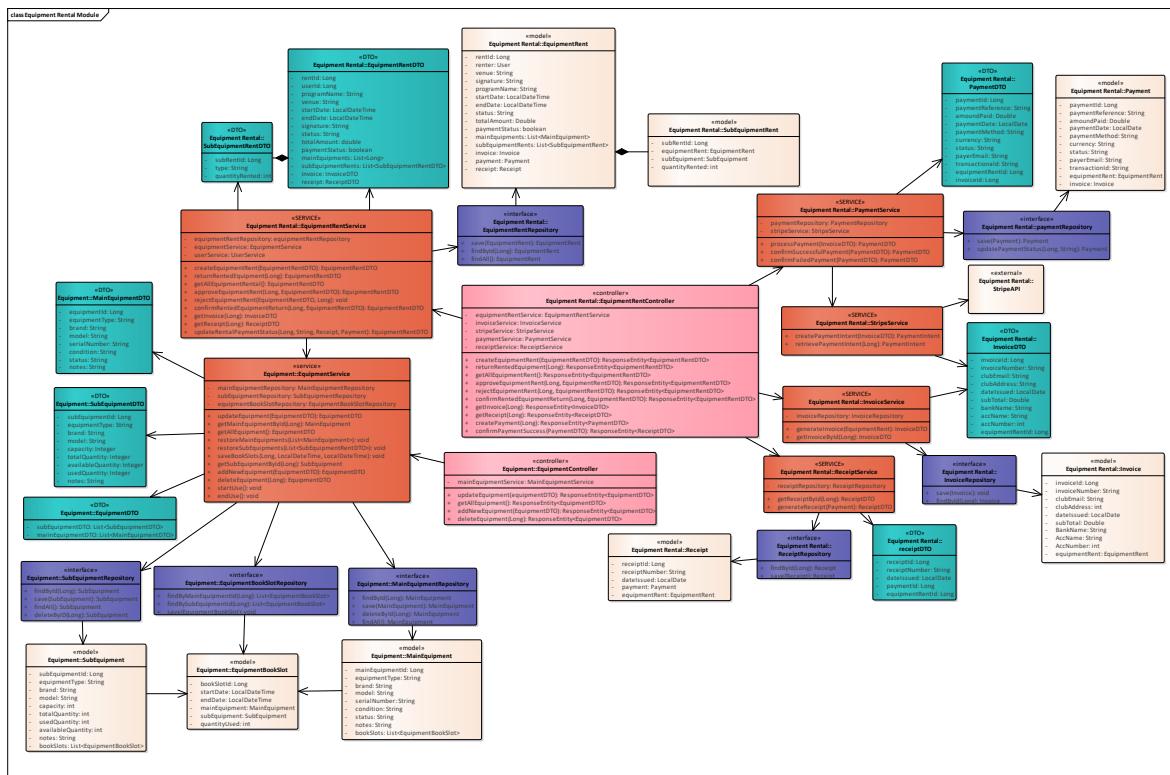


Figure 4.8 Class Diagram for Equipment Rent.

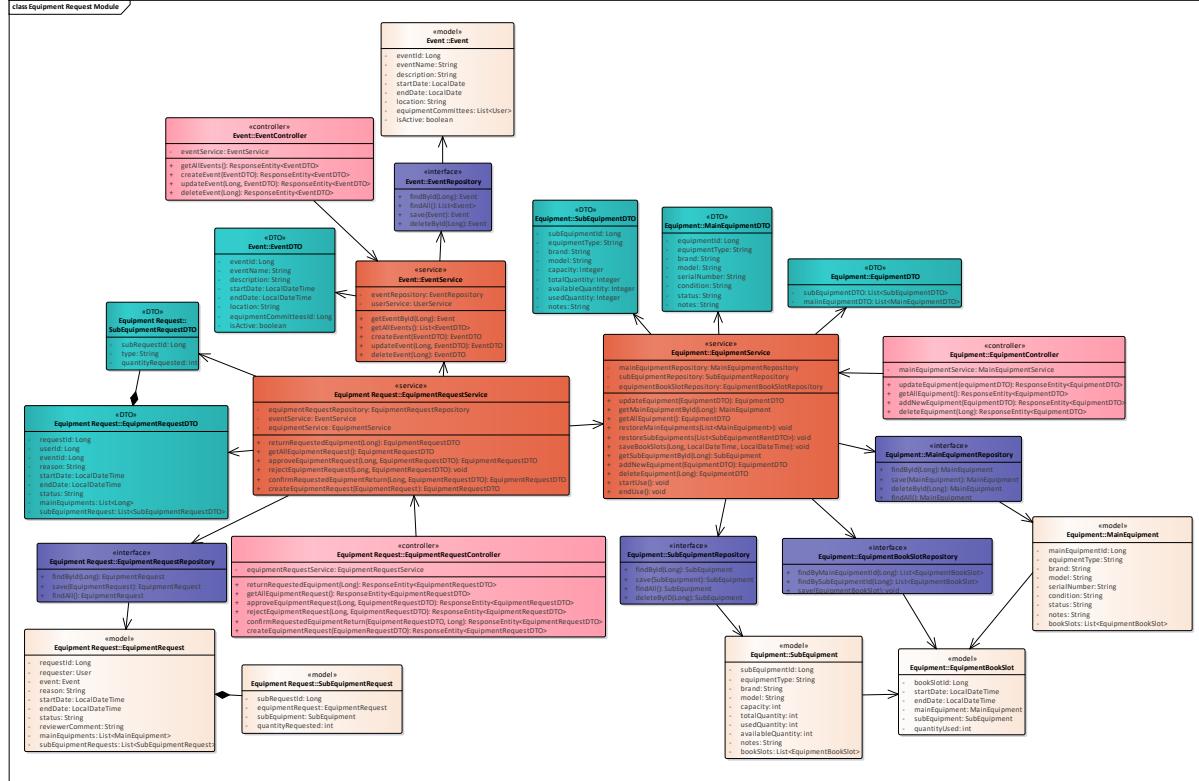


Figure 4.9 Class Diagram for Equipment Request.

4.4.2.4 Equipment Management Module Class Diagram

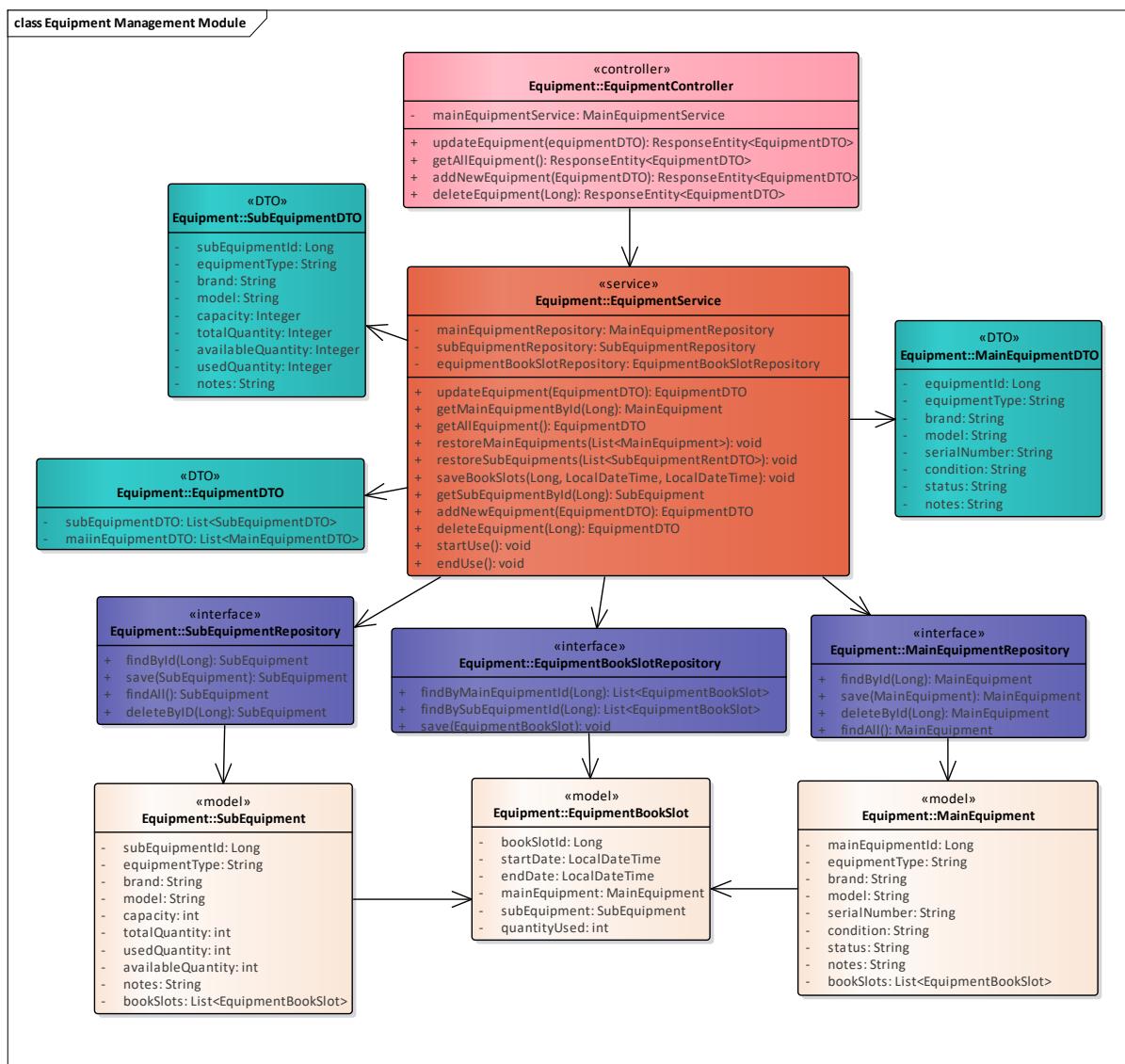


Figure 4.10 Class Diagram for Equipment Management Module.

4.4.2.5 Role and Event Management Module Class Diagram

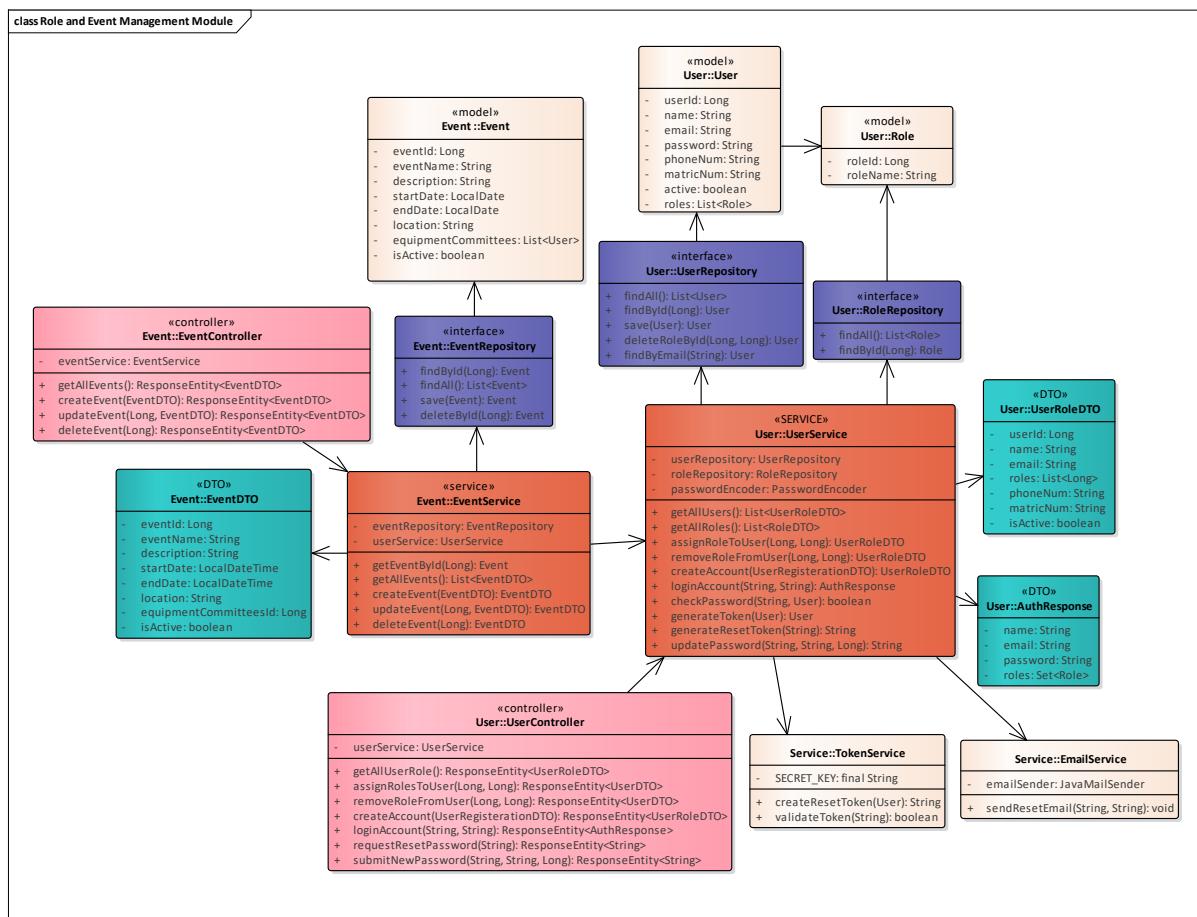


Figure 4.11 Class Diagram for Role and Event Management Module.

4.5 Information Viewpoint

4.5.1 Design Concerns

Information viewpoint defines the significant sticky data content of IFoto. The relevant issues are:

- **Persistent data structure:** relational database design housed with normalised entities to keep separation of concern.
- **Data content:** The user and role management, equipment management, rental and request management, and financial transactions.
- **Data management-Strategies:** these include: referential integrity constraints and Centralized database.
- **Accessing schemes of data:** foreign key and primary key relations so as to run efficient queries.
- **Definition of metadata:** Uniformity of data types and constraints on a data field in all the entities.

Data Storage and Organization

The key entities in data in this system are converted to a relational databases form where each of the entities in the form signify a certain concept in this systems business sense. The system will be able to process equipment renting and requests, equipment inventory, users and their roles as well as financial transactions on an interdependent data tabular structure.

The major data or system entities are stored in a relational database named as **IFoto Database** which comprises **12** entities that collectively manage the complete equipment rental lifecycle.

Table 4.6 Table of Description for each Entity

Entity Name	Description
Role	Defines user roles and permissions within the system.
User	Stores user account information including personal details, contact information, and authentication data
UserRole	Junction table linking users to their assigned roles, enabling role-based access control

UserEvent	Records user activities and system events for audit trails and activity monitoring
Event	Defines system events and activities that can be logged and tracked
EquipmentRequest	Manages rental requests submitted by users, including dates, status, and review comments
MainEquipment	Master catalog of available equipment with specifications, brands, models, and rental rates
EquipmentBookSlot	Tracks equipment availability and booking time slots for scheduling
SubEquipment	Manages individual equipment units and their specific attributes within equipment categories
SubEquipmentRequest	Links specific equipment units to rental requests with quantities
SubEquipmentRent	Records actual rental transactions for equipment units
Invoice	Generates billing documents for completed rentals
Payment	Processes and tracks payment transactions
Receipt	Provides payment confirmation and transaction records

4.5.2 Design View (ERD Diagram)

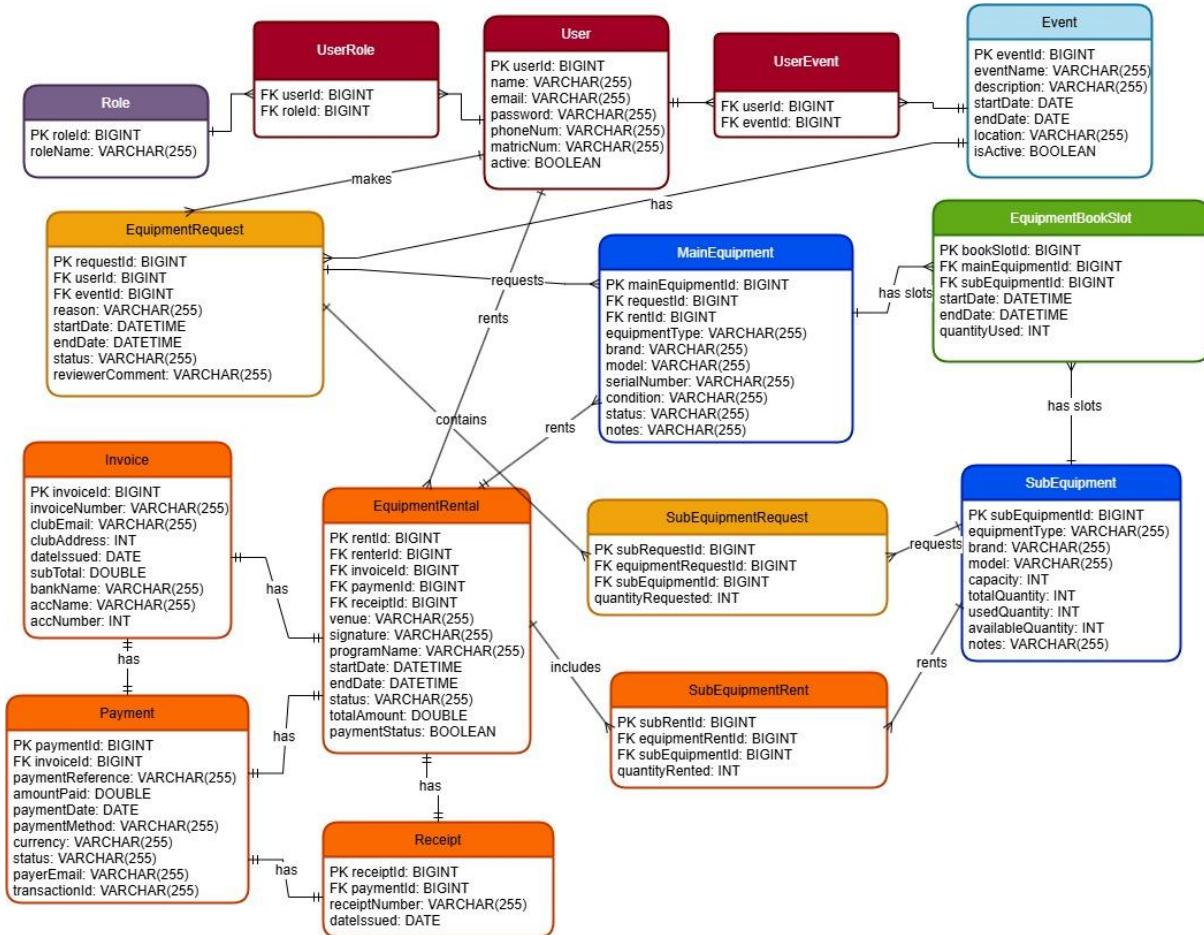


Figure 4.12 Entity Relationship Diagram for Inventory Management System for Kelab Fotokreatif (IFoto).

Table 4.7 Data Dictionary for Inventory Management System for Kelab Fotokreatif (IFoto).

Entity Name	Field Name	Data Type	Description
Role	roleId	BIGINT	Primary key, unique identifier for a role
	roleName	VARCHAR(255)	Name of the role
User	userId	BIGINT	Primary key, unique identifier for a user
	name	VARCHAR(255)	Full name of the user
	email	VARCHAR(255)	Email address of the user
	password	VARCHAR(255)	Password for user authentication
	phoneNum	VARCHAR(255)	Phone number of the user
	matricNum	VARCHAR(255)	Matriculation number of the user (e.g., student ID)
	active	BOOLEAN	Indicates if the user account is active

UserRole	userId	BIGINT	Foreign key, references User.userId
	roleId	BIGINT	Foreign key, references Role.roleId
UserEvent	userId	BIGINT	Foreign key, references User.userId
	eventId	BIGINT	Foreign key, references Event.eventId
Event	eventId	BIGINT	Primary key, unique identifier for an event
	eventName	VARCHAR(255)	Name of the event
	description	VARCHAR(255)	Description of the event
	startDate	DATE	Start date of the event
	endDate	DATE	End date of the event
	location	VARCHAR(255)	Location where the event takes place
	isActive	BOOLEAN	Indicates if the event is active
EquipmentRequest	requestId	BIGINT	Primary key, unique identifier for an equipment request
	userId	BIGINT	Foreign key, references User.userId
	eventId	BIGINT	Foreign key, references Event.eventId
	reason	VARCHAR(255)	Reason for the equipment request
	startDate	DATETIME	Start date and time of the request period
	endDate	DATETIME	End date and time of the request period
	status	VARCHAR(255)	Status of the equipment request
	reviewerComment	VARCHAR(255)	Comments from the reviewer
MainEquipment	mainEquipmentId	BIGINT	Primary key, unique identifier for main equipment
	requestId	BIGINT	Foreign key, references EquipmentRequest.requestId
	renterId	BIGINT	Foreign key, references User.userId
	equipmentType	VARCHAR(255)	Type of the equipment
	brand	VARCHAR(255)	Brand of the equipment
	model	VARCHAR(255)	Model of the equipment
	serialNumber	VARCHAR(255)	Serial number of the equipment
	condition	VARCHAR(255)	Condition of the equipment
	status	VARCHAR(255)	Status of the equipment
	notes	VARCHAR(255)	Additional notes about the equipment

SubEquipmentRequest	subEquipmentRequestid	BIGINT	Primary key, unique identifier for sub-equipment request
	equipmentRequestId	BIGINT	Foreign key, references EquipmentRequest.requestId
	subEquipmentId	BIGINT	Foreign key, references SubEquipment.subEquipmentId
	quantityRequested	INT	Quantity of sub-equipment requested
SubEquipment	subEquipmentId	BIGINT	Primary key, unique identifier for sub-equipment
	mainEquipmentId	BIGINT	Foreign key, references MainEquipment.mainEquipmentId
	equipmentType	VARCHAR(255)	Type of the sub-equipment
	brand	VARCHAR(255)	Brand of the sub-equipment
	model	VARCHAR(255)	Model of the sub-equipment
	capacity	INT	Capacity of the sub-equipment
	totalQuantity	INT	Total quantity available
	usedQuantity	INT	Quantity currently in use
	availableQuantity	INT	Quantity available for use
	notes	VARCHAR(255)	Additional notes about the sub-equipment
EquipmentRental	rentalId	BIGINT	Primary key, unique identifier for equipment rental
	renterId	BIGINT	Foreign key, references User.userId
	paymentId	BIGINT	Foreign key, references Payment.paymentId
	receiptId	BIGINT	Foreign key, references Receipt.receiptId
	venue	VARCHAR(255)	Venue of the rental
	signature	VARCHAR(255)	Signature for the rental agreement
	startDate	DATETIME	Start date and time of the rental
	endDate	DATETIME	End date and time of the rental
	status	VARCHAR(255)	Status of the rental
	totalAmount	DOUBLE	Total amount for the rental
SubEquipmentRent	subRentId	BIGINT	Primary key, unique identifier for sub-equipment rent
	equipmentRentId	BIGINT	Foreign key, references EquipmentRental.rentalId
	subEquipmentId	BIGINT	Foreign key, references SubEquipment.subEquipmentId

	quantityRented	INT	Quantity of sub-equipment rented
Payment	paymentId	BIGINT	Primary key, unique identifier for payment
	invoiceld	BIGINT	Foreign key, references Invoice.invoiceld
	paymentReference	VARCHAR(255)	Reference number for the payment
	amountPaid	DOUBLE	Amount paid
	paymentDate	DATE	Date of the payment
	paymentMethod	VARCHAR(255)	Method of payment
	currency	VARCHAR(255)	Currency of the payment
	status	VARCHAR(255)	Status of the payment
	payerEmail	VARCHAR(255)	Email of the payer
	transactionId	VARCHAR(255)	Transaction identifier
Invoice	invoiceld	BIGINT	Primary key, unique identifier for invoice
	equipmentRentId	BIGINT	Foreign key, references EquipmentRental.rentalld
	invoiceNumber	VARCHAR(255)	Invoice number
	clubEmail	VARCHAR(255)	Club email address
	clubAddress	VARCHAR(255)	Club address
	dateIssued	DATE	Date the invoice was issued
	subTotal	DOUBLE	Subtotal amount
	bankName	VARCHAR(255)	Name of the bank
	accName	VARCHAR(255)	Account name
	accNumber	INT	Account number
Receipt	receiptId	BIGINT	Primary key, unique identifier for receipt
	paymentId	BIGINT	Foreign key, references Payment.paymentId
	equipmentRentId	BIGINT	Foreign key, references EquipmentRental.rentalld
	receiptNumber	VARCHAR(255)	Receipt number
	dateIssued	DATE	Date the receipt was issued
EquipmentBookSlot	bookSlotId	BIGINT	Primary key, unique identifier for equipment book slot
	mainEquipmentId	BIGINT	Foreign key, references MainEquipment.mainEquipmentId
	subEquipmentId	BIGINT	Foreign key, references SubEquipment.subEquipmentId
	startDate	DATETIME	Start date and time of the booking slot
	endDate	DATETIME	End date and time of the booking slot

	quantityUsed	INT	Quantity of equipment used in the slot
--	--------------	-----	----------------------------------------

4.6 Interface Viewpoint

4.6.1 Design Concerns

The Interface viewpoint provides information designers, programmers, and testers the means to know how to correctly use the services provided by a design subject. This description includes the details of external and internal interfaces not provided in the SRS. This viewpoint consists of a set of interface specifications for each entity and focuses on how various components and users interact with the system.

On the user side, the system has a web-based interface, which is responsive and easy to use created with the application of React. Actors like Administrator, Equipment Executive Committee, Equipment Committee Member, and the User can communicate to the system using different pages say Rent Equipment, Request Equipment, Manage Events and Manage Equipment. The pages are developed in such a way as to give real-time feedback to user input. An example is that during the selection of equipment, availability of an equipment as well as open booking slots is shown by the system, and on confirmation of returns, users receive immediate feedback regarding the condition and status of the equipment.

In the case of software interfaces, the system comprises RESTful APIs constructed by Spring Boot that enables the backend services to communicate with the frontend (React with RTK Query). Controllers expose these APIs and these are organized around the major use cases such as log in, assigning role, rentals management and tracking inventory. Also, the API endpoints are well described, and follows the URL structures and request-response patterns based on Data Transfer Objects (DTOs).

Regarding the internal interfaces, services and repositories communicate by invoking methods that hide data access and business operations. DTOs just serve as the interlayers to decouple the issues between the controller and service layers thereby providing modularity and testability.

Even though, there are no special interfaces used by the hardware (other than standard client hardware types e.g., laptops, desktops, mobile devices) to connect to the system, the system is conceptually not hardware specific and responsive across devices.

4.6.2 Design View (Interfaces)

4.6.2.1 User Interface for Equipment Request and Rentals Module

The screenshot shows the 'Rent Equipment' section of the IFoto application. At the top, there is a search bar labeled 'Search product, supplier, order'. On the right side, there are icons for a bell通知 and a user profile. Below the search bar, there are two tabs: 'Rent Equipment' (selected) and 'Rental List'. A sidebar on the left includes links for 'Rent Equipment', 'Orders', 'Settings', and 'Log Out'. The main content area is divided into two sections: 'Camera' and 'Camera Lens'. Each section has a table with columns for No., Brand, Model, Condition, Status, Notes, and Action (with 'Add' and 'Edit' buttons). The 'Camera' section contains 8 entries, and the 'Camera Lens' section contains 2 entries.

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	SD Mark IV	Good	Available	-	Add
2	Canon	SD Mark IV	Good	Available	-	Add
3	Canon	SD Mark II	Broken	Unavailable	EU	Add
4	Canon	SD Mark II	Good	Available	-	Add
5	Canon	60D	Good	Available	-	Add
6	Nikon	D90	Broken	Unavailable	KBV	Add
7	Nikon	D90	Good	Available	-	Add
8	Nikon	D7000	Good	Available	-	Add

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Add
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	Add

Figure 4.13 Interface of List of Equipment (Rent Equipment)

The screenshot shows the 'Cart' page of the IFoto application. At the top, there is a search bar labeled 'Search product, supplier, order'. On the right side, there is a 'Rental Form' button. Below the search bar, there is a breadcrumb navigation 'Equipment List > Cart'. A sidebar on the left includes links for 'Rent Equipment', 'Orders', 'Settings', and 'Log Out'. The main content area is divided into two sections: 'Camera' and 'Camera Lens'. Each section has a table with columns for No., Brand, Model, Condition, Status, Notes, and Action (with 'Edit' and 'Remove' buttons). The 'Camera' section contains 3 entries, and the 'Camera Lens' section contains 2 entries.

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	SD Mark IV	Good	Available	-	Edit Remove
2	Canon	SD Mark IV	Good	Available	-	Edit Remove
3	Canon	SD Mark II	Good	Available	EU	Edit Remove

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Edit Remove
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	Edit Remove

Figure 4.14 Interface of Cart Page (Rent Equipment)

Rental Form

Renter Name	<input type="text"/>
Phone No.	<input type="text"/>
Email	<input type="text"/>
Matric No.	<input type="text"/>
Program Name	<input type="text"/>
Venue	<input type="text"/>
Start Date	<input type="text"/>
End Date	<input type="text"/>

Cancel **Submit Form**

Figure 4.15 Interface of Rental Form (Rent Equipment)

Rental List

FutureReady (ID123124123123)	Pay Now
1A/Krighnarajapuram, 3 rd street sulur	
Muhammad Taufiq	
20/6/2025 - 25/6/2025	Status: Approved (Unpaid)
	View Invoice View Receipt
CONVOCATION - 68 (ID123124123123)	Pay Now
1A/Krighnarajapuram, 3 rd street sulur	
Ahmad Hazim	
23/6/2025 - 26/6/2025	Status: Rejected
	View Invoice View Receipt
WEDDING (ID123124123123)	Return Equipment
1A/Krighnarajapuram, 3 rd street sulur	
Nabil Iman	
29/6/2025 - 1/7/2025	Status: Approved (Paid)
	View Invoice View Receipt

Figure 4.16 Interface of List of Equipment Rents (Rent Equipment)

The screenshot shows the IFoto Rent Equipment interface. At the top left is the logo and navigation links for Rent Equipment and Orders. The main content area has a search bar and a user profile icon at the top right. Below is a section titled "FutureReady details" with a breadcrumb trail: Equipment List > Rental List > FutureReady. A modal window for "FutureReady (ID123124123123)" is open, displaying address (1A/Krithnarajapuram, 3 rd street sulur), total amount (RM 75.00), date range (20/6/2025 - 25/6/2025), status (Approved), and buttons for Pay Now, Cancel, View Invoice, and View Receipt. Below the modal is a table for "Camera" equipment, listing three items: Canon 5D Mark IV, Canon 5D Mark IV, and Canon 5D Mark II, all in good condition and available. There are edit and remove buttons for each row. A "Filters" button is also present. Below the camera table is another for "Camera Lens", showing one item: Canon EF 50mm Prime, in good condition and available. There are edit and remove buttons for this row as well. A "Filters" button is also present.

Figure 4.17 Interface of Specific Equipment Rent Details (Rent Equipment)

The screenshot shows the IFoto Rent Equipment interface. At the top left is the logo and navigation links for Rent Equipment and Orders. The main content area has a search bar and a user profile icon at the top right. Below is a section titled "Invoice" with a breadcrumb trail: Equipment List > Rental List > FutureReady > Invoice. A modal window displays an invoice from "UTM PHOTO" for a rental period from 19/05/2025 to 24/05/2025. The invoice details include the customer's name (HAKIM HAKIM), account number (MUHAMMAD ALIF NABIL), and payment information (MOSR BANK, Account Name: MUHAMMAD ALIF NABIL, Account No.: 1234567890). The invoice lists two items: 1 Set Camera (Body + Lens biasa) and Speedlight, with a subtotal of RM95 and a total of RM95. The payment method is cash. The invoice is signed by "Accepted by HAKIM HAKIM".

Figure 4.18 Interface of Invoice (Rent Equipment)

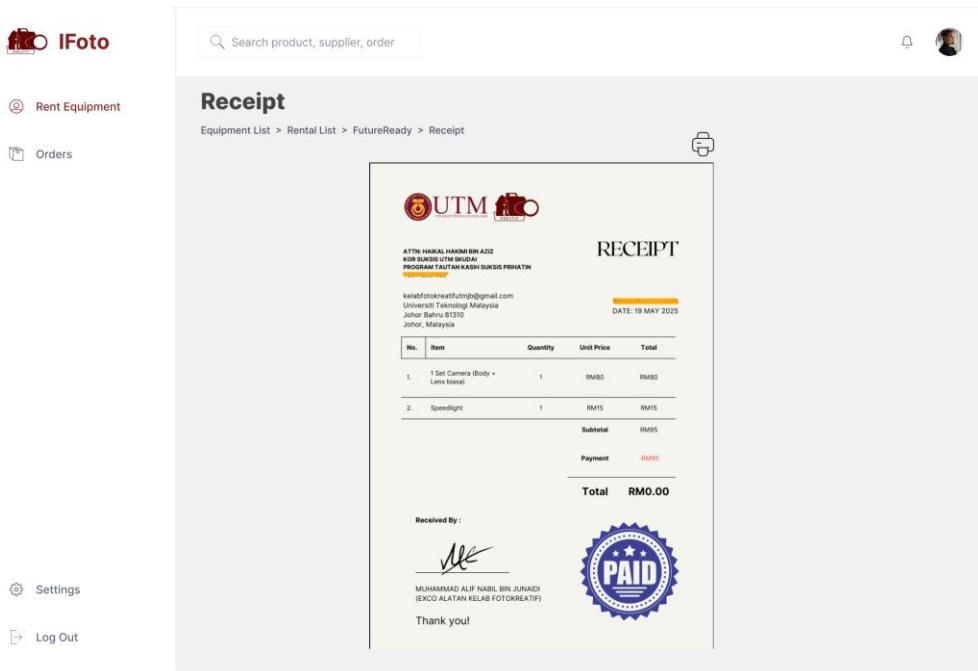


Figure 4.19 Interface of Receipt (Rent Equipment)

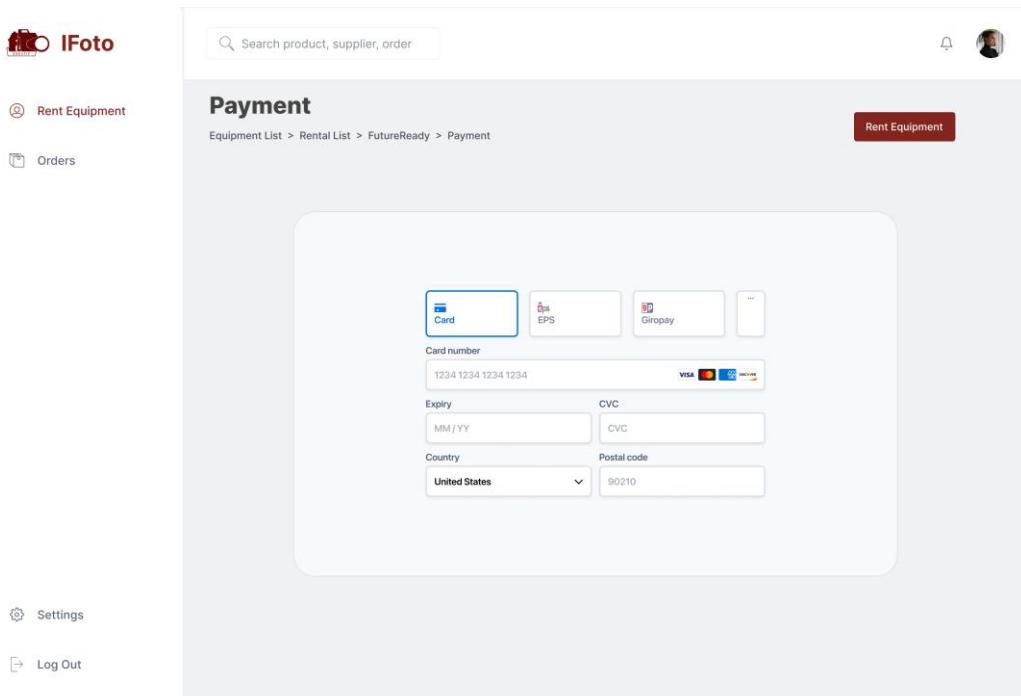


Figure 4.20 Interface of Payment Page (Rent Equipment)

The screenshot shows the 'FutureReady details' page within the iFoto Rent Equipment application. At the top, there's a search bar and a user profile icon. Below the header, a breadcrumb navigation shows 'Equipment List > Rental List > FutureReady'. The main content area displays a summary for 'FutureReady (ID123124123123)'. The summary includes the address '1A/Krihnarajapuram, 3 rd street sulur', total amount 'RM 75.00', contact 'Muhammad Taufiq', rental period '20/6/2025 - 25/6/2025', and status 'Approved'. A red 'Confirm Return' button is visible. Below this, two tables are shown: 'Camera' and 'Camera Lens', both with columns for No., Brand, Model, Condition, Status, Notes, and Action (Edit, Remove). The 'Camera' table has three entries, all marked as Available. The 'Camera Lens' table has one entry, also marked as Available.

Figure 4.21 Interface of Equipment Return Page (Rent Equipment)

The screenshot shows the 'Cart' page within the iFoto Rent Equipment application. The header includes a search bar and a user profile icon. The breadcrumb navigation shows 'Equipment List > Cart'. On the right, a red 'Rental Form' button is visible. The main content area displays a 'Camera' table with three entries, all marked as Available. A modal window is overlaid on the page, featuring a green checkmark icon and the word 'SUCCESS'. Inside the modal, a message reads: 'You have successfully submitted a Equipment Rental Form. Please wait approval before proceed with payment.' Below the modal, a 'Camera Lens' table is partially visible, showing two entries, both marked as Available. The bottom of the screen shows navigation links for 'Settings' and 'Log Out'.

Figure 4.22 Example of Interface for Successful Action (Rent Equipment)

Figure 4.23 Interface of List of Event (Request Equipment)

Figure 4.24 Interface of List of Equipment (Request Equipment)

Figure 4.25 Interface of Cart Page (Request Equipment)

Figure 4.26 Interface of Equipment Request Form (Request Equipment)

The screenshot shows the iFoto Rent Equipment application interface. At the top left is the logo 'iFoto'. To its right is a search bar with the placeholder 'Search product, supplier, order' and a user profile icon. Below the header, there are navigation links: 'Rent Equipment', 'Orders', 'Settings', and 'Log Out'. The main content area has a title 'KBV details' and a breadcrumb trail: 'Equipment List > Rental List > FutureReady'. A sub-section titled 'KBV' contains a summary box with the ID 'KBV (ID123124123123)', address '1A/Krighnarajapuram, 3 rd street sulur', contact 'Ahmad Saifudin', rental period '20/6/2025 - 25/6/2025', and status 'Ongoing'. A 'Return Equipment' button is also present. Below this is a table for 'Camera' equipment, listing three items: 1 Canon 5D Mark IV (Good, Available), 2 Canon 5D Mark IV (Good, Available), and 3 Canon 5D Mark II (Good, Available). Each row includes 'Edit' and 'Remove' buttons. A 'Filters' button is located at the top right of the table. Below the camera table is another table for 'Camera Lens', showing one item: 1 Canon Prime Canon EF 50mm (Good, Available). This table also includes a 'Filters' button.

Figure 4.27 Interface of Specific Equipment Request Details (Request Equipment)

4.6.2.2 User Interface for Equipment Management Module

The screenshot shows the 'Club's Equipment List' section of the application. On the left sidebar, there are links for Rent Equipment, Manage Equipment, Equipment Requests, Equipment Returns, Orders, Settings, and Log Out. The main area displays two tables: 'Camera' and 'Camera Lens'. Each table has columns for No., Brand, Model, Serial Number, Condition, Status, Notes, and Action (Edit Equipment, Remove). The 'Camera' table contains 8 entries, and the 'Camera Lens' table contains 2 entries. A search bar at the top is set to 'Search product, supplier, order'. Top right icons include a bell and a user profile.

No.	Brand	Model	Serial Number	Condition	Status	Notes	Action
1	Canon	5D Mark IV	XXX12390	Good	Available	-	Edit Equipment Remove
2	Canon	5D Mark IV	XXX12390	Good	Available	-	Edit Equipment Remove
3	Canon	5D Mark II	XXX12390	Broken	Unavailable	EU	Edit Equipment Remove
4	Canon	5D Mark II	XXX12390	Good	Available	-	Edit Equipment Remove
5	Canon	60D	XXX12390	Good	Available	-	Edit Equipment Remove
6	Nikon	D90	XXX12390	Broken	Unavailable	KBV	Edit Equipment Remove
7	Nikon	D90	XXX12390	Good	Available	-	Edit Equipment Remove
8	Nikon	D7000	XXX12390	Good	Available	-	Edit Equipment Remove

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Edit Equipment Remove
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	Edit Equipment Remove

Figure 4.28 Interface of List of Club's Equipment (Manage Equipment)

The screenshot shows the 'Update Equipment Form' dialog box over the 'Club's Equipment List' background. The dialog has fields for Drag image here or Browse image, Brand (set to Canon), Model (set to Canon), Serial Number (set to XXX12390), Condition (set to Good), Status (set to Available), and Notes (set to EU). It includes a 'Cancel' and 'Save Equipment' button. The background shows the same equipment list as Figure 4.28, with the camera table visible.

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Edit Equipment Remove
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	Edit Equipment Remove

Figure 4.29 Interface of Update Equipment Form (Manage Equipment)

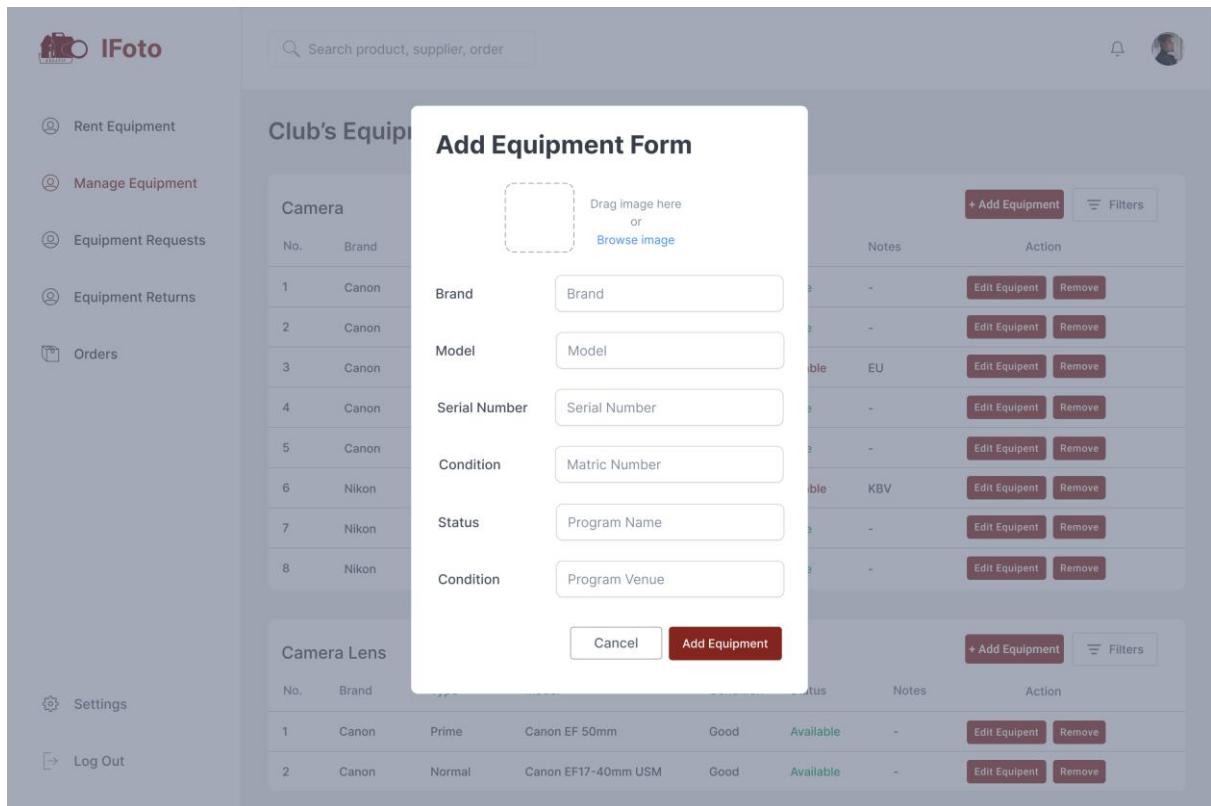


Figure 4.30 Interface of Add Equipment Form (Manage Equipment)

Figure 4.31 Interface of List of Equipment Request and Rentals (Manage Equipment Requests and Rentals)

The screenshot shows the 'Manage Equipment Requests and Rentals' section of the IFoto software. On the left sidebar, there are several navigation items: Rent Equipment, Manage Equipment, Equipment Requests (highlighted in red), Equipment Returns, and Orders. The main content area has a search bar at the top. Below it, a title 'Manage Equipment Requests and Rentals' is followed by a breadcrumb 'Requests and Rentals List > FutureReady'. A section titled 'FutureReady' displays a request for 'FutureReady (ID123124123123)'. The details include address '1A/Krihnarajapuram, 3 rd street sulur', total amount 'RM 75.00', contact 'Muhammad Taufiq', and date range '20/6/2025 - 25/6/2025'. Status is 'Approved'. Buttons for 'Approve' (green) and 'Reject' (red) are shown. Below this, there are two tables: 'Camera' and 'Camera Lens', each with columns for No., Brand, Model, Condition, Status, Notes, and Action. The 'Camera' table lists three items: Canon 5D Mark IV, Canon 5D Mark IV, and Canon 5D Mark II. The 'Camera Lens' table lists one item: Canon Prime EF 50mm. At the bottom left, there are 'Settings' and 'Log Out' links.

Figure 4.32 Interface of Managing Equipment Request and Rentals (Manage Equipment Requests and Rentals)

The screenshot shows the 'Manage Equipment Returns' section of the IFoto software. The left sidebar includes Rent Equipment, Manage Equipment, Equipment Requests, Equipment Returns (highlighted in red), and Orders. The main content area features a search bar. Below it, a title 'Manage Equipment Returns' and a subtitle 'Equipment Return List'. A section titled 'Manage Equipment Rent Return' shows a return for 'FutureReady (ID123124123123)' with details: address '1A/Krihnarajapuram, 3 rd street sulur', contact 'Muhammad Taufiq', date range '20/6/2025 - 25/6/2025', and status 'Pending Return'. A 'Manage' button is present. Another section titled 'CONVOCATION - 68 (ID123124123123)' shows a return for 'CONVOCATION - 68 (ID123124123123)' with details: address '1A/Krihnarajapuram, 3 rd street sulur', contact 'Ahmad Hazim', date range '23/6/2025 - 26/6/2025', and status 'Returned'. A 'Manage' button is present. A final section titled 'Manage Equipment Request Return' shows a return for 'KBV (ID123124123123)' with details: address '1A/Krihnarajapuram, 3 rd street sulur', contact 'Muhammad Taufiq', date range '20/6/2025 - 25/6/2025', and status 'Approved'. A 'Manage' button is present. At the bottom left, there are 'Settings' and 'Log Out' links.

Figure 4.33 Interface of List of Equipment Returns (Manage Equipment Returns)

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	5D Mark IV	Good	Available	-	<button>Manage</button> <button>Remove</button>
2	Canon	5D Mark IV	Good	Available	-	<button>Manage</button> <button>Remove</button>
3	Canon	5D Mark II	Good	Available	EU	<button>Change</button> <button>Remove</button>

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	<button>Manage</button> <button>Remove</button>

Figure 4.34 Interface of Managing Equipment Returns (Manage Equipment Returns)

4.6.2.3 User Interface for Role and Event Management Module

Event List

No.	Name	Description	Location	Status	Start Date	End Date	Action
1	KBV	Video Lesson	SUB	Ongoing	23/5/2025	28/5/2025	Manage Event Manage Member
2	MMM	Minggu Mesra Mahasiswa	DSI	Ongoing	23/5/2025	28/5/2025	Manage Event Manage Member
3	Wedding	Perkahwinan Adam	Pondok	Upcoming	23/6/2025	28/6/2025	Manage Event Manage Member

Figure 4.35 Interface of List of Event

Manage Member

Event List > Manage Equipment Committee Member

No.	Name	email	Phone No.	Matric No.	Roles	Action
1	Muhammad Taufiq	example@gmail.com	011-XXXXXXX	A22ECXXX	View Roles	+ Add
2	Muhammad Nabil	example@gmail.com	011-XXXXXXX	A22EMXXX	View Roles	+ Add
3	Ahmad Saifudin	example@gmail.com	011-XXXXXXX	A23ESXXX	View Roles	+ Add

No.	Name	email	Phone No.	Matric No.	Roles	Action
1	Muhammad Alif	example@gmail.com	011-XXXXXXX	A22ECXXX	View Roles	Remove
2	Ahmad Hazim	example@gmail.com	011-XXXXXXX	A22EMXXX	View Roles	Remove

Figure 4.36 Interface of Manage Club Members

The screenshot shows the iFoto software interface. At the top left is the logo 'iFoto'. To its right is a search bar with the placeholder 'Search product, supplier, order'. On the far right are a bell icon and a user profile picture. The main area has a light gray background with a darker gray modal window in the center.

Event List

Add Event Form

List of Events

No.	Name
1	KBV
2	MMM
3	Wedding

Event Name: Renter Name

Description: Phone Number

Location: Email

Matric No.: Matric Number

Start Date: dd/mm/yy

End Date: dd/mm/yy

Assign Equipment Committee Member:

+ Add

Cancel **Create Event**

Filters

Action

Start Date	End Date	Action
5/5/2025	28/5/2025	Manage Event Manage Member
5/5/2025	28/5/2025	Manage Event Manage Member
6/6/2025	28/6/2025	Manage Event Manage Member

Rent Equipment

Manage Event

Orders

Settings

Log Out

Figure 4.37 Interface of Add Event Form

4.6.2.4 Software Interface

The software interface for Inventory Management System for Kelab Fotokreatif are shown in the table 4.8 below.

Table 4.8 Software Interface for IFoto

Software Requirements		
Computer/Laptop		
Software	Requirement Details	
Google Chrome	Version 90 or later (Latest: Version 124+)	
Microsoft Edge	Version 90 or later (Latest: Version 124+)	
Safari	Version 13 or later (Latest: Version 17+)	
Visual Studio Code	Version 1.60 or later	
Programming Languages	Java	JDK 17 or later
	JavaScript (ES6+)	Any modern browser
Frameworks	Spring Boot	Version 2.7.x or 3.x (preferably with JDK 17 support)
	ReactJS	Version 18 or later
MySQL	Version 8.0 or later	
GitHub	access via browser or Git CLI v2.30 or later	
Smartphone		
Software	Requirement Details	
Google Chrome on Android	Version 90 or later (Latest: Version 124+)	
Safari on IOS	Version 13 or later (Latest: Version 17+)	

4.6.2.5 Hardware Interface

The minimum hardware interface for Inventory Management System for Kelab Fotokreatif are shown in the table 4.9 below. Any system specification better than what stated below can run IFoto smoothly.

Table 4.9 Hardware Interface for IFoto

Software Requirements	
Computer/Laptop	
Software	Requirement Details (Minimum Requirements)
Operating System	Windows 10 (64-bit) or later version
	macOS 10.15 (Catalina) or later version
	Ubuntu 20.04 LTS or equivalent or later version
Processor	Intel Core i3 (8th Gen or later) or higher
	AMD Ryzen 3 (3000 series or later) or higher
	Apple M1 or M2 (for macOS users)
Smartphone	
Software	Requirement Details (Minimum Requirements)
Operating System	Android 10 (Q) or newer
	iOS 13 or newer
Processor	Snapdragon 665 / Exynos 9611 / Equivalent or higher
	Apple A11 Bionic (iPhone 8) or higher

4.7 Structure Viewpoint

4.7.1 Design Concerns

Structure point of view explains to the user and the developer on how the packages interact within the system and gives whole picture about the architectural structure and internal organization of the software system. Such a perspective covers a number of important design issues.

The structure viewpoint shows how the system is separated into individual and cohesive packages that describe its own functionality. Each package is a subsystem which consist of sets of components with well-defined responsibilities, such as Equipment Management Module, Authentication and Authorization Module, Equipment Request and Rental Module, and Role and Event Management module contains their own well-defined components that work together to build package. This modular approach encourages maintainability, testability, and parallel development by different teams.

Structure viewpoint conveys the relationship amongst packages portrayed by directional arrows. Such a visualization assists developers to comprehend the impacts of modifications throughout the system and displays possible locations where tight coupling can induce issues in maintenance.

System scalability is coped with in the form of the modular structure that enables the addition of new packages without substantial changes in existing components. The definite division of concern between the functionalities (Equipment Management Module, Authentication and Authorization Module, Equipment Request and Rental Module, and Role and Event Management module) allows its gradual increases in specific areas of the system by load.

4.7.2 Design View (Package Diagram)

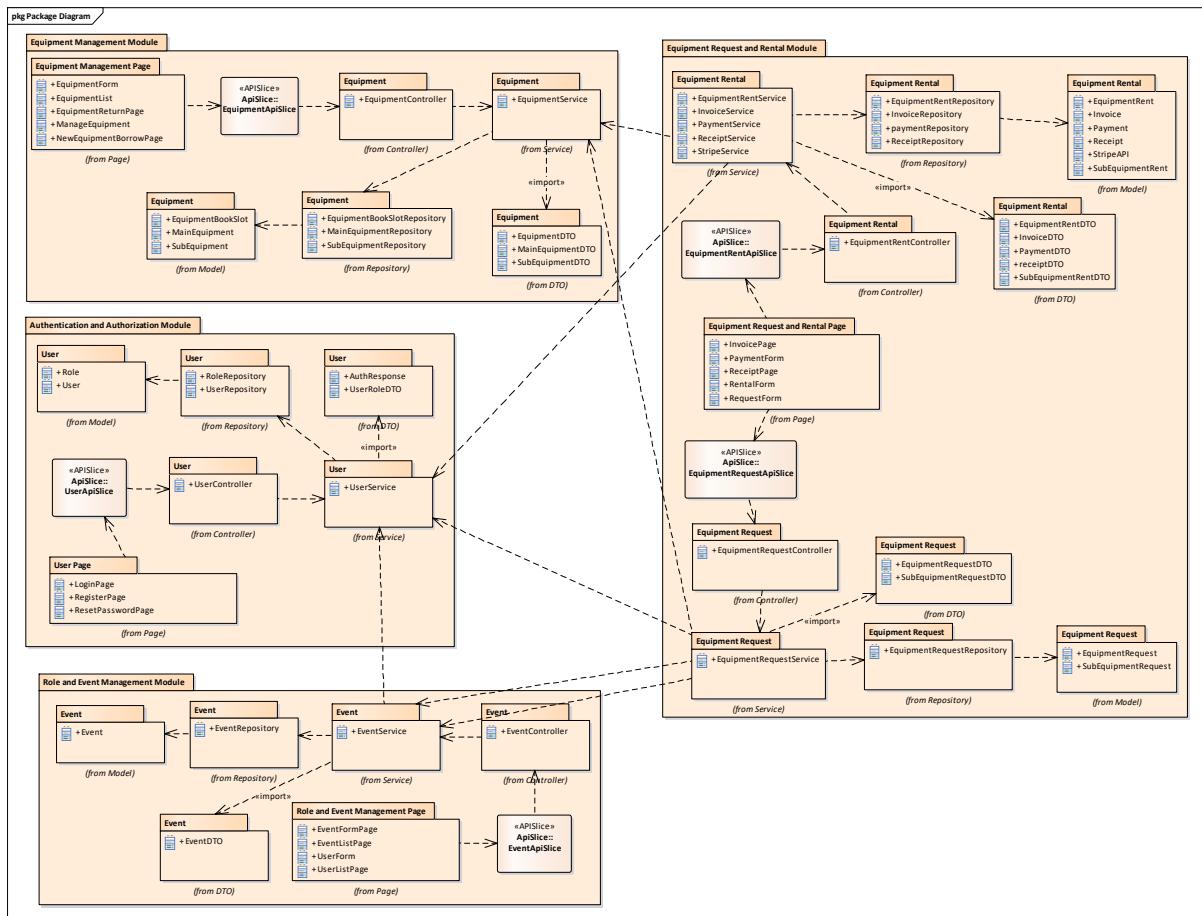


Figure 4.38 Package Diagram for Inventory Management System for Kelab Fotokreatif (IFoto).

Package Diagram in Figure 437 illustrate the overview of Inventory Management System for Kelab Fotokreatif (IFoto). This system consists of four main modules which are Equipment Management Module, Authentication and Authorization Module, Equipment Request and Rental Module, and Role and Event Management module. Each of this module follows the Model-View-Controller (MVC) architectural style and pattern where it consists of Page and APISlice as View, Controller, Service and Data Transfer Object (DTO) as Controller and Repository and Model as Model of the system.

1. Authentication and authorization module

This module authenticates accounts; grants access depending on role and handles user accounts and role delegates. The class examples according to each Subcategory in MVC architecture style are below:

Model:

- User represents their data (name, email, roles).
- Role identifies the roles of the user and their authorities.

Repository:

- UserRepository offer an interface to carry CRUD operation with respect to users.
- RoleRepository deals with roles queries and mappings.

Service:

- AuthService manages the user logging in, password hashing and token.
- RoleService takes responsibility of authorization queries and assignments of roles.

Controller:

- AuthController processes the endpoints of login, register and reset a password.
- RoleController is used to create and delete endpoints to assign and remove roles.

DTO:

- LoginDTO contains frontend to backend login credentials.
- User DTO transfers the user profile details between services.

API Slice:

- UserAPISlice.js is responsible to make communication with the backend via API and it is used to Log users in (sending credentials and receiving tokens), Create new accounts, Reset passwords.

Page:

- LoginPage.jsx, a form is presented to the user in which they can log in.
- RegisterPage.jsx offer the registration form fields that are responsible send the user information at the backend to create an account.

2. Equipment Management Module

This module oversees keeping a trace, update and hold available equipment (main or sub). The examples of classes that are organised by the MVC style of architecture are represented below:

Model:

- MainEquipment stands for non-continuous equipment such as cameras.
- SubEquipment Equipped with bulk-based equipment such as batteries, in quantities.

Repository:

- MainEquipmentRepository loads the data of main equipment.
- SubEquipmentRepository finds equipment sub-equipment.

Service:

- EquipmentService act as service centre logic to retrieve or update equipment and book time slots.

Controller:

- EquipmentController has APIs which can be used to retrieve equipment inventory and to update the same.

DTO:

- MainEquipmentDTO exchanges data with the frontend behind it (camera/lenses data).
- SubEquipmentDTO copies mass sub-equipment data such as quantity and type.

API Slice:

- EquipmentApiSlice sends calls to back-end services in order to get all equipment available on the system, update equipment information (such as its status, its condition), or to get equipment using the ID.

Page:

- ManageEquipmentPage.jsx displays a list of all equipment. Admin or committee can edit details like condition or notes.

- EquipmentDetailPage.jsx shows full information of a selected equipment, including its booking slots (availability).

3. Equipment Request and Rental Module

This module would enable the users to request and rent equipment. It has equipment request logic, rent, confirm returns, and booking slots tracking. The examples of the classes classified as per MVC architectural style are shown below:

Model:

- EquipmentRent records the information of an ongoing hire.
- EquipmentRequest is used to capture equipment request related to events.
- SubEquipmentRent, SubEquipmentRequest records are records related to the Quantity based rent/request.

Repository:

- EquipmentRentRepository deals with every persistence and query of EquipmentRent data.
- EquipmentRequestRepository store and retrieve EquipmentRequest data

Service:

- EquipmentRentService coordinates rental logic, approval, and return handling.
- EquipmentRequestService handles request submission and approval workflow.

Controller:

- EquipmentRentService organizes the logic of accepting the equipment rent, approving its availability, and organizes the returns.
- EquipmentRequestService deals with the creating and acceptance of equipment requests.

DTO:

- EquipmentRentDTO object carries over EquipmentRent Data across layers and controllers.
- EquipmentRequestDTO object carries event data and equipment details across layers and controllers.

API Slice:

- EquipmentRentApiSlice deals with equipment Rent creation, confirmation of returns, and fetching equipment rent and request information as well as payment.

Pages:

- RentEquipmentPage.jsx allows members and the general user to rent out available equipment.
- RequestEquipmentPage.jsx is solely accessible to the members of the Equipment Committee to request equipment needed in the upcoming events.

4. Role and Event Management Module

This module facilitates the assignment of user roles, and organizing and management of events. It is vital in connection of equipment usage with real events. The examples of classes of each Subcategory in MVC architecture style are presented below:

Model

- Event contains event information like name, date and assigned committee.

Repository:

- EventRepository gives interface to carry out CRUD operations on Event model.

Service:

- EventService manages event creation, editing, and member role assignment.

Controller

- EventController manages CRUD operations for events.

DTO

- EventDTO transfers event info between frontend and backend.

API Slice:

- EventApiSlice coordinates fetching event details, assigning, or removing roles, managing event details (creating, editing, assigning users to events)

Pages:

- AssignRolePage.jsx enables the admin to be able to see the entire users and assign their roles.
- ManageEventPage.jsx enables admins to create or edit event details together with assigning the committee members.

4.8 Interaction Viewpoint

4.8.1 Design Concerns

The help of sequence diagrams is given that defines functionalities of the system. Additionally, it determines methodologies of interaction between entities, the question of why, where, how, and at which level activities are carried out in the architecture structure.

The interaction perspective depicts the dynamic behaviour of the system in the sense of how various components interact with each other to achieve certain use cases which is defined in terms of business logic of the stakeholders. The sequence diagrams show the flow of messages, functions calls and data among the object entities that offer the perfect comprehension of a case of use.

The viewpoint describes on how the system manage alternate and exceptional flow during entity interactions. It defines how the system performs alternate flow, error propagation, recovery mechanisms, and fallback procedures when normal flows are interrupted to ensure system robustness and reliability.

4.8.2 Design View (Sequence Diagram)

4.8.2.1 Authentication and Authorization Module

4.8.2.1.1 UC001: Register An Account

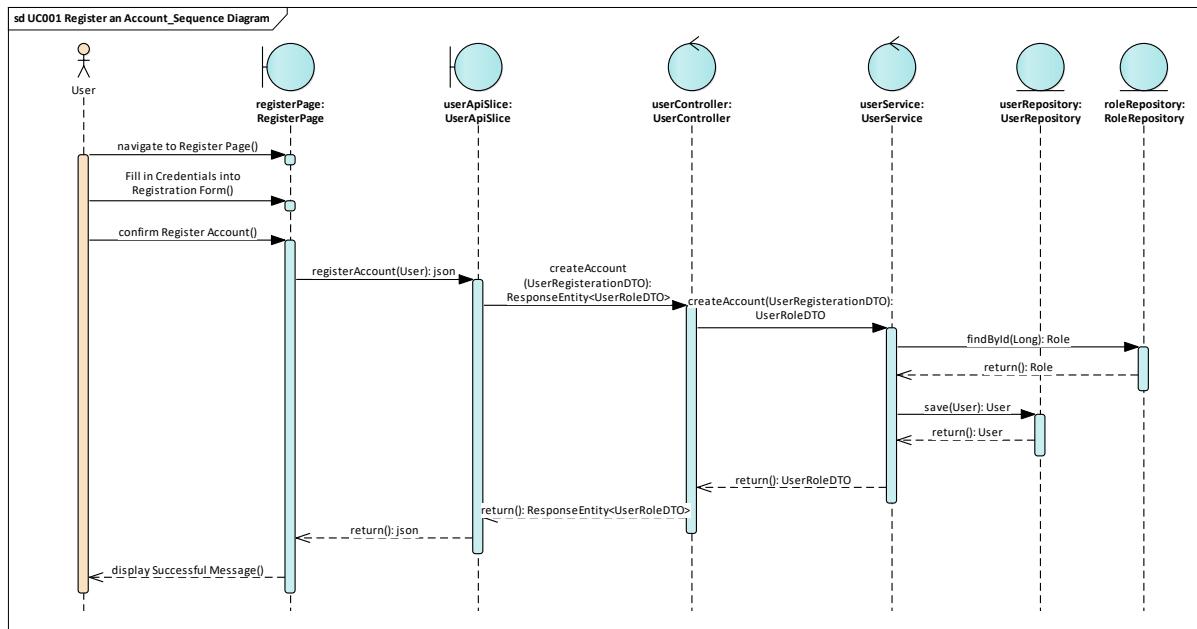


Figure 4.39 Sequence Diagram for UC001 Register an Account

4.8.2.1.2 UC002: Login

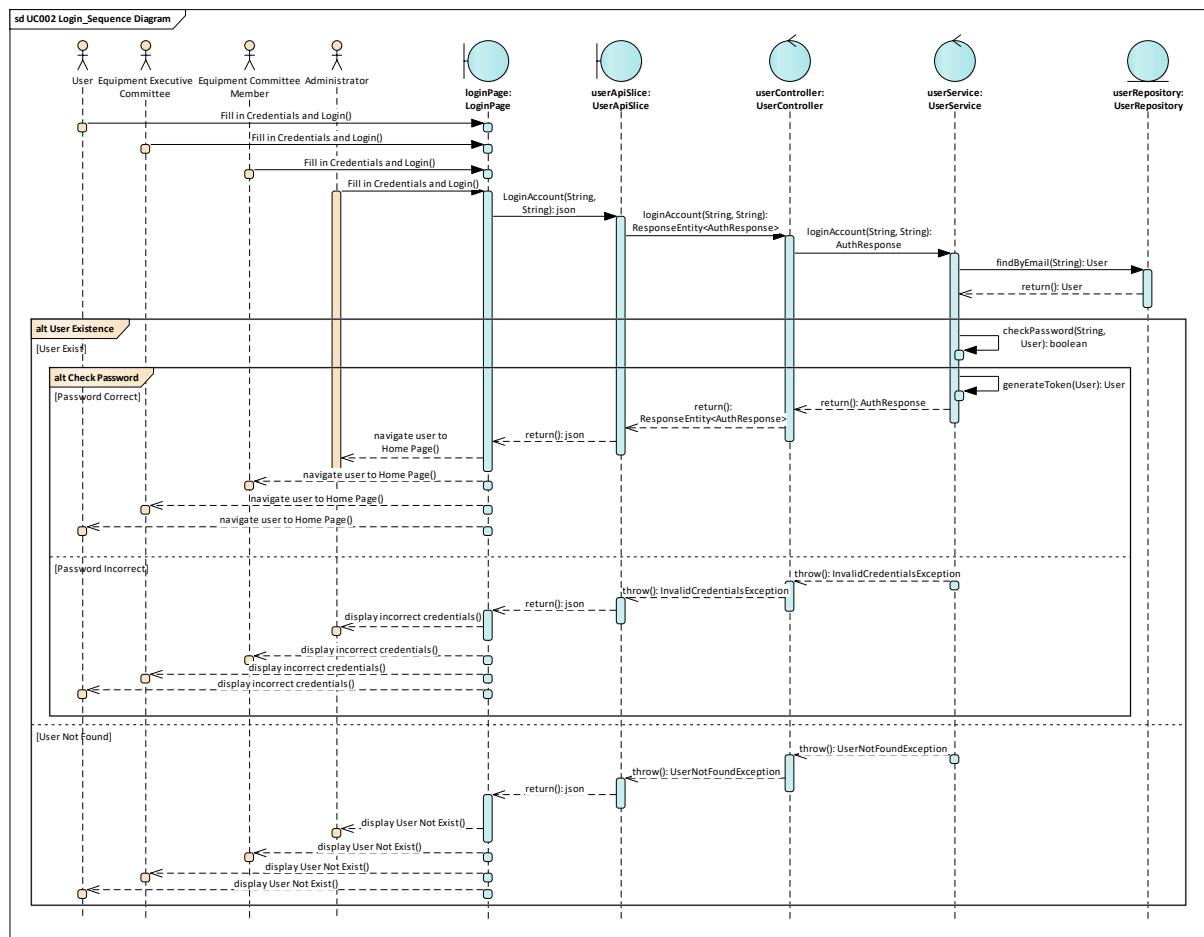


Figure 4.40 Sequence Diagram for UC002 Login

4.8.2.1.3 UC003: Reset Password

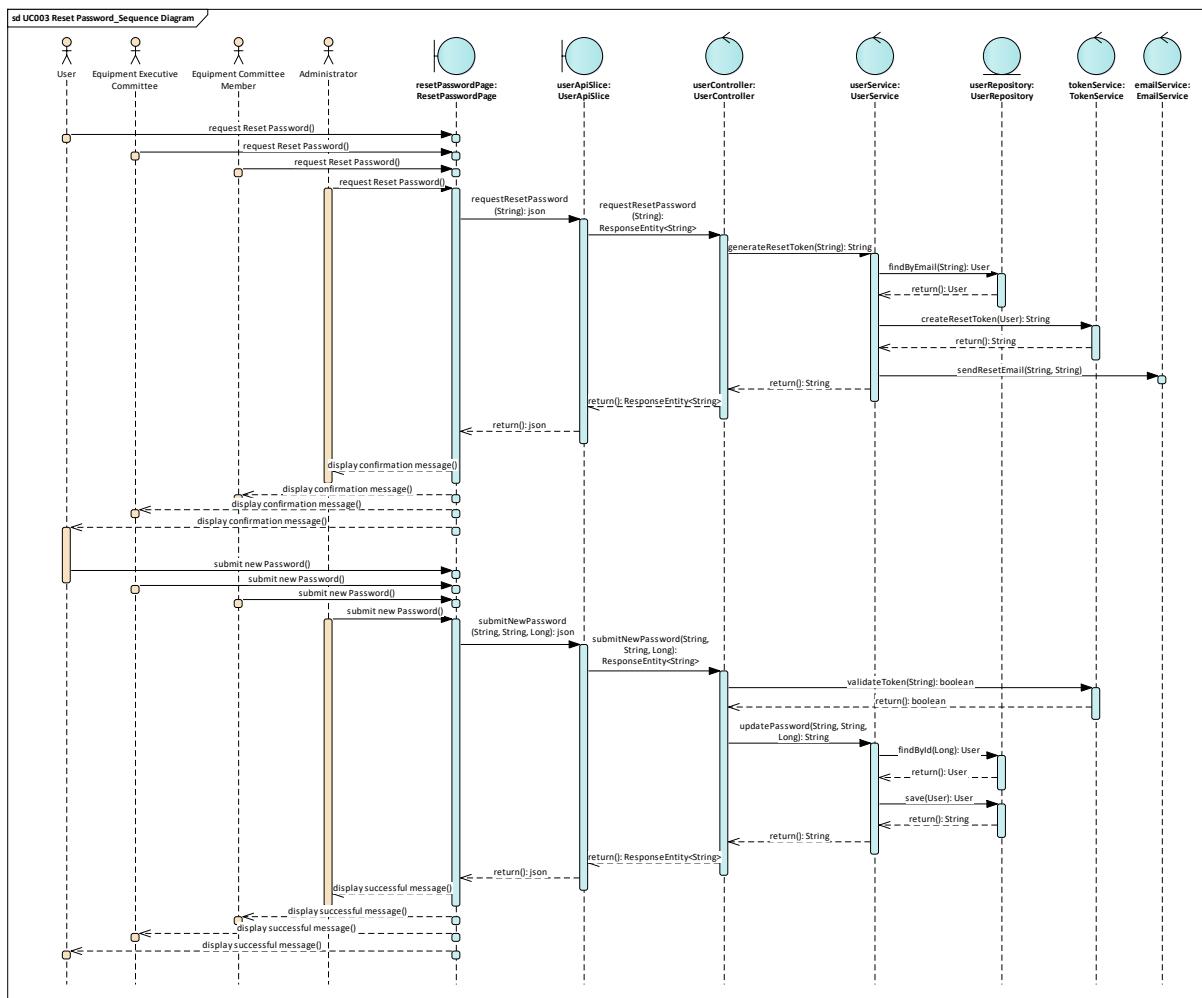


Figure 4.41 Activity Diagram for UC003 Reset Password

4.8.2.2 Equipment Request and Rental Module

4.8.2.2.1 UC004: View Equipment List

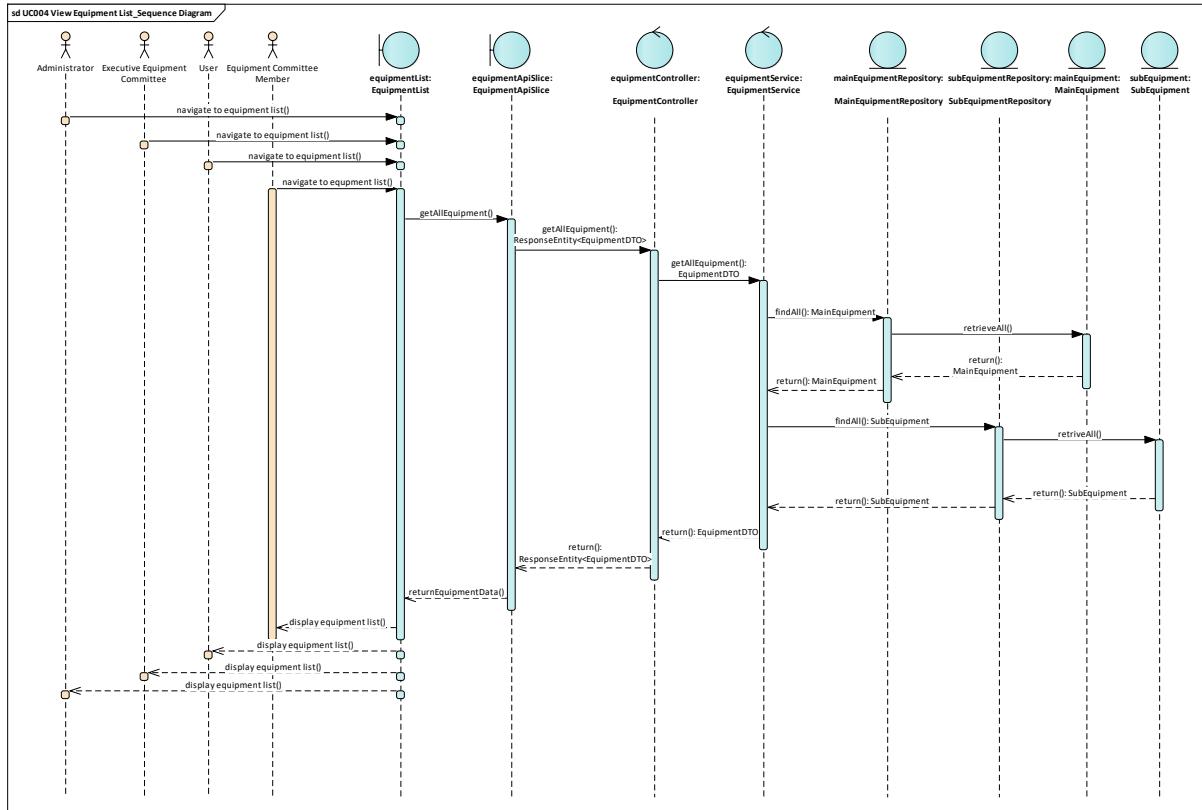


Figure 4.42 Activity Diagram for UC004 View Equipment List

4.8.2.2.2 UC005: Rent Equipment

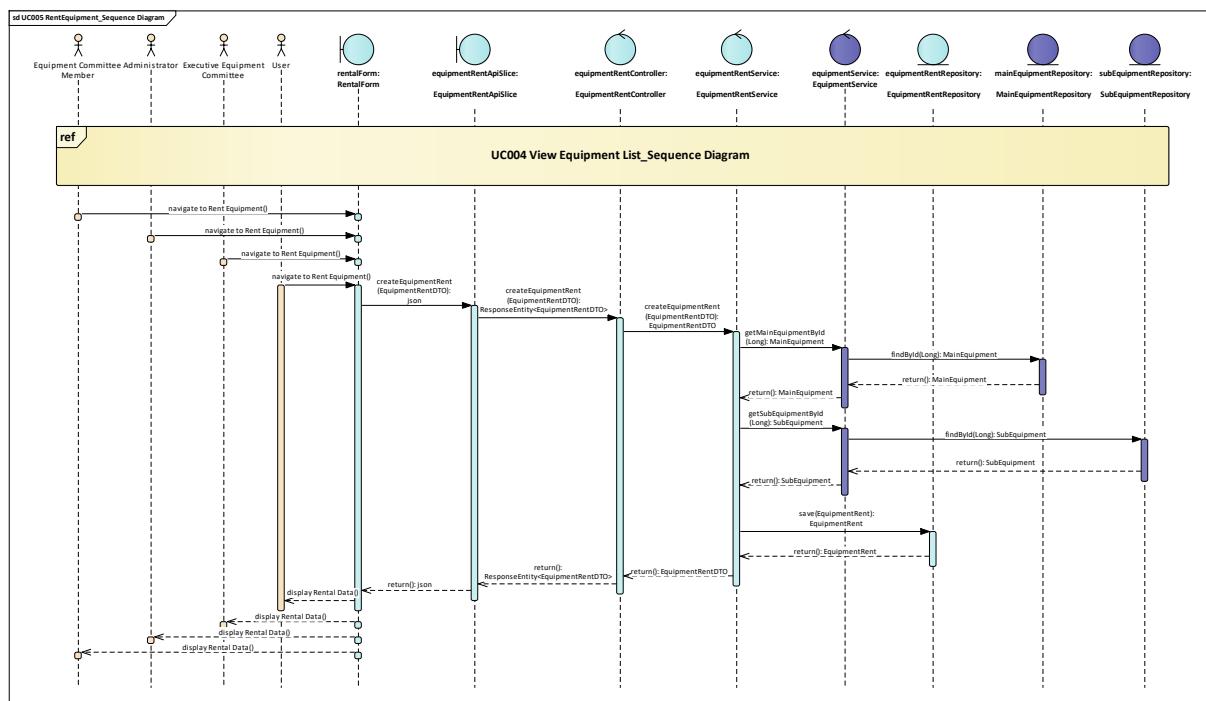


Figure 4.43 Sequence Diagram for UC005 Rent Equipment

4.8.2.2.3 UC006: Return Equipment

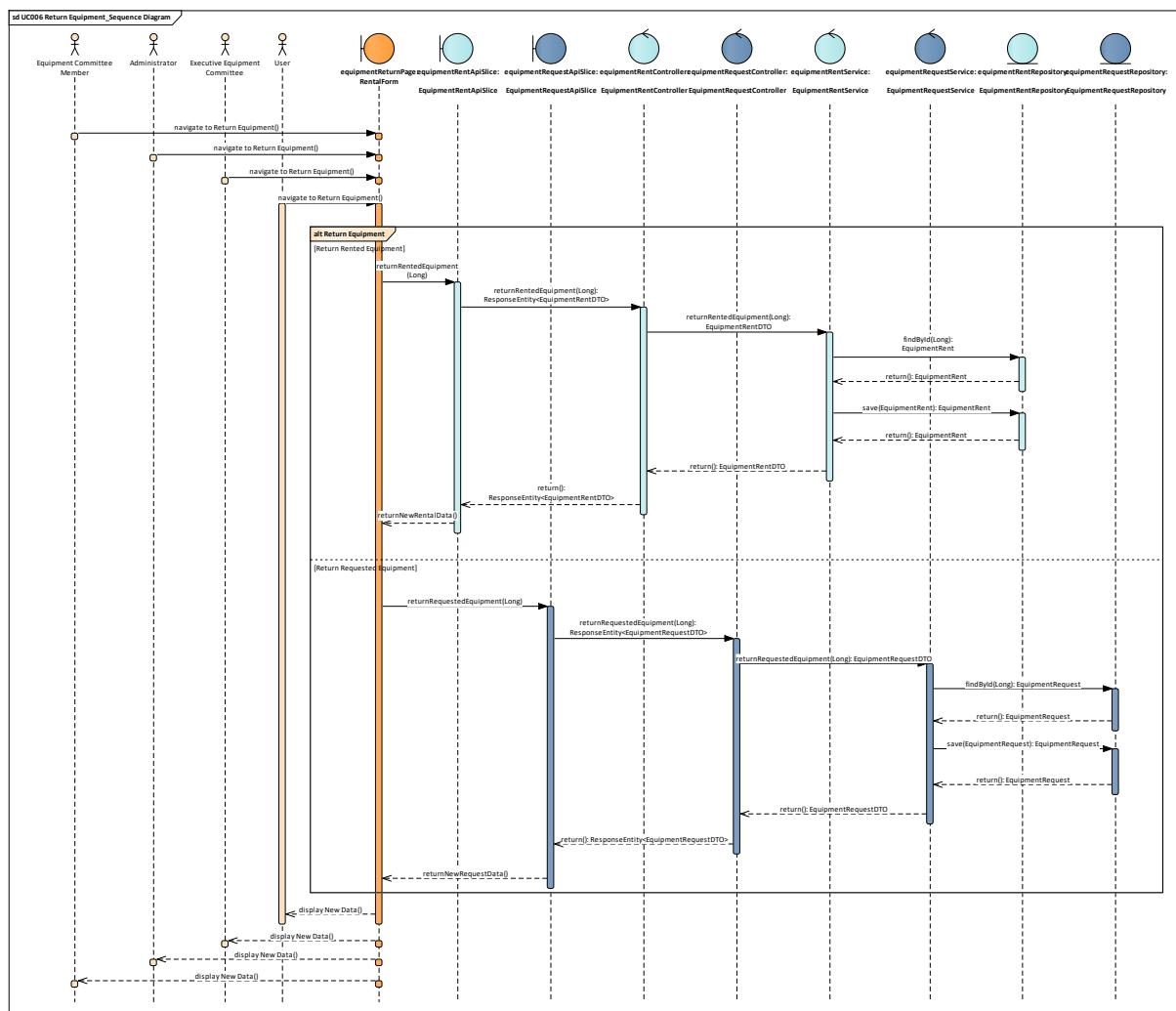


Figure 4.44 Sequence Diagram for UC006 Return Equipment

4.8.2.2.4 UC007: Make Payment

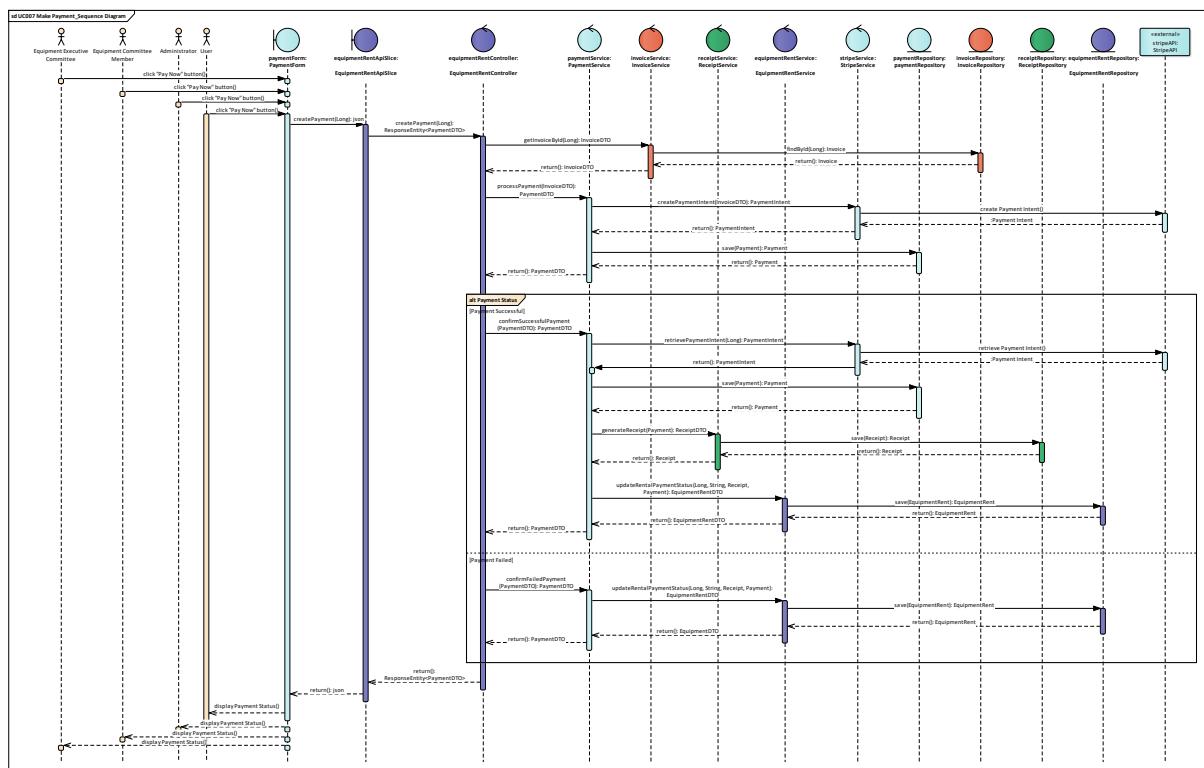


Figure 4.45 Sequence Diagram for UC007 Make Payment

4.8.2.2.5 UC008: View Generated Invoice and Receipt

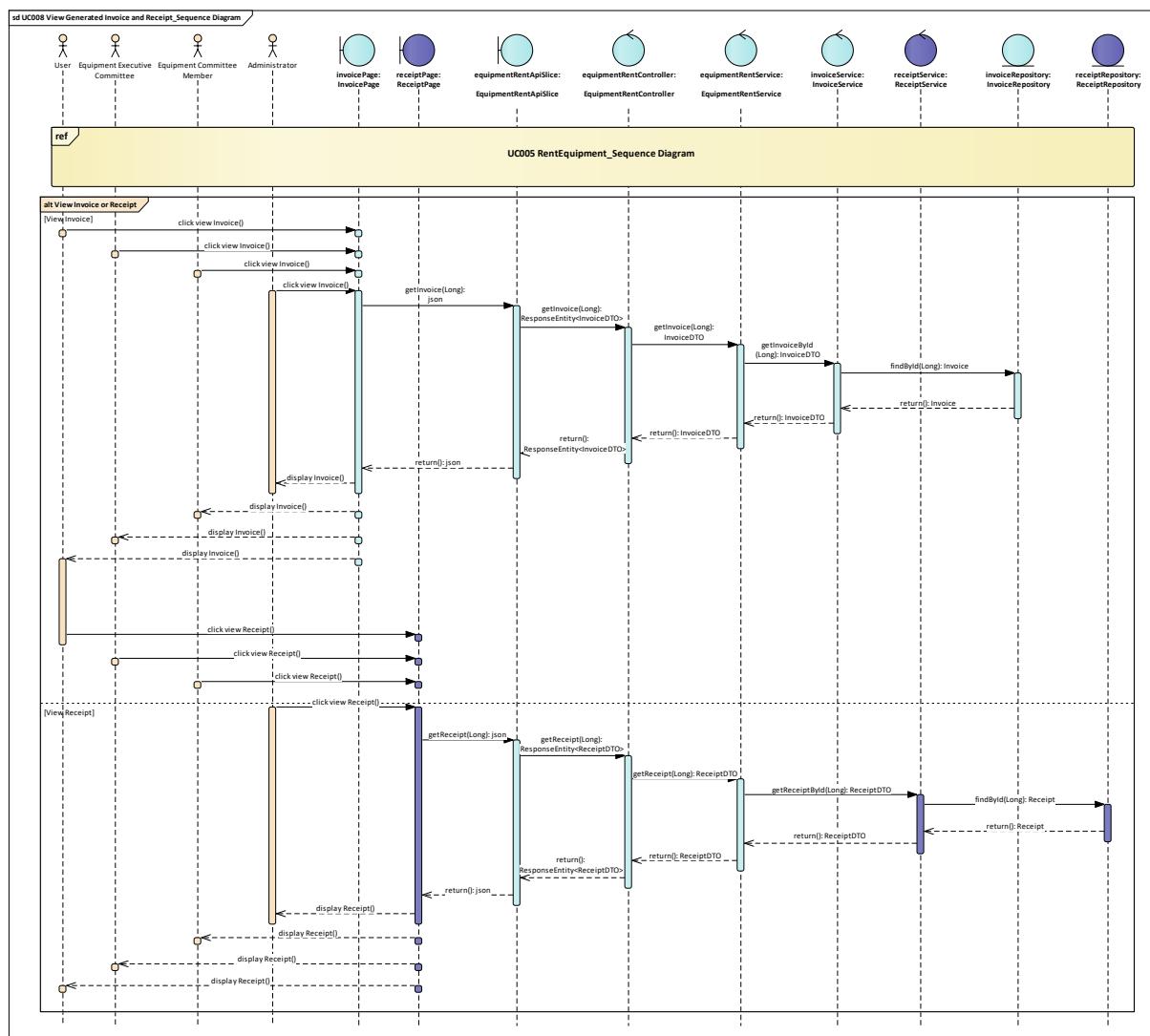


Figure 4.46 Sequence Diagram for UC008 View Generated Invoice and Receipt

4.8.2.2.6 UC009: Request Equipment

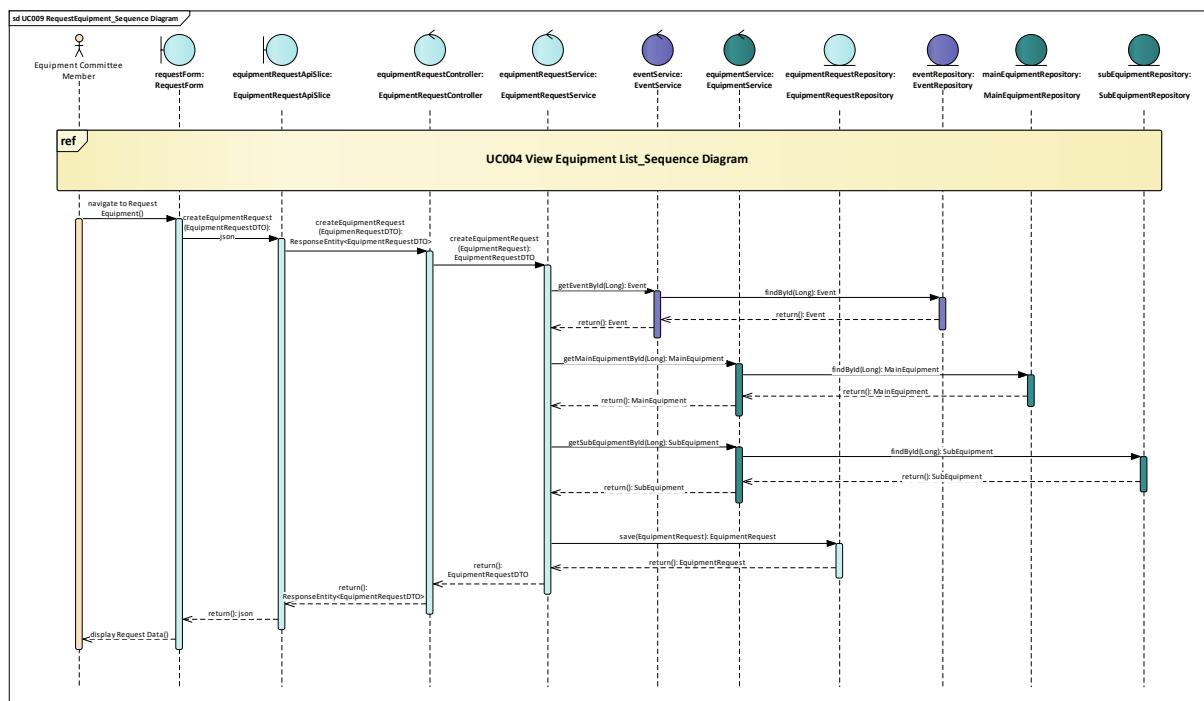


Figure 4.47 Sequence Diagram for UC009 Request Equipment

4.8.2.3 Equipment Request and Rental Module

4.8.2.3.1 UC010: Monitor Equipment

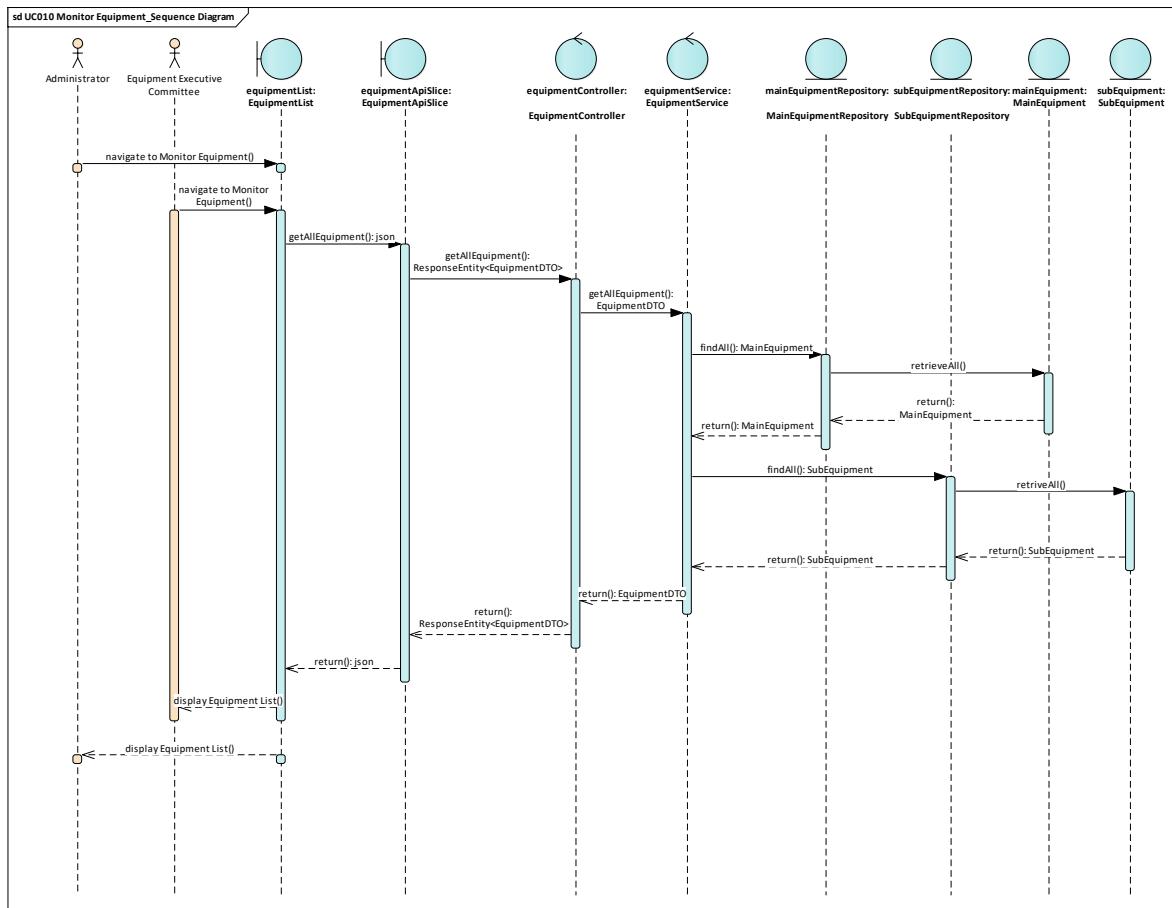


Figure 4.48 Sequence Diagram for UC010 Monitor Equipment

4.8.2.3.2 UC011: Update Equipment

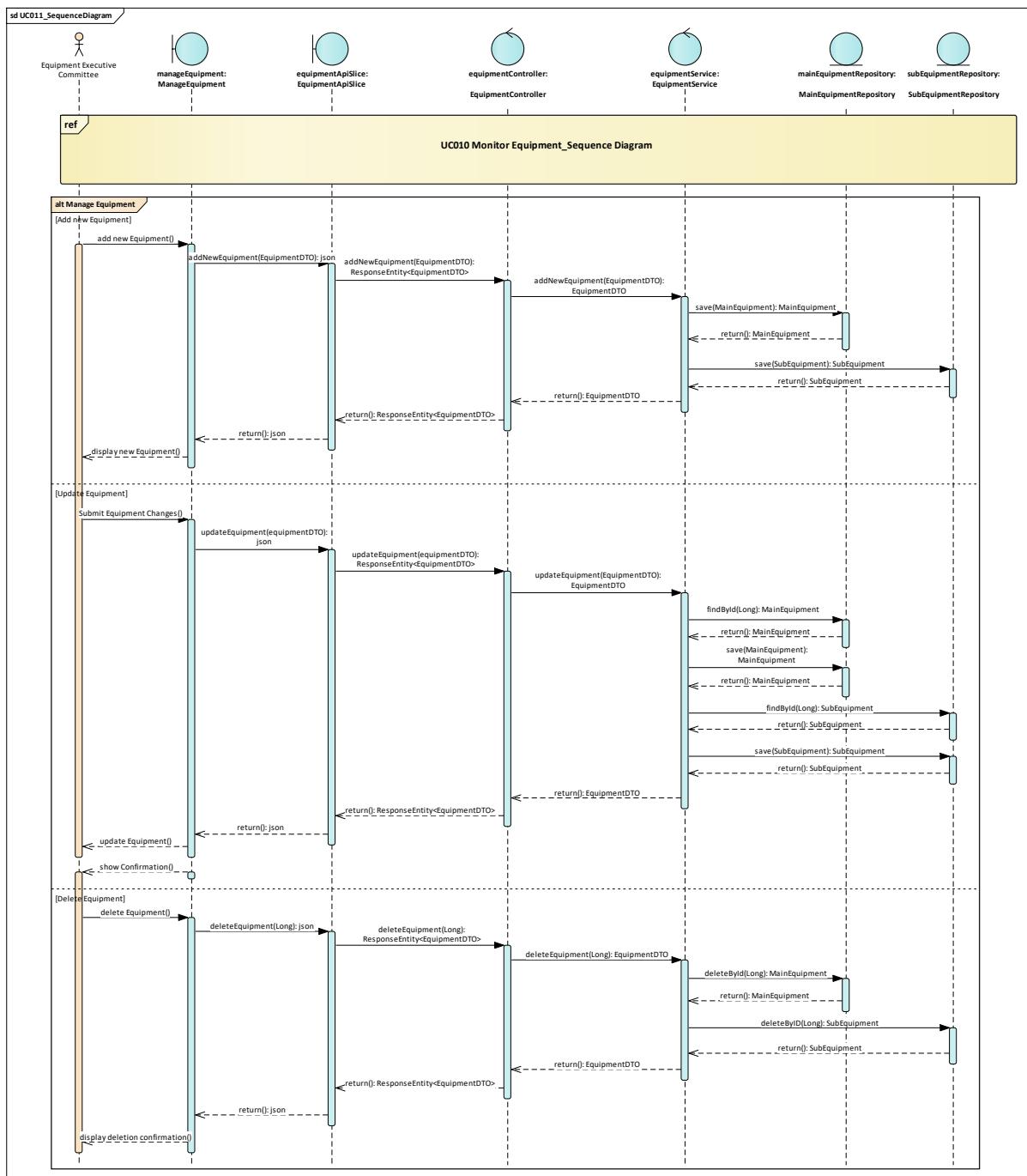


Figure 4.49 Sequence Diagram for UC011 Update Equipment

4.8.2.3.3 UC012: Manage Equipment Requests and Rentals

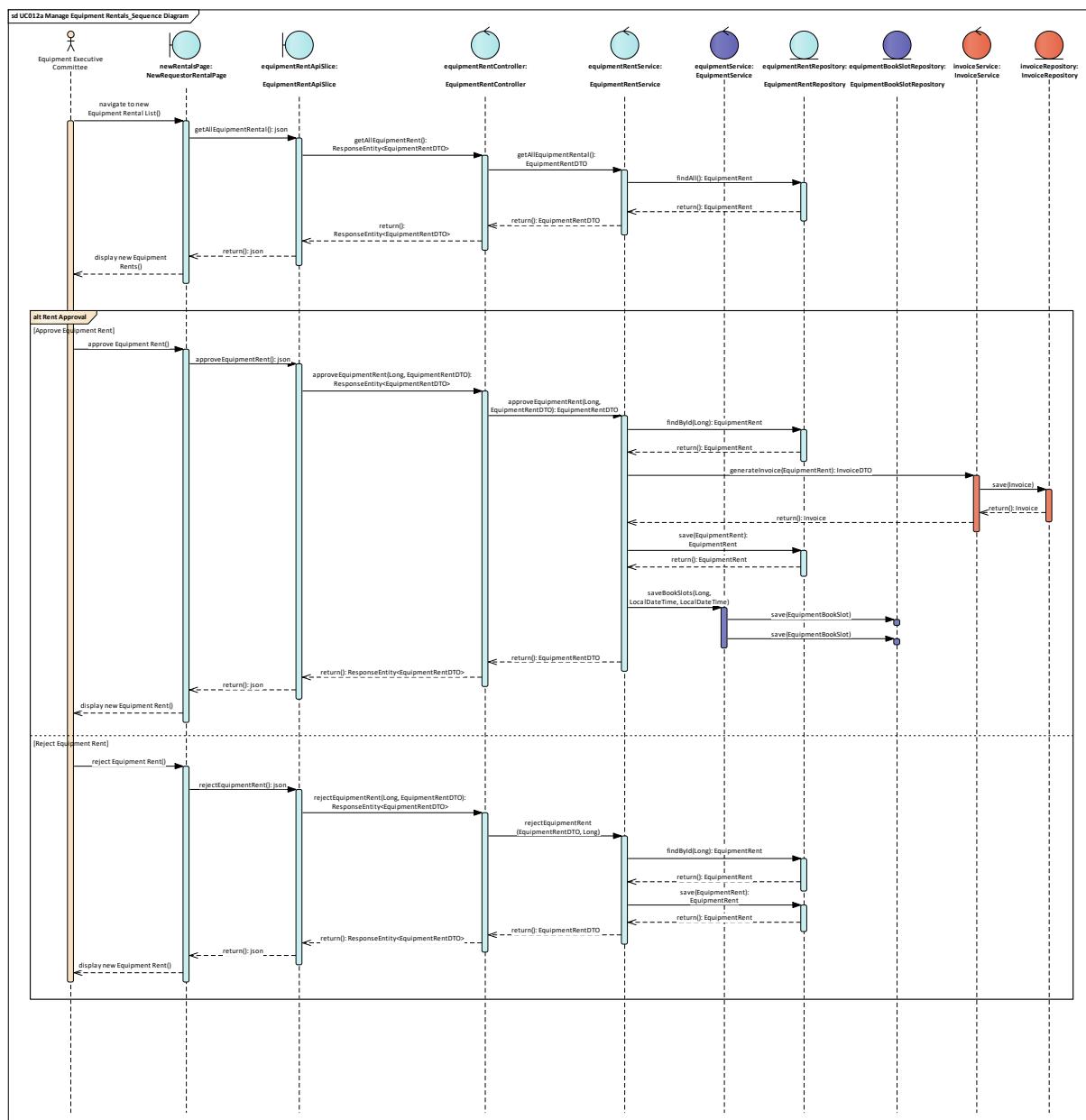


Figure 4.50 Sequence Diagram for UC012a Manage Equipment Requests and Rentals

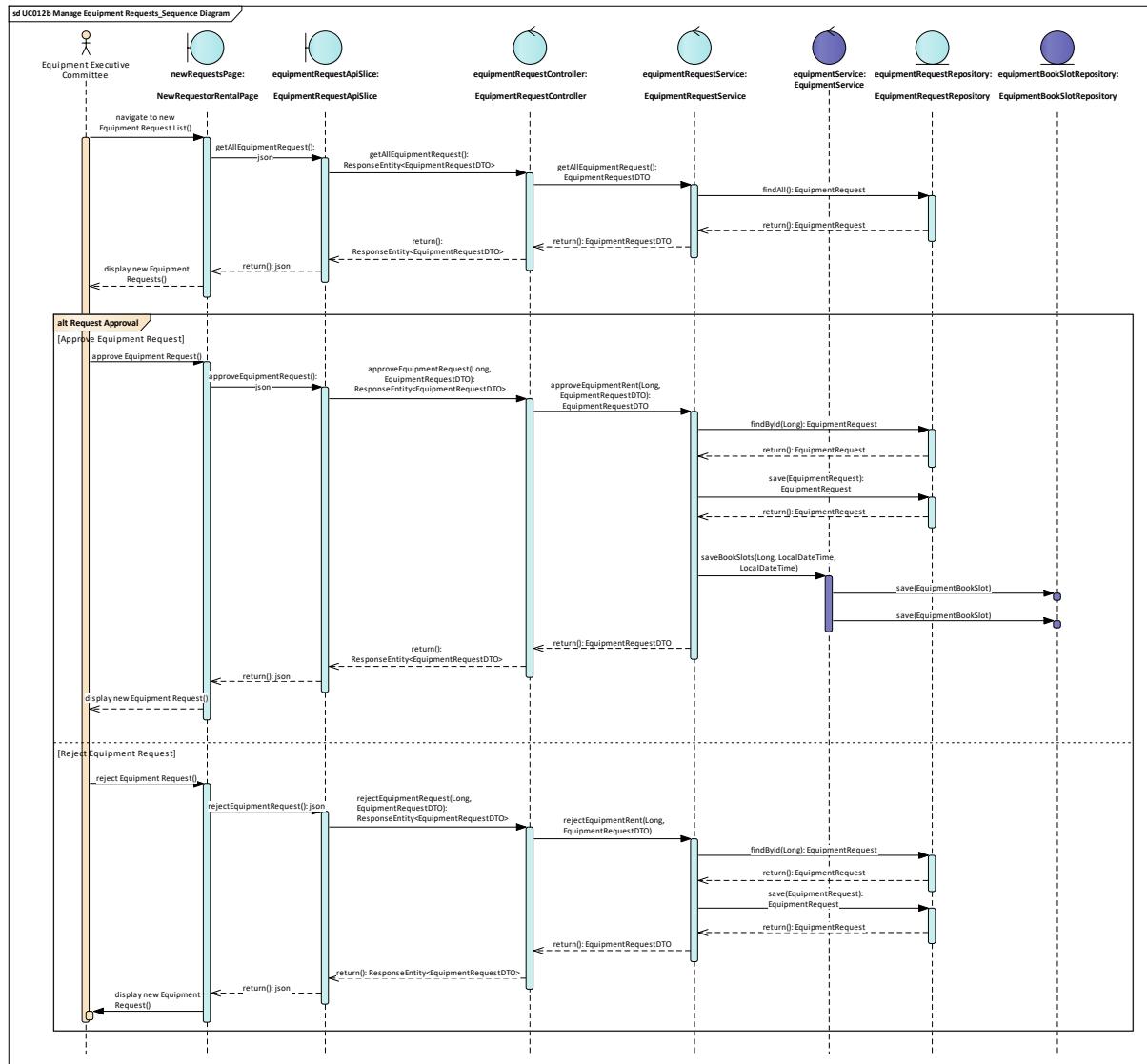


Figure 4.51 Sequence Diagram for UC012b Manage Equipment Requests and Rentals

4.8.2.3.4 UC013: Confirm Equipment Return

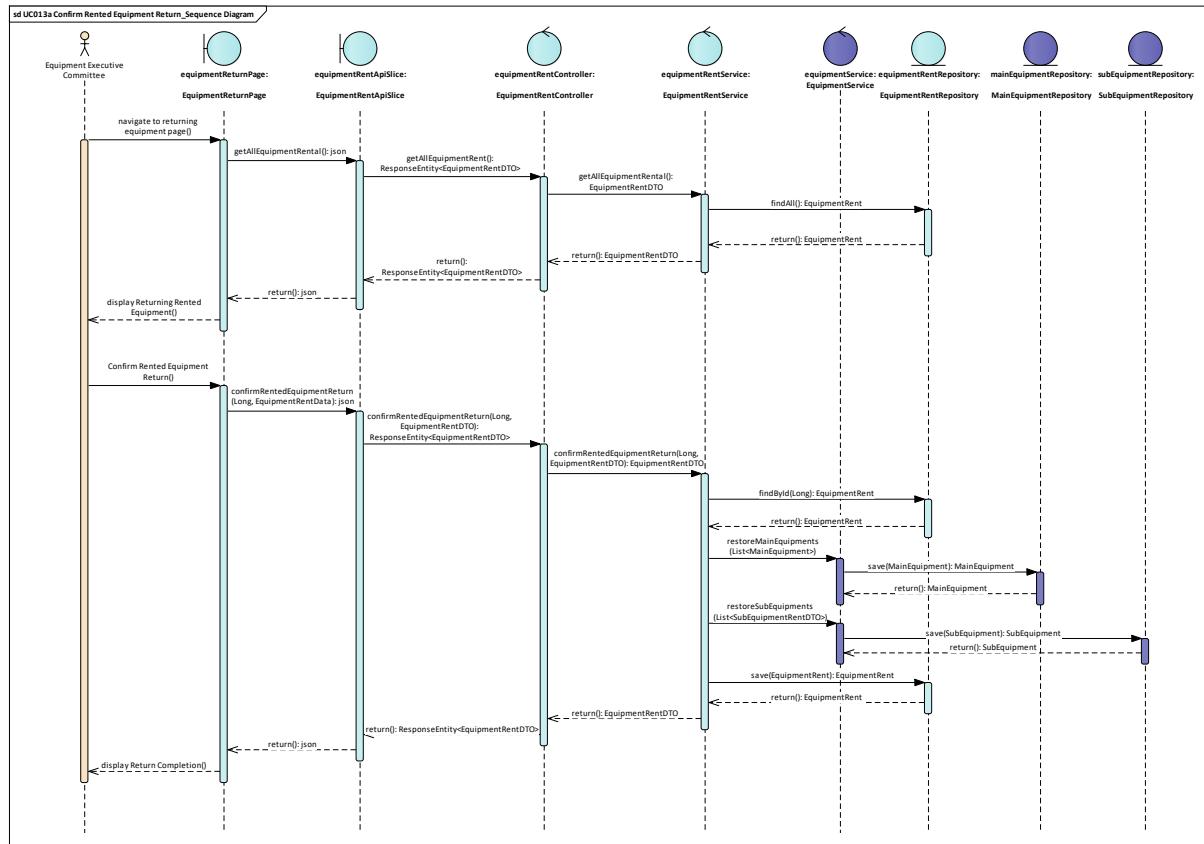


Figure 4.52 Sequence Diagram for UC013a Confirm Equipment Return

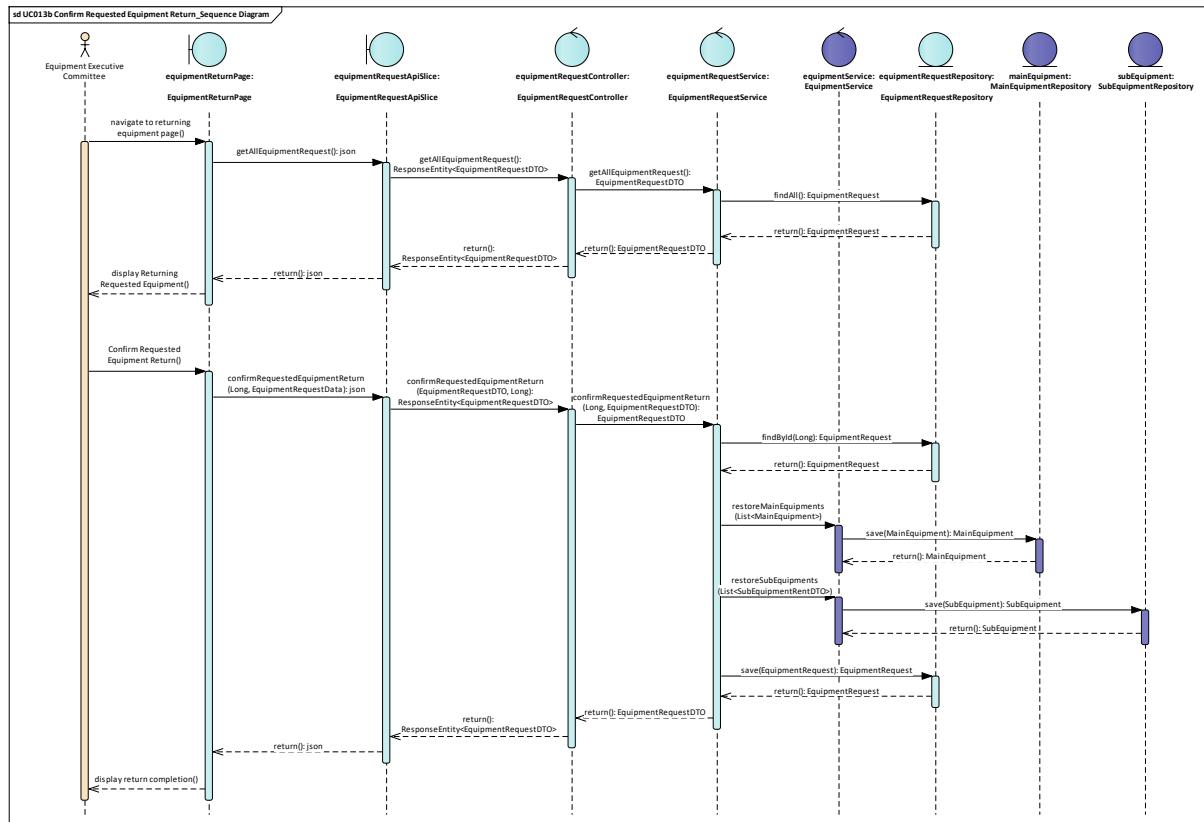


Figure 4.53 Sequence Diagram for UC013b Confirm Equipment Return

4.8.2.4 Role and Event Management Module

4.8.2.4.1 UC014: Manage Members Roles

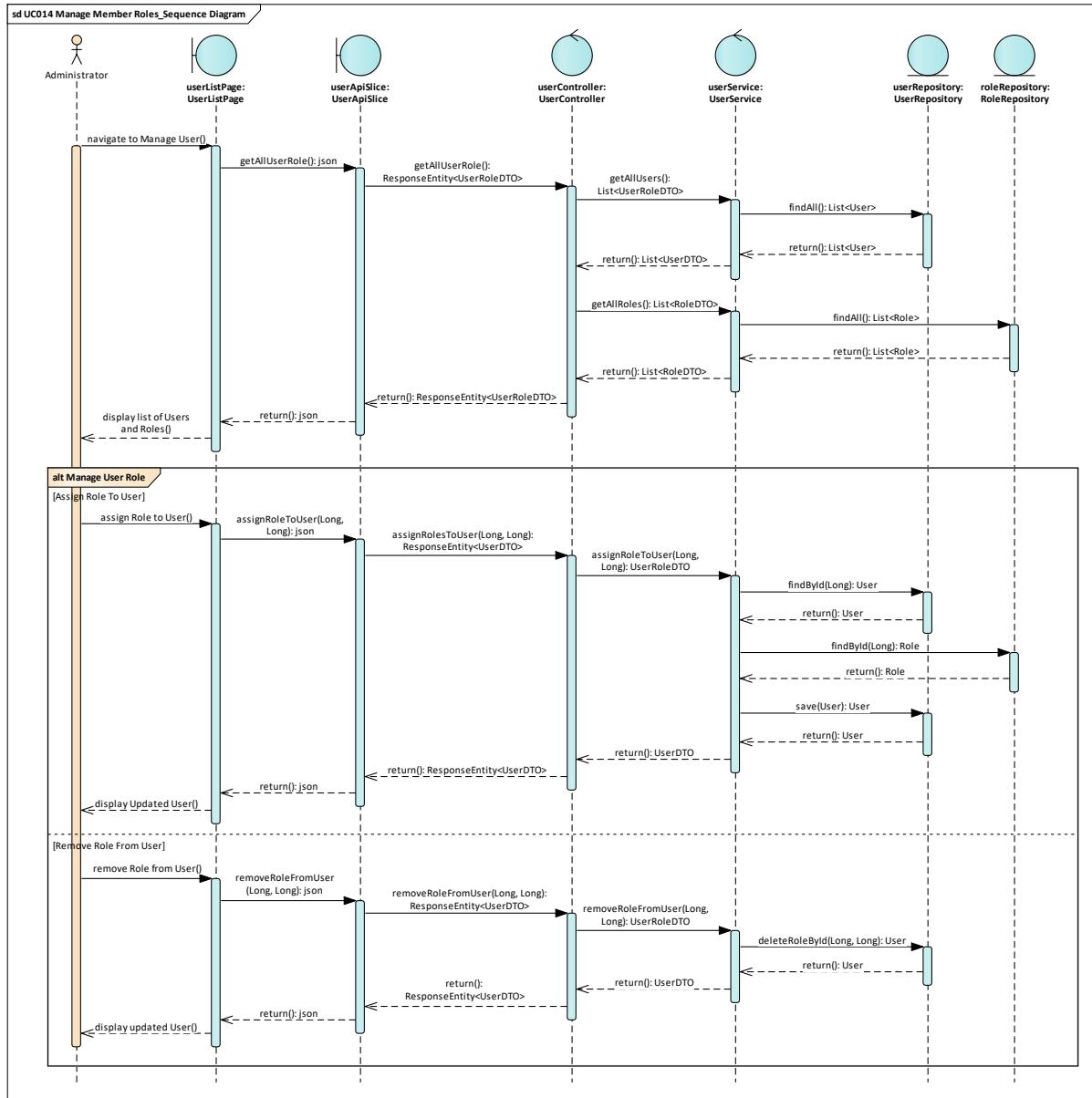


Figure 4.54 Sequence Diagram for UC014 Manage Member Roles

4.8.2.4.2 UC015: Manage Event

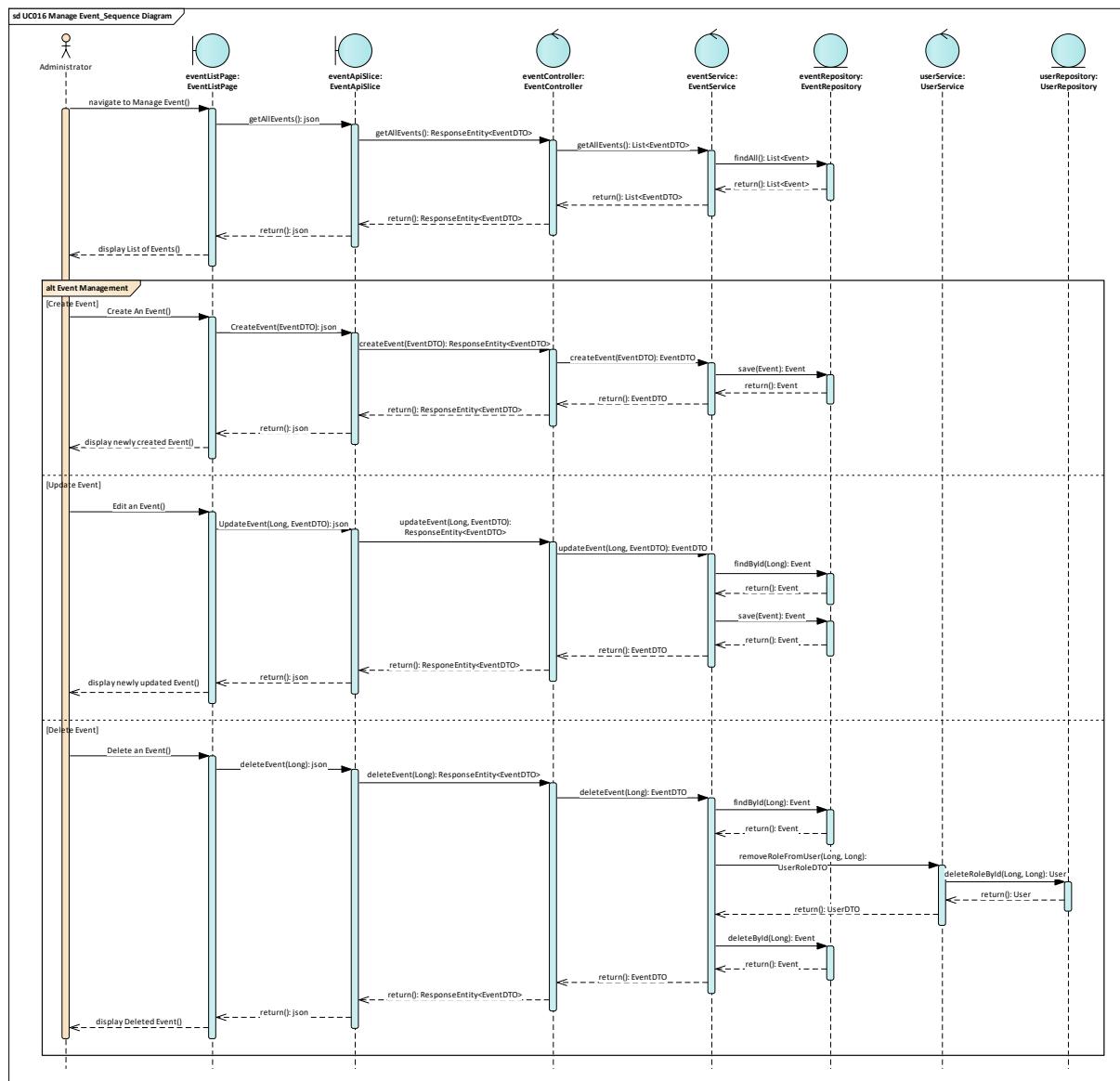


Figure 4.55 Sequence Diagram for UC015 Manage Event

4.9 State Dynamic Viewpoint

4.9.1 Design Concerns

In this section, system behaviour and states of the system are given via the state transition diagram. Designers, developers, and testers are informed about dynamic view of the system which includes state definitions, transition conditions, event handling mechanisms, and behavioural patterns that govern system operations.

The state dynamic perspective establishes the number of operational states system entities can be in during their lifecycle such as error states, intermediate processing states, final states, and initial states. For example, Equipment Request and Rental Module includes such states as Submitted, Approved, Rejected, Pending Return, and Returned as well as conditions and business rules to control transitions between the states.

The transition of states is regulated by certain conditions, business rules as well as validation criteria which should be met before a state transition can take place. These guard conditions including user permissions, data validation requirement, resources availability and business process constraints, are documented in the viewpoint. This will guarantee that state changes are made only when the right conditions are set and system integrity and adherence to business rule is maintained.

4.9.2 Design View (State Machine Diagram)

4.9.2.1 State Diagram EquipmentRequest class

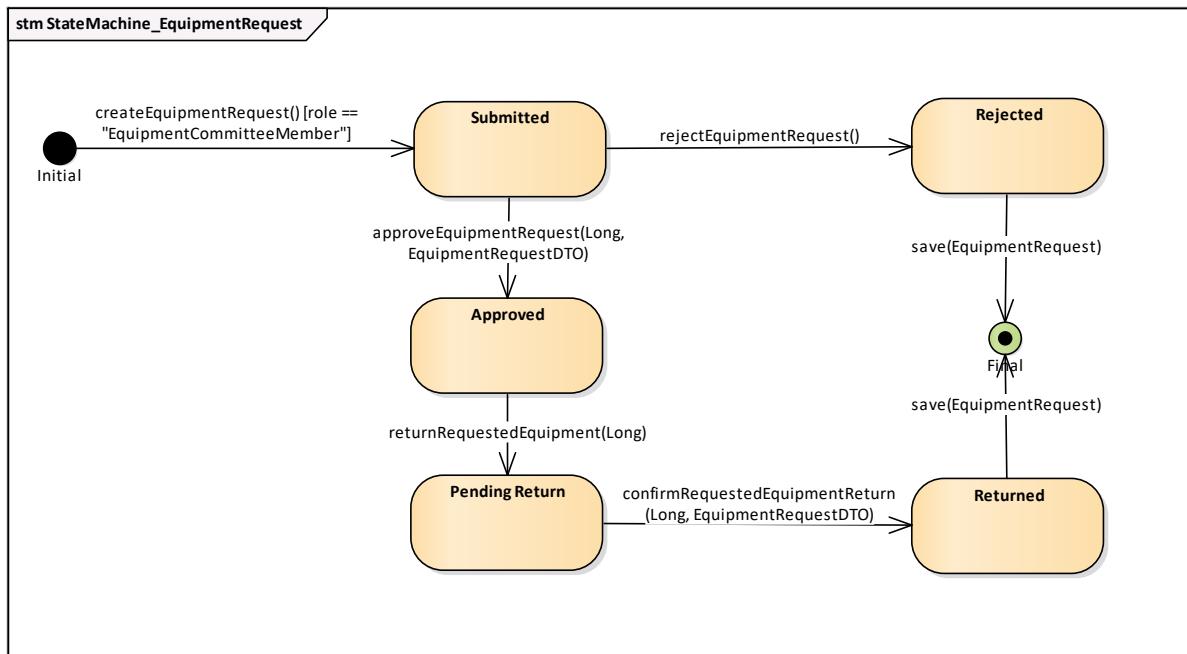


Figure 4.56 State Diagram for EquipmentRequest class

4.9.2.2 State Diagram EquipmentRent class

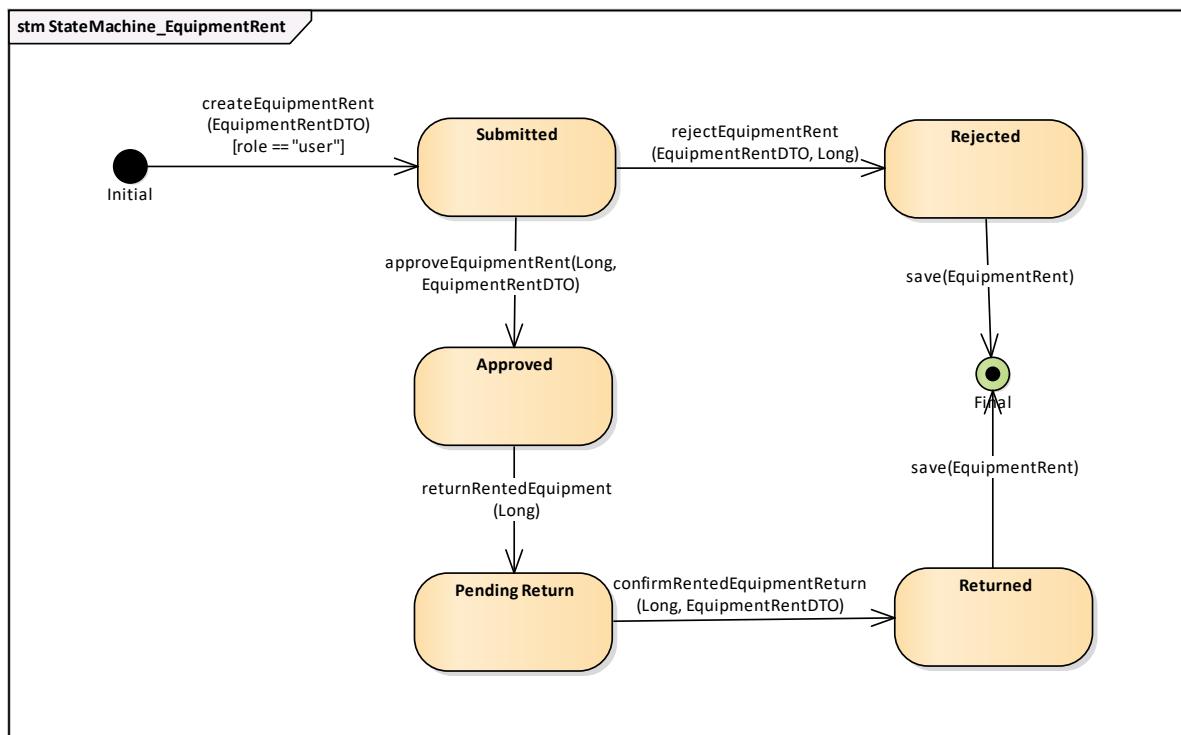


Figure 4.57 State Diagram for EquipmentRent class

4.9.2.3 State Diagram Payment class

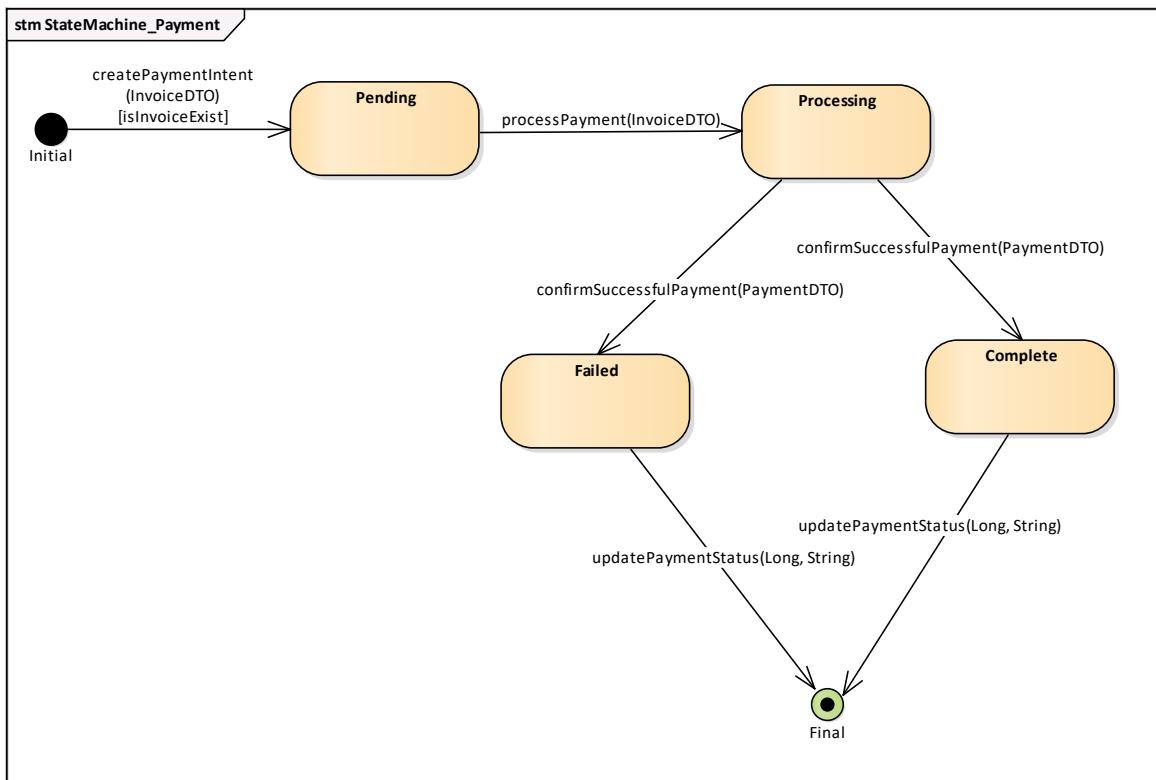


Figure 4.58 State Diagram for Payment class

4.9.2.4 State Diagram MainEquipment class

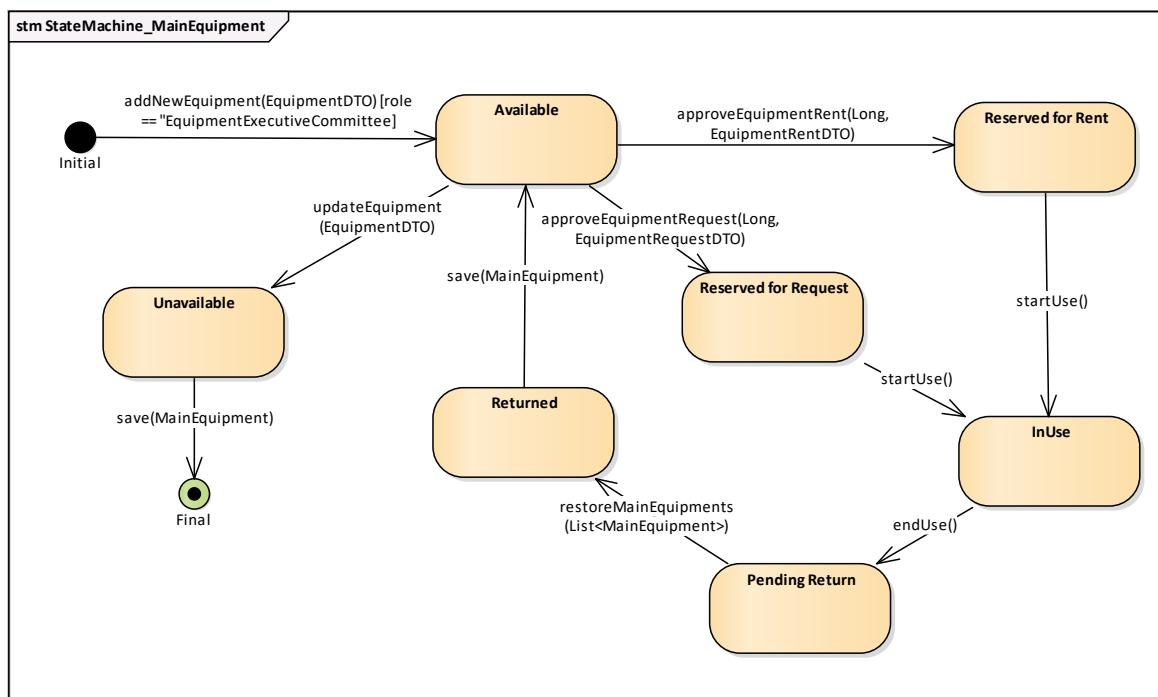


Figure 4.59 State Diagram for MainEquipment class

4.10 Algorithm Viewpoint

4.10.1 Design Concerns

The Algorithm perspective gives information required by algorithm analysts, programmers, and anyone concerned with time-space performance and the details of how to process information before it gets implemented, and to contribute to formulating unit test plans. It deals with the inner workings of the system components, notably in the implementation of methods and functions that achieve the necessary functionality.

It is a perspective on the most important part of the processing logic in each service or class operation, covering the control flow, data handling, validation constraints, condition branches, and loops. As an example, the `createEquipmentRent()` method in the `EquipmentRentService` performs the following steps: mapping a DTO to the entity, establishing relationships between sub-equipment and a rental, setting equipment availability on or off and storing the data in the database. Analogously, the `restoreSubEquipments()` method updates the arithmetical amounts on quantity fields to correspond with returned equipment. These algorithms provide uniformity to the data and application of business rules specified within the use cases. The algorithm perspective helps to promote maintainability, extensibility, and unit testing, because this logic is captured explicitly in the code.

4.10.2 Design View (Activity Diagram)

4.10.2.1 Authentication and Authorization Module

4.10.2.1.1 UC001: Register An Account

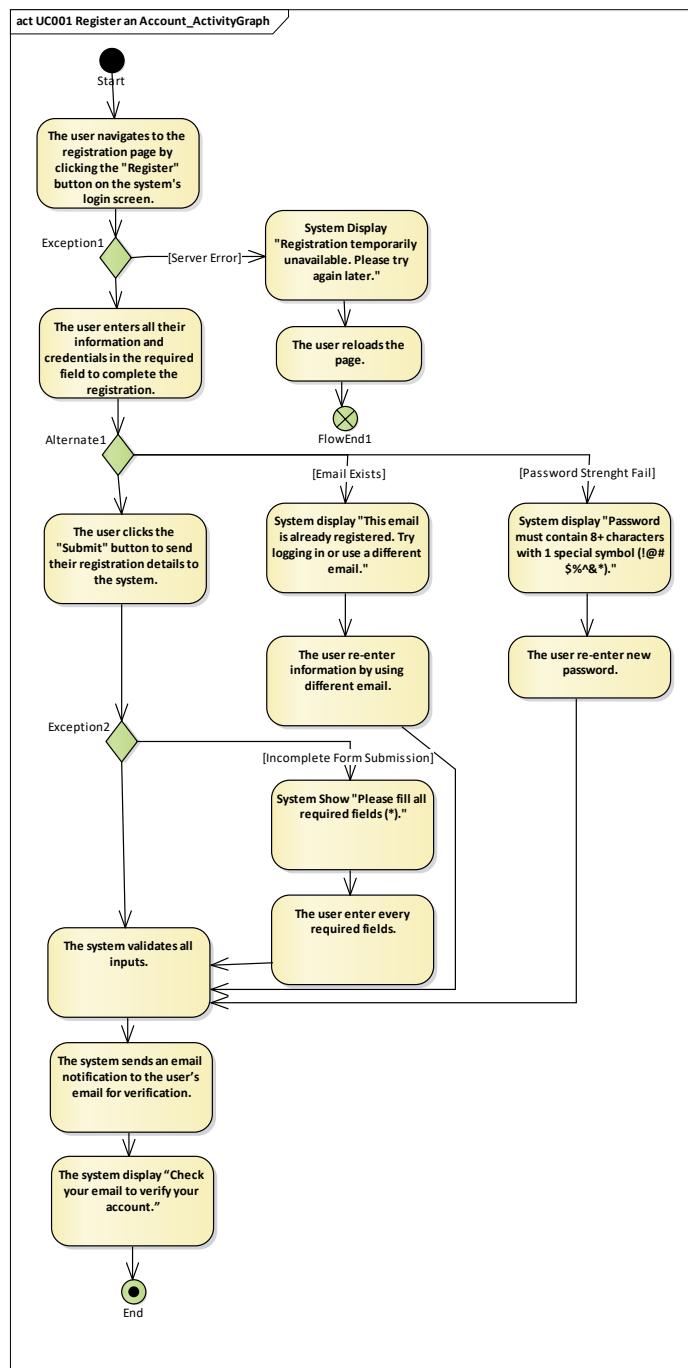


Figure 4.60 Activity Diagram for UC001 Register an Account

4.10.2.1.2 UC002: Login

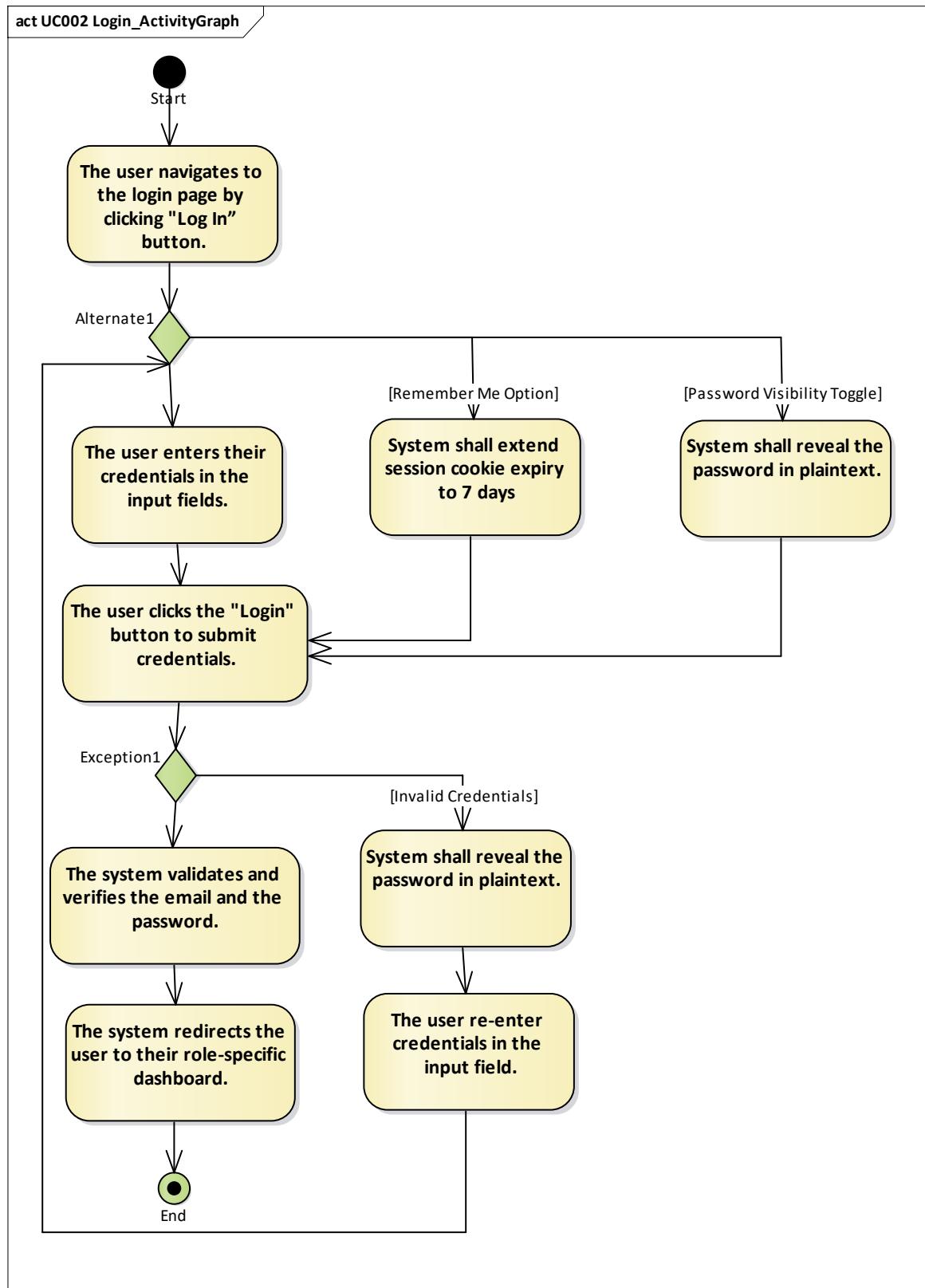


Figure 4.61 Activity Diagram for UC002 Login

4.10.2.1.3 UC003: Reset Password

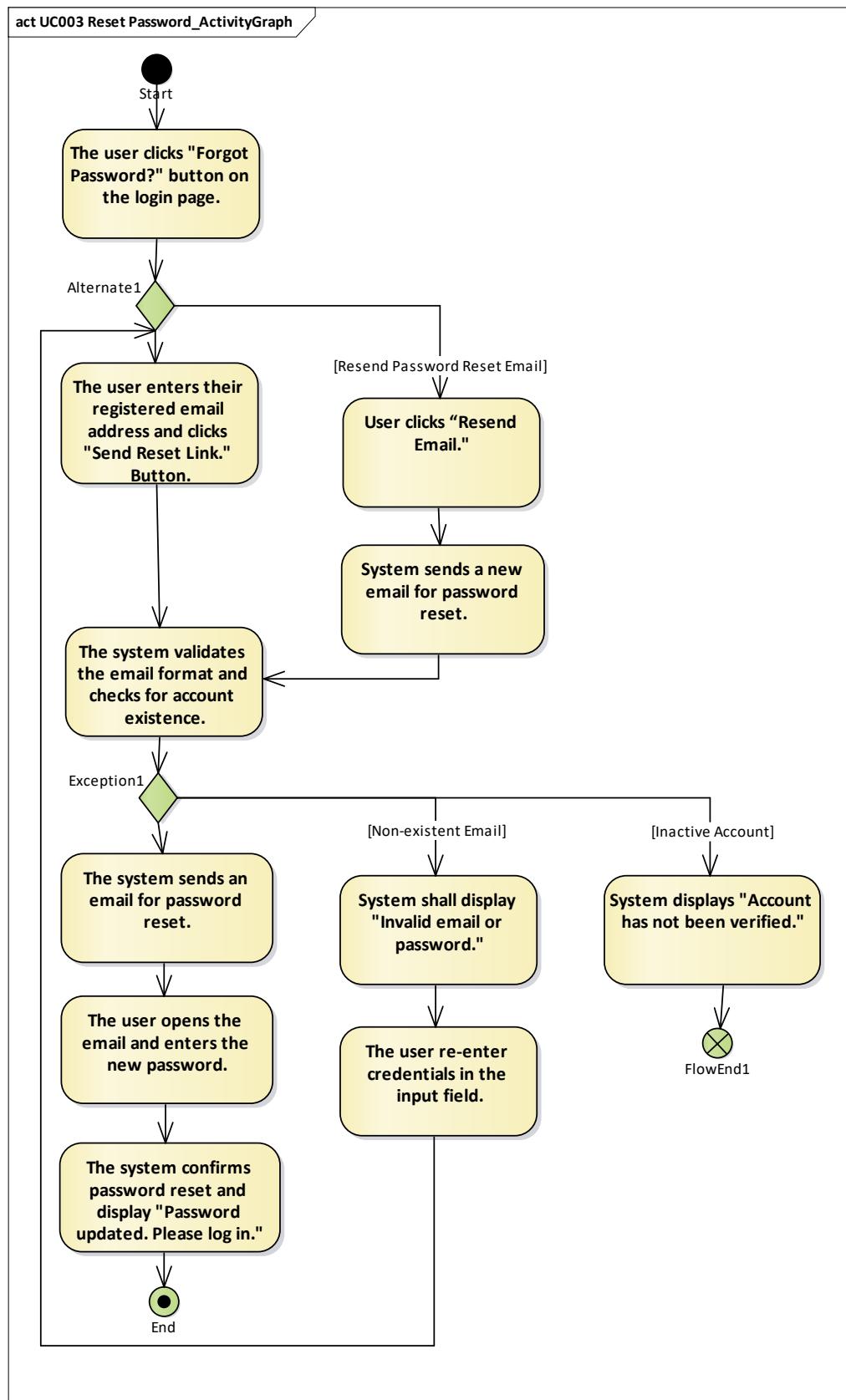


Figure 4.62 Activity Diagram for UC003 Reset Password

4.10.2.2 Equipment Request and Rental Module

4.10.2.2.1 UC004: View Equipment List

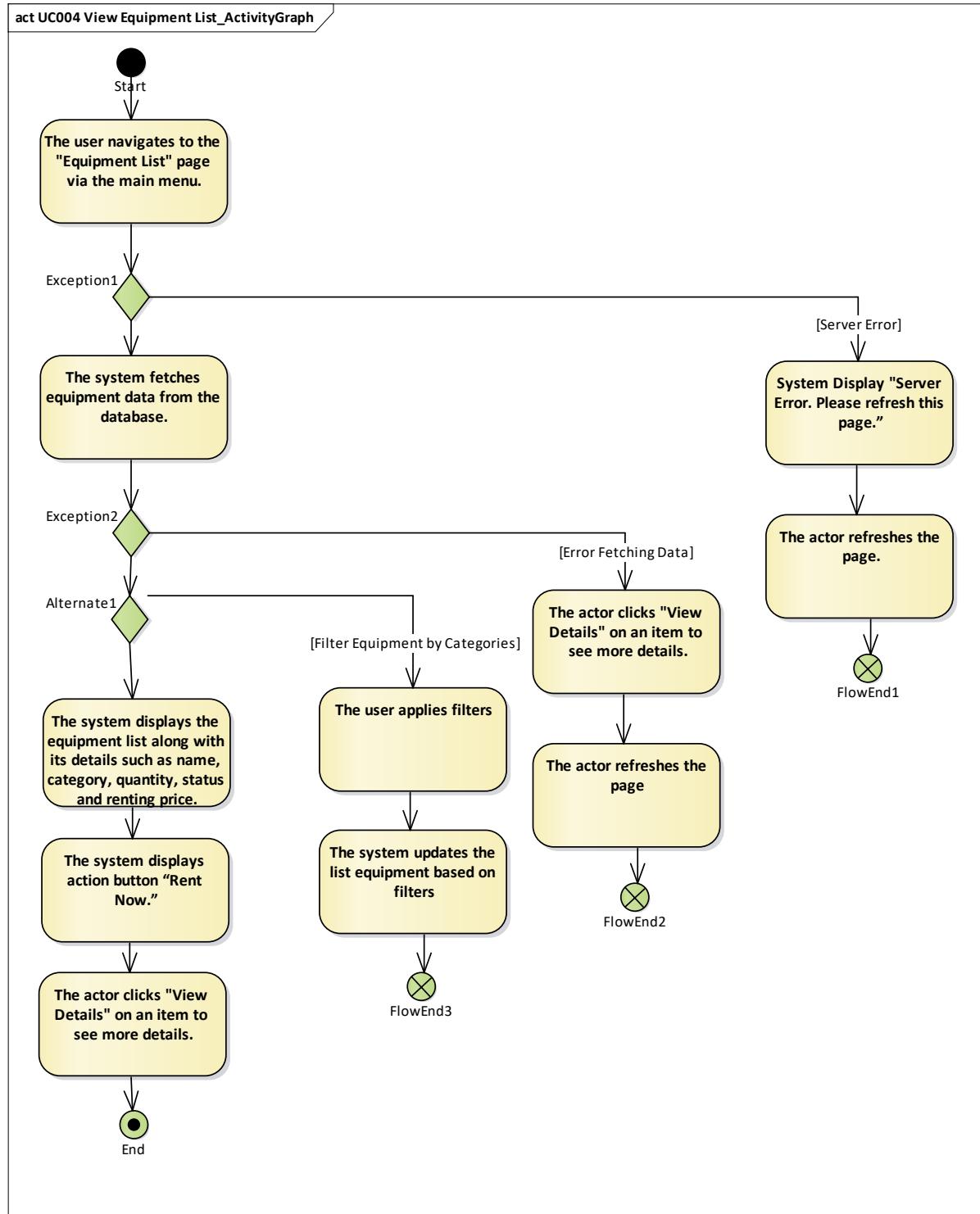


Figure 4.63 Activity Diagram for UC004 View Equipment List

4.10.2.2.2 UC005: Rent Equipment

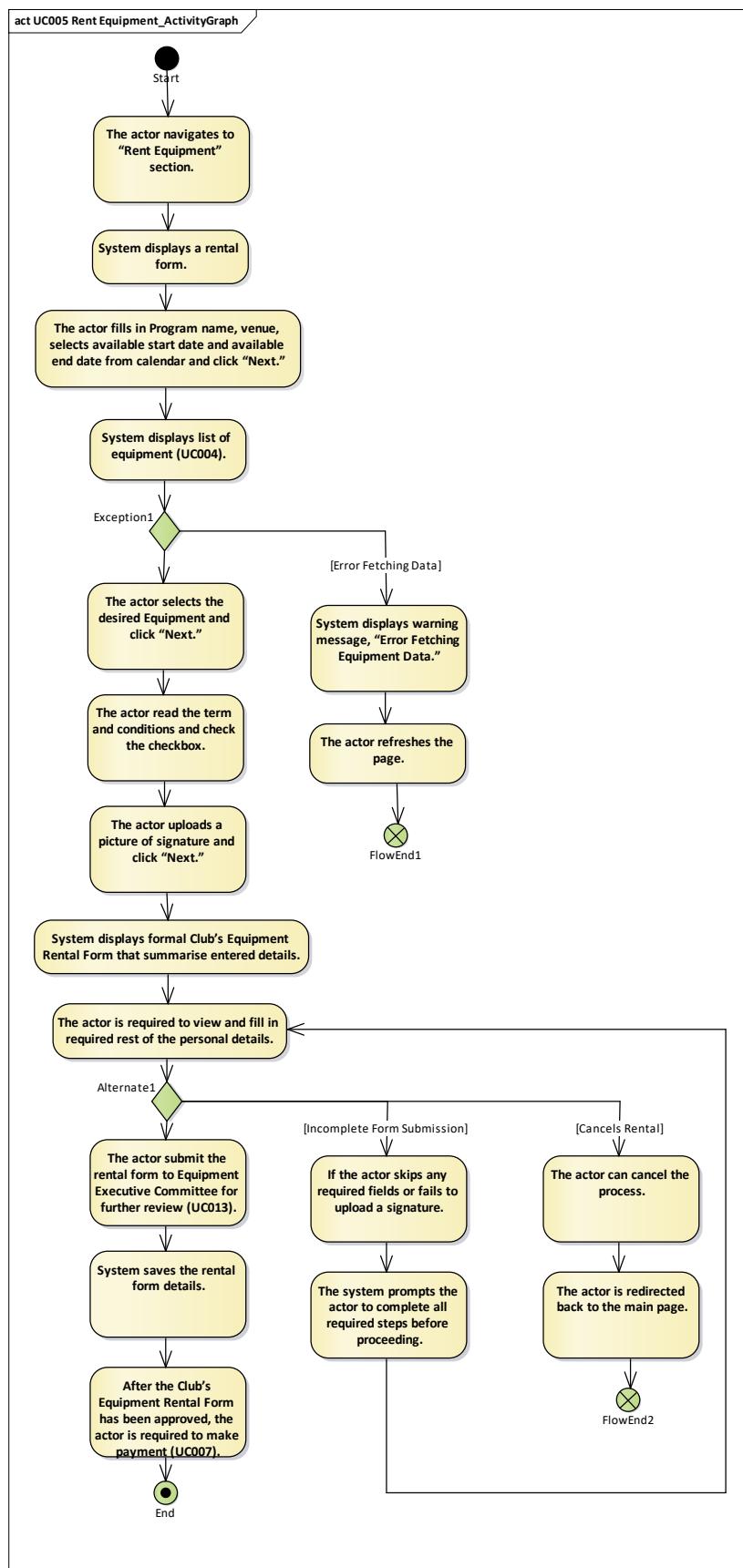


Figure 4.64 Activity Diagram for UC005 Rent Equipment

4.10.2.2.3 UC006: Return Equipment

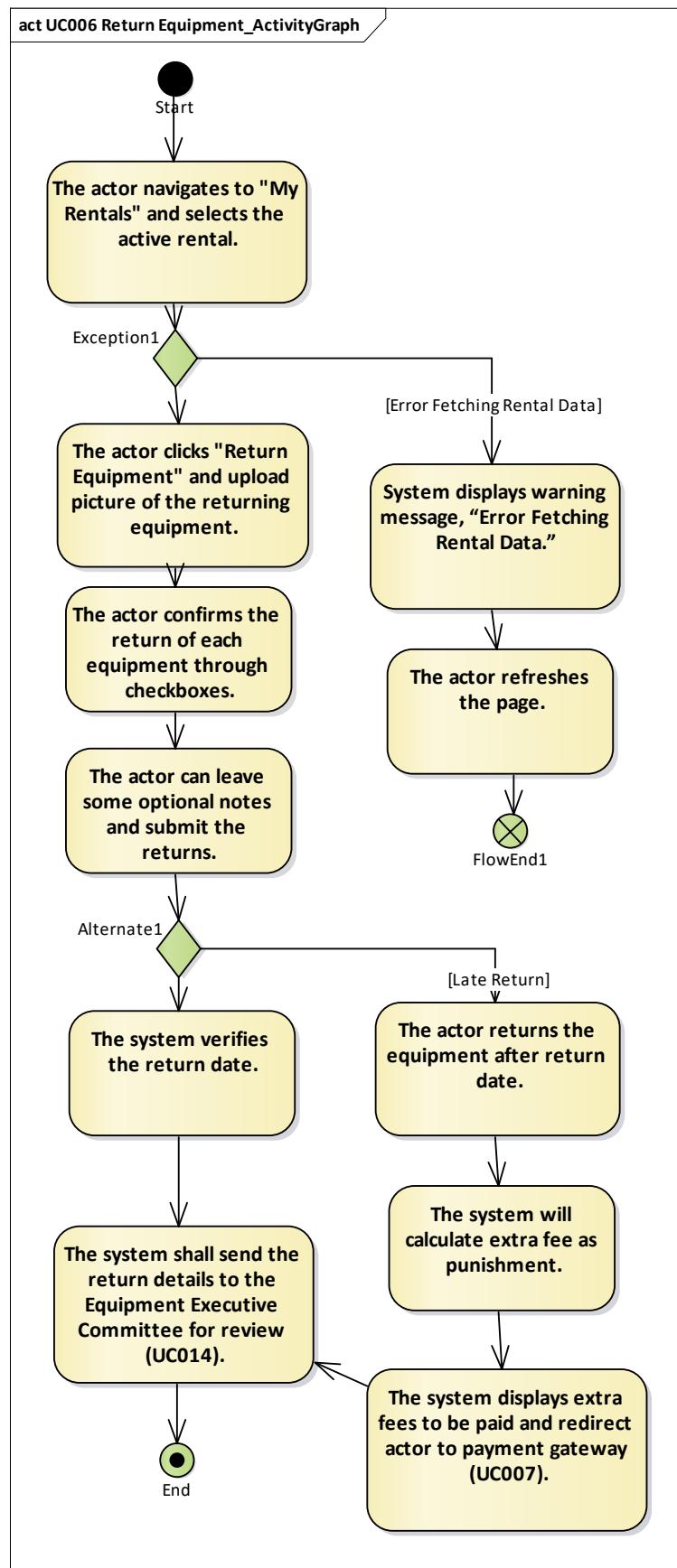


Figure 4.65 Activity Diagram for UC006 Return Equipment

4.10.2.2.4 UC007: Make Payment

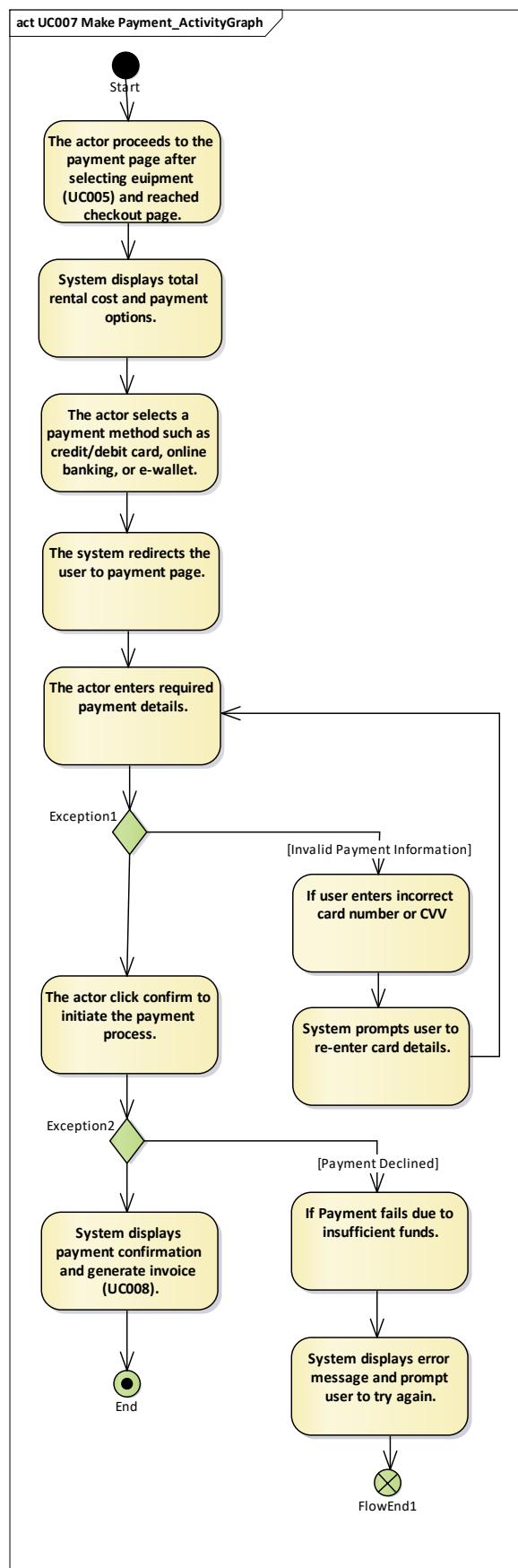


Figure 4.66 Activity Diagram for UC007 Make Payment

4.10.2.2.5 UC008: View Generated Invoice and Receipt

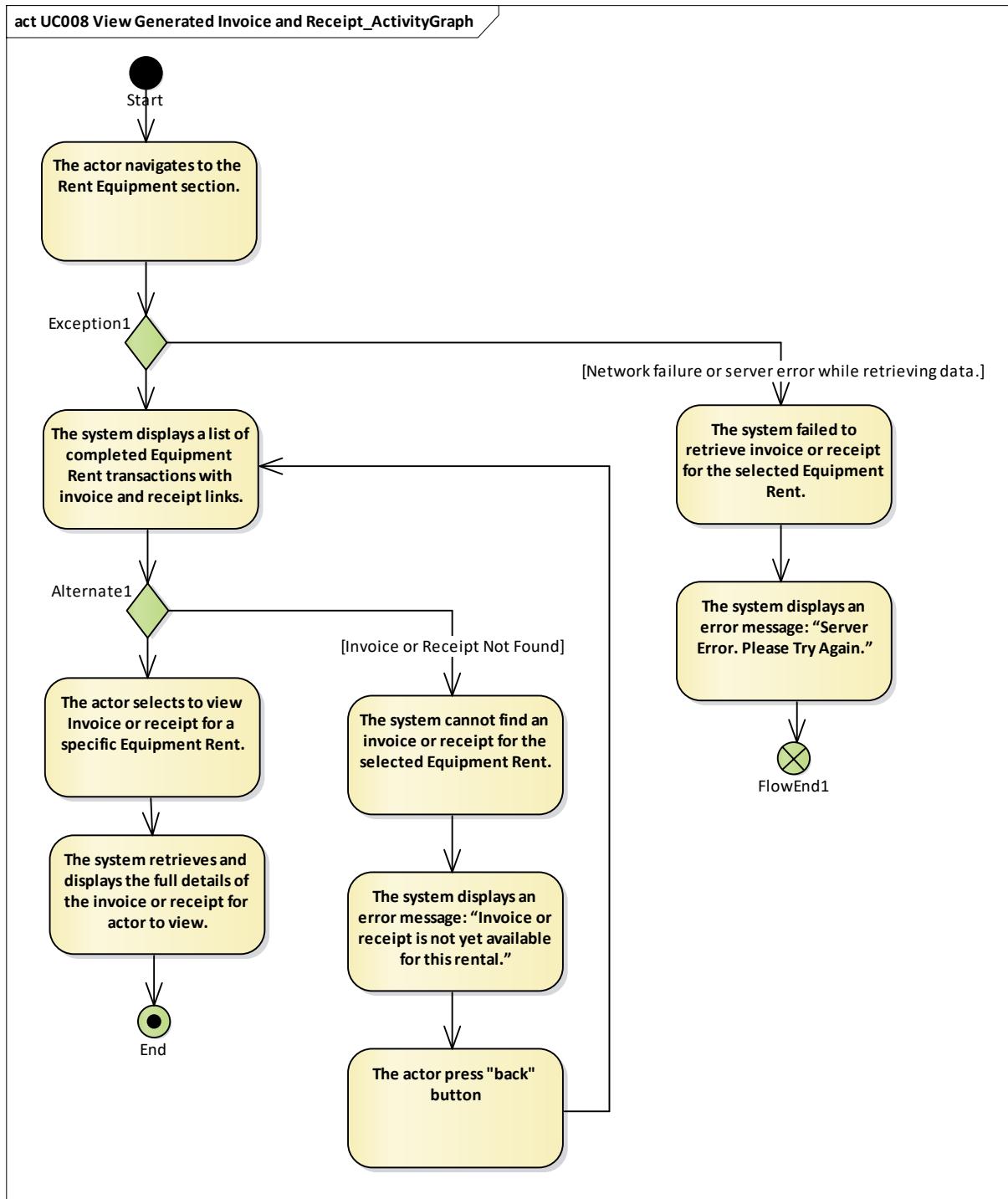


Figure 4.67 Activity Diagram for UC008 View Generated Invoice and Receipt

4.10.2.2.6 UC009: Request Equipment

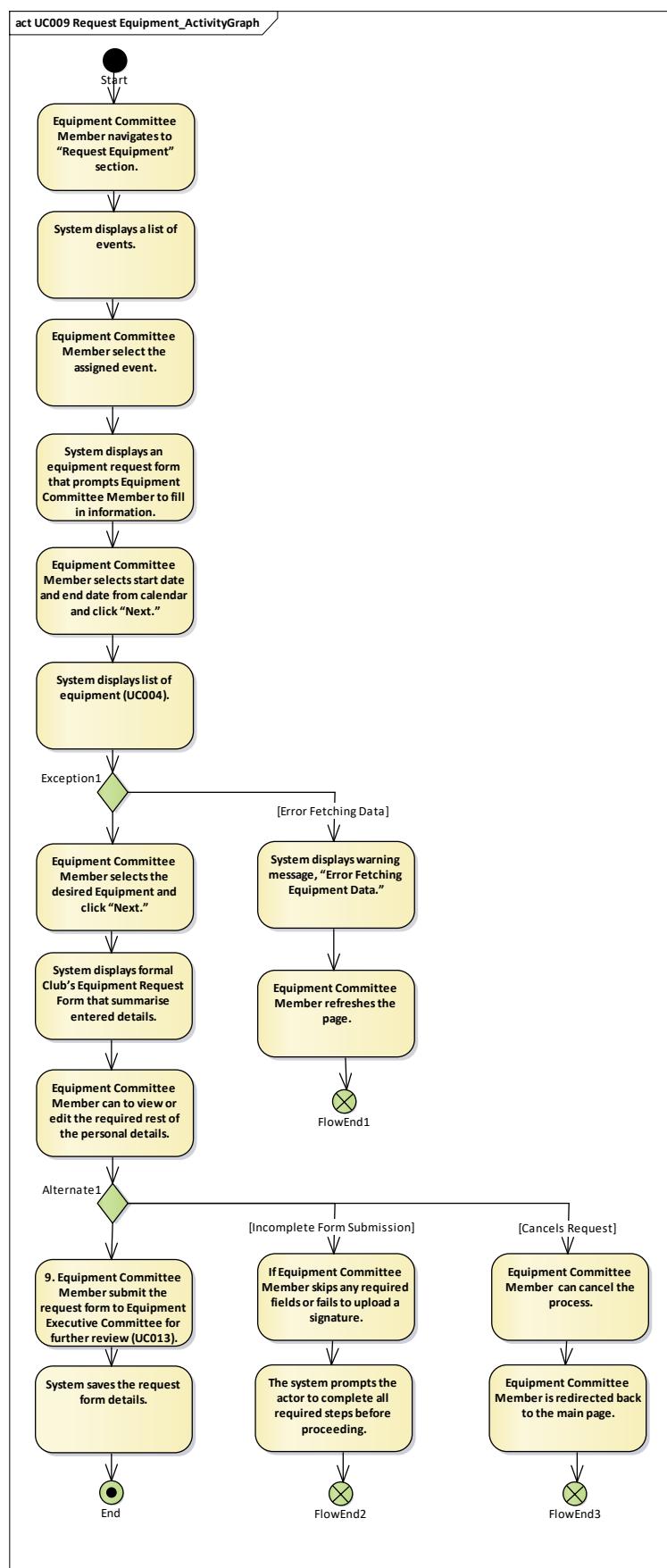


Figure 4.68 Activity Diagram for UC009 Request Equipment

4.10.2.3 Equipment Request and Rental Module

4.10.2.3.1 UC010: Monitor Equipment

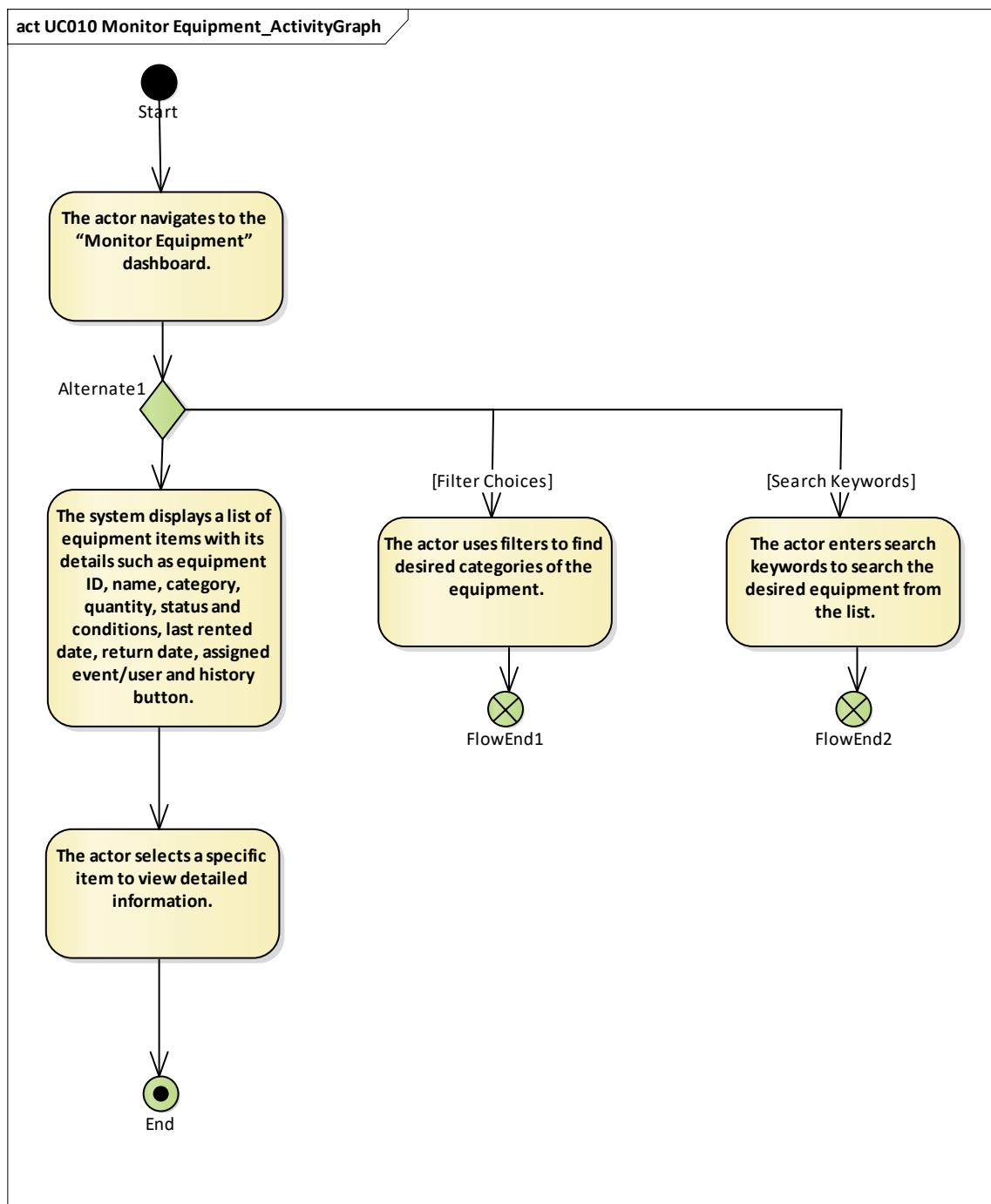


Figure 4.69 Activity Diagram for UC010 Monitor Equipment

4.10.2.3.2 UC011: Update Equipment

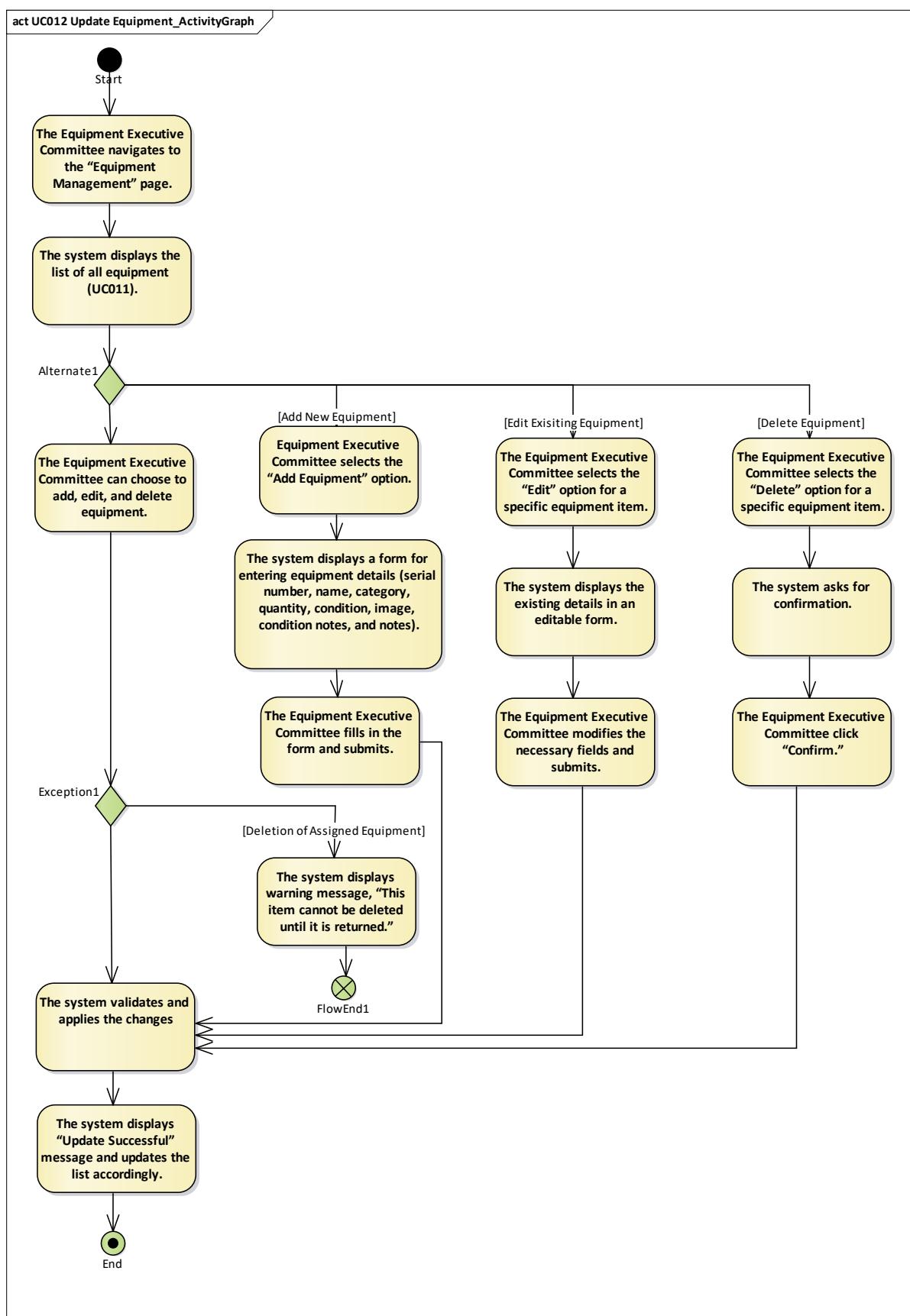


Figure 4.70 Activity Diagram for UC011 Update Equipment

4.10.2.3.3 UC012: Manage Equipment Requests and Rentals

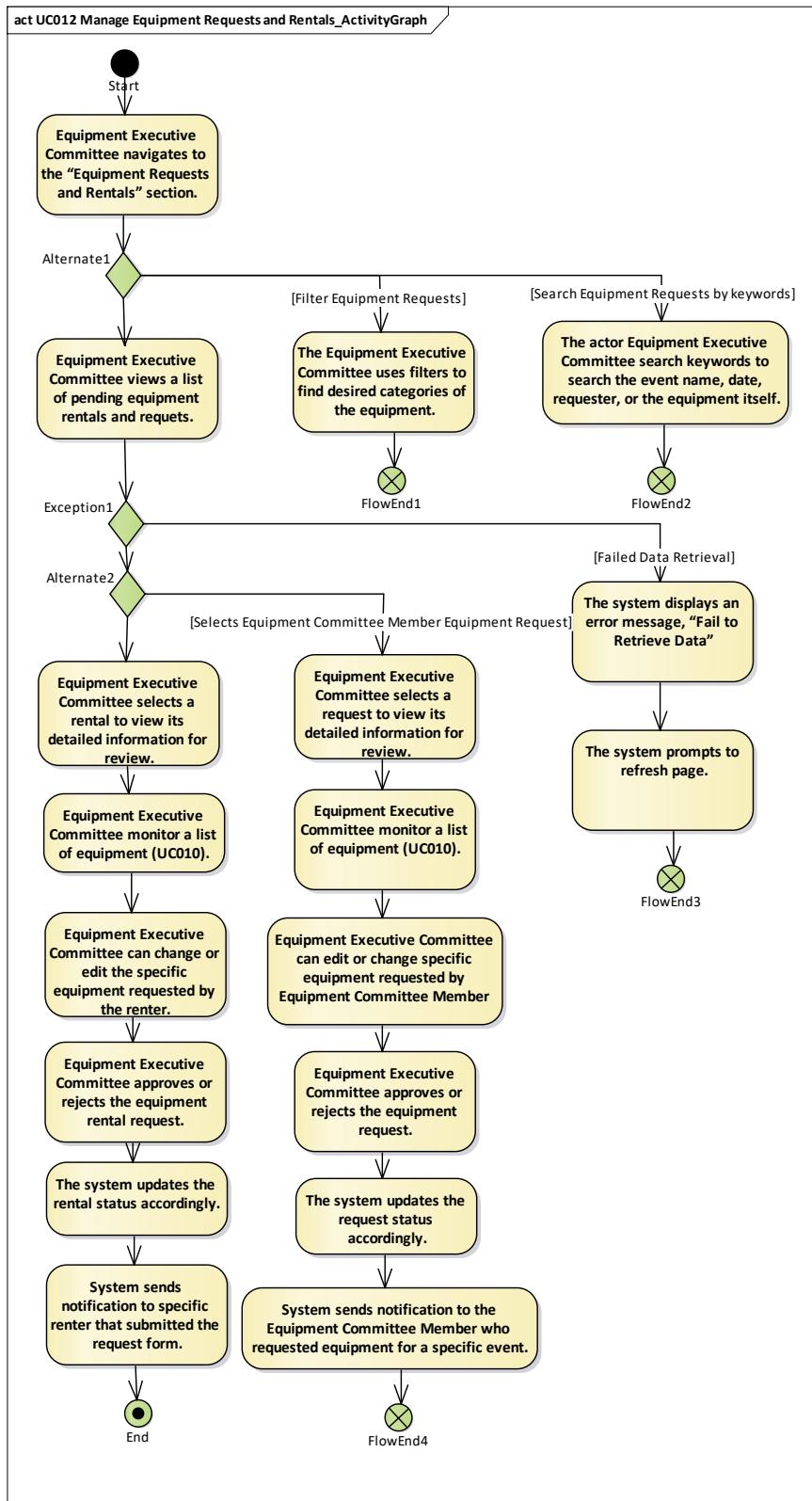


Figure 4.71 Activity Diagram for UC012 Manage Equipment Requests and Rentals

4.10.2.3.4 UC013: Confirm Equipment Return

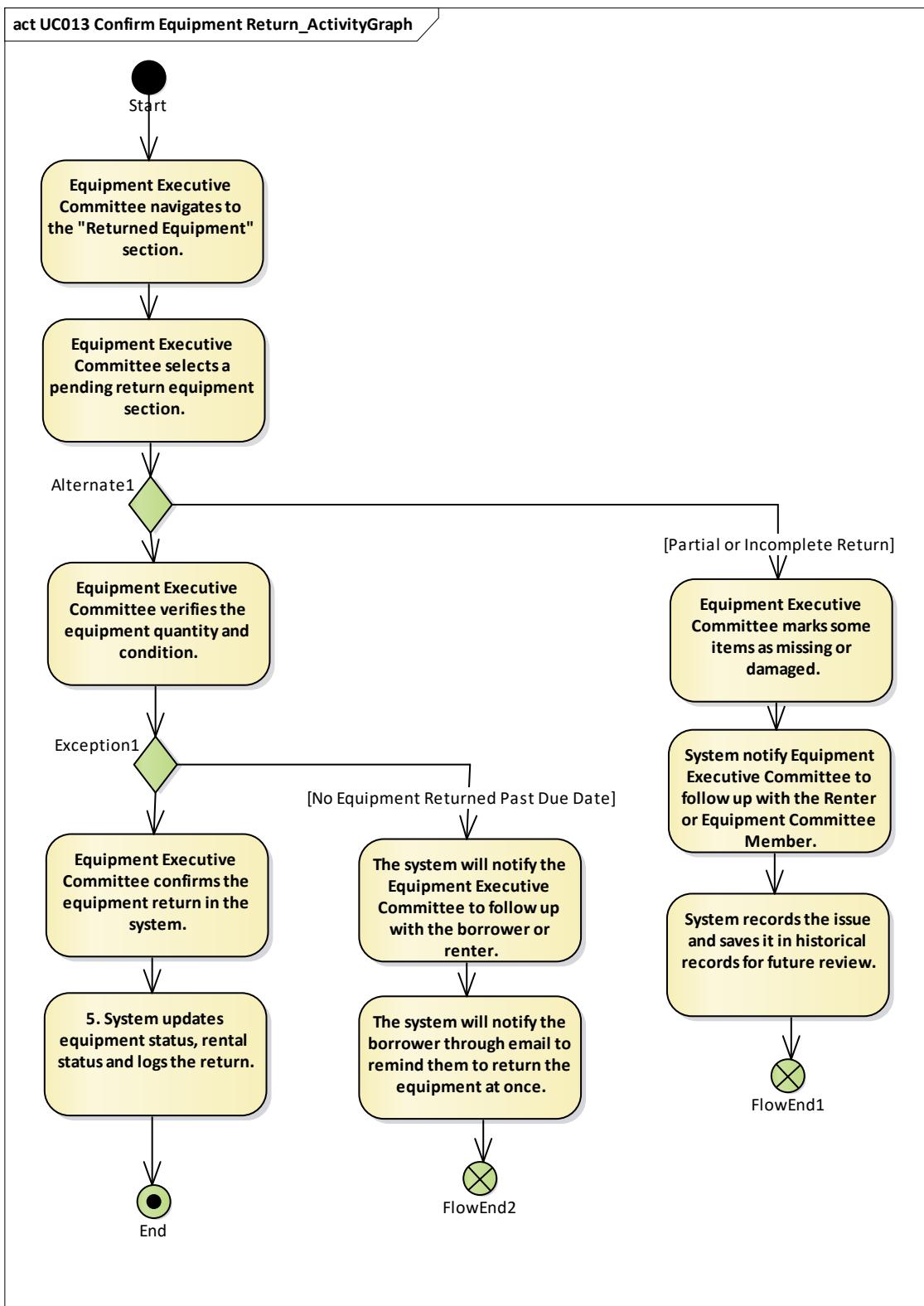


Figure 4.72 Activity Diagram for UC013 Confirm Equipment Return

4.10.2.4 Role and Event Management Module

4.10.2.4.1 UC014: Manage Members Roles

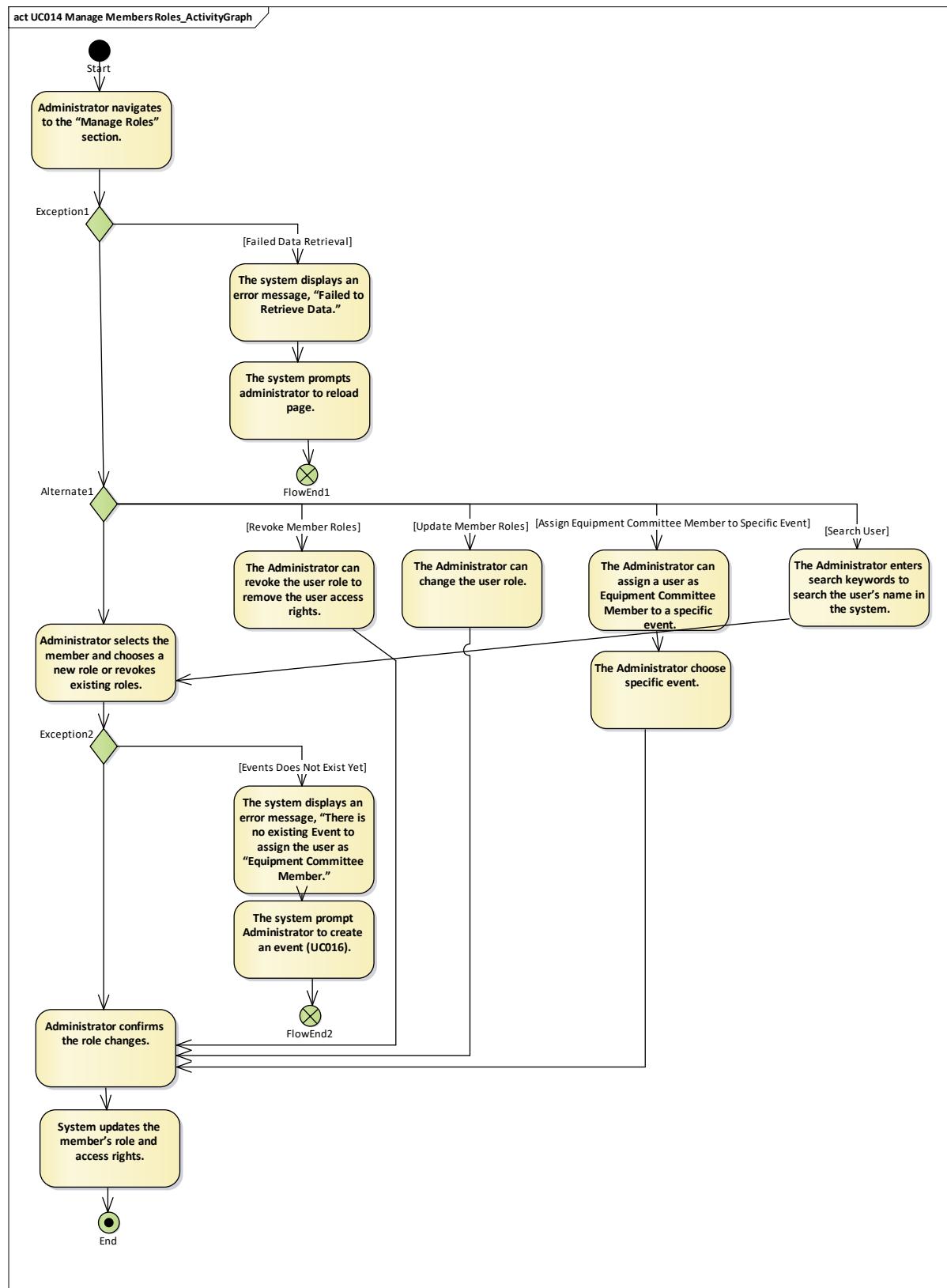


Figure 4.73 Activity Diagram for UC014 Manage Member Roles

4.10.2.4.2 UC015: Manage Event

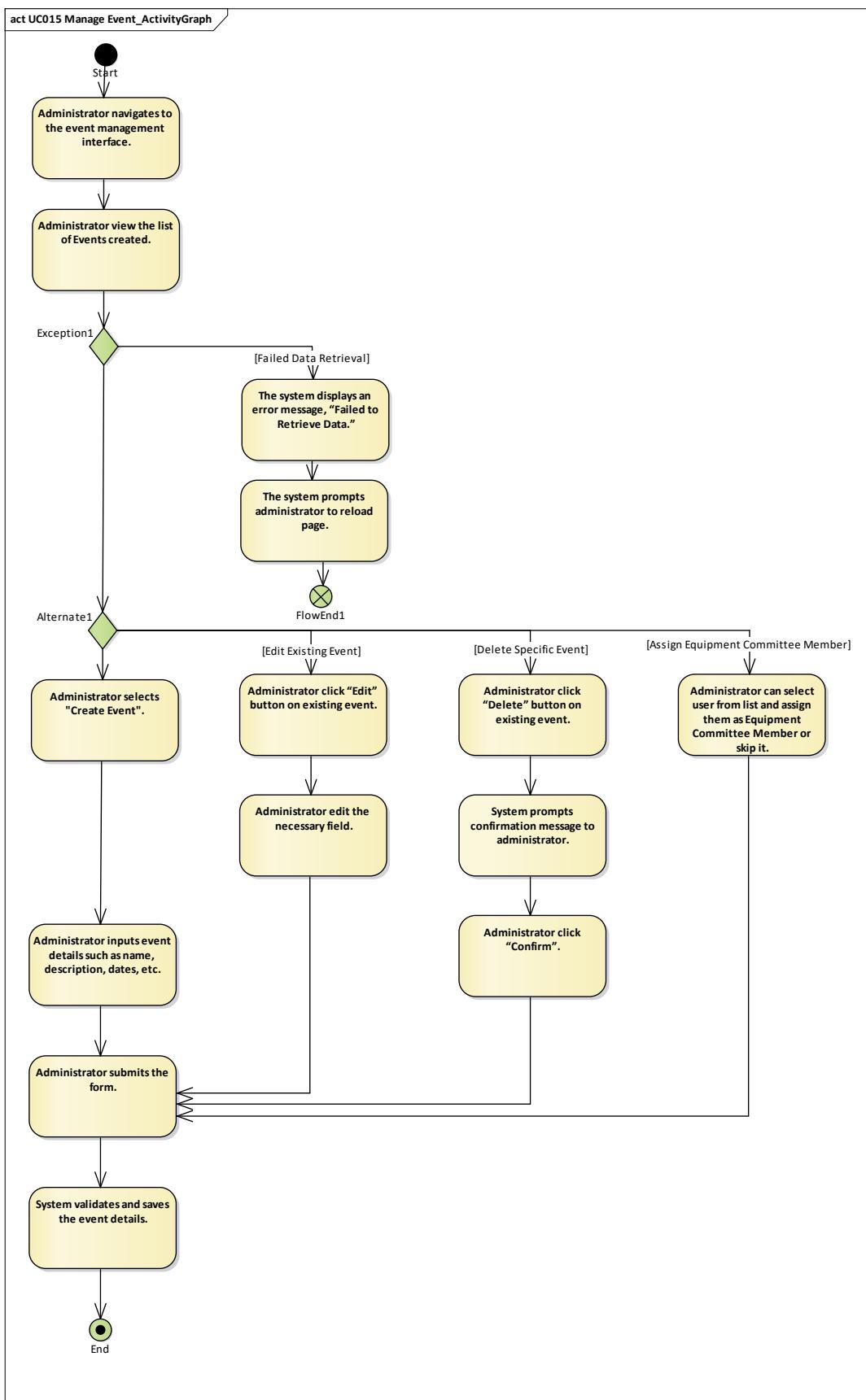


Figure 4.74 Activity Diagram for UC015 Manage Event

5 User Interface Design

5.1 User Interface for Equipment Request and Rentals Module

The screenshot shows the 'Rent Equipment' module. At the top, there is a search bar labeled 'Search product, supplier, order'. On the right side, there are icons for a bell and a user profile. Below the search bar, there are two tabs: 'Cart' and 'Rental List', with 'Rental List' being active. The main area is titled 'Rent Equipment' and contains two sections: 'Camera' and 'Camera Lens'. Each section has a table with columns: No., Brand, Model, Condition, Status, Notes, and Action. The 'Camera' section lists 8 items, and the 'Camera Lens' section lists 2 items. Each row in the tables includes an 'Add' button in the Action column.

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	SD Mark IV	Good	Available	-	Add
2	Canon	SD Mark IV	Good	Available	-	Add
3	Canon	SD Mark II	Broken	Unavailable	EU	Add
4	Canon	SD Mark II	Good	Available	-	Add
5	Canon	60D	Good	Available	-	Add
6	Nikon	D90	Broken	Unavailable	KBV	Add
7	Nikon	D90	Good	Available	-	Add
8	Nikon	D7000	Good	Available	-	Add

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Add
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	Add

Figure 5.1 Interface of List of Equipment (Rent Equipment)

The screenshot shows the 'Cart' page, which is part of the 'Equipment List > Cart'. At the top, there is a search bar labeled 'Search product, supplier, order'. On the right side, there is a 'Rental Form' button. The main area is titled 'Cart' and contains two sections: 'Camera' and 'Camera Lens'. Each section has a table with columns: No., Brand, Model, Condition, Status, Notes, and Action. The 'Camera' section lists 3 items, and the 'Camera Lens' section lists 2 items. Each row in the tables includes 'Edit' and 'Remove' buttons in the Action column.

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	SD Mark IV	Good	Available	-	Edit Remove
2	Canon	SD Mark IV	Good	Available	-	Edit Remove
3	Canon	SD Mark II	Good	Available	EU	Edit Remove

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Edit Remove
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	Edit Remove

Figure 5.2 Interface of Cart Page (Rent Equipment)

Rental Form

Renter Name	<input type="text"/>
Phone No.	<input type="text"/>
Email	<input type="text"/>
Matric No.	<input type="text"/>
Program Name	<input type="text"/>
Venue	<input type="text"/>
Start Date	<input type="text"/>
End Date	<input type="text"/>

Cancel **Submit Form**

Figure 5.3 Interface of Rental Form (Rent Equipment)

Rental List

Manage Equipment Rent			
FutureReady (ID123124123123) 1A/Krighnarajapuram, 3 rd street sulur Muhammad Taufiq 20/6/2025 - 25/6/2025		Status: Approved (Unpaid)	Pay Now View Invoice View Receipt
CONVOCATION - 68 (ID123124123123) 1A/Krighnarajapuram, 3 rd street sulur Ahmad Hazim 23/6/2025 - 26/6/2025		Status: Rejected	Pay Now View Invoice View Receipt
WEDDING (ID123124123123) 1A/Krighnarajapuram, 3 rd street sulur Nabil Iman 29/6/2025 - 1/7/2025		Status: Approved (Paid)	Return Equipment View Invoice View Receipt

Figure 5.4 Interface of List of Equipment Rents (Rent Equipment)

The screenshot shows the 'FutureReady details' page. At the top, there's a search bar and a user profile icon. Below the header, the breadcrumb navigation shows 'Equipment List > Rental List > FutureReady'. The main content area displays the rental information for 'FutureReady (ID123124123123)'. The details include the address '1A/Kirinagarajapuram, 3 rd street sulur', total amount 'RM 75.00', and rental period '20/6/2025 - 25/6/2025'. The status is 'Approved'. There are 'Pay Now' and 'Cancel' buttons. Below this, there are two tables: one for 'Camera' (listing three Canon 5D Mark IV models) and one for 'Camera Lens' (listing one Canon Prime lens). Both tables have columns for No., Brand, Model, Condition, Status, Notes, and Action (Edit, Remove). On the left sidebar, there are links for 'Rent Equipment', 'Orders', 'Settings', and 'Log Out'.

Figure 5.5 Interface of Specific Equipment Rent Details (Rent Equipment)

The screenshot shows the 'Invoice' page. The header includes a search bar and a user profile icon. The breadcrumb navigation shows 'Equipment List > Rental List > FutureReady > Invoice'. The main content area displays an invoice from 'UTM' dated '19 MAY 2025'. The invoice details a rental for '1 Set Camera (Body + Lens biasa)' and a 'Speedlight', totaling 'RM95'. Payment information shows a transaction ID '00000000000000000000000000000000' and account details for 'MUHAMMAD ALIF NABIL'. The invoice is signed off by 'Accepted by HAKAL HAKIM'. On the left sidebar, there are links for 'Rent Equipment', 'Orders', 'Settings', and 'Log Out'.

Figure 5.6 Interface of Invoice (Rent Equipment)

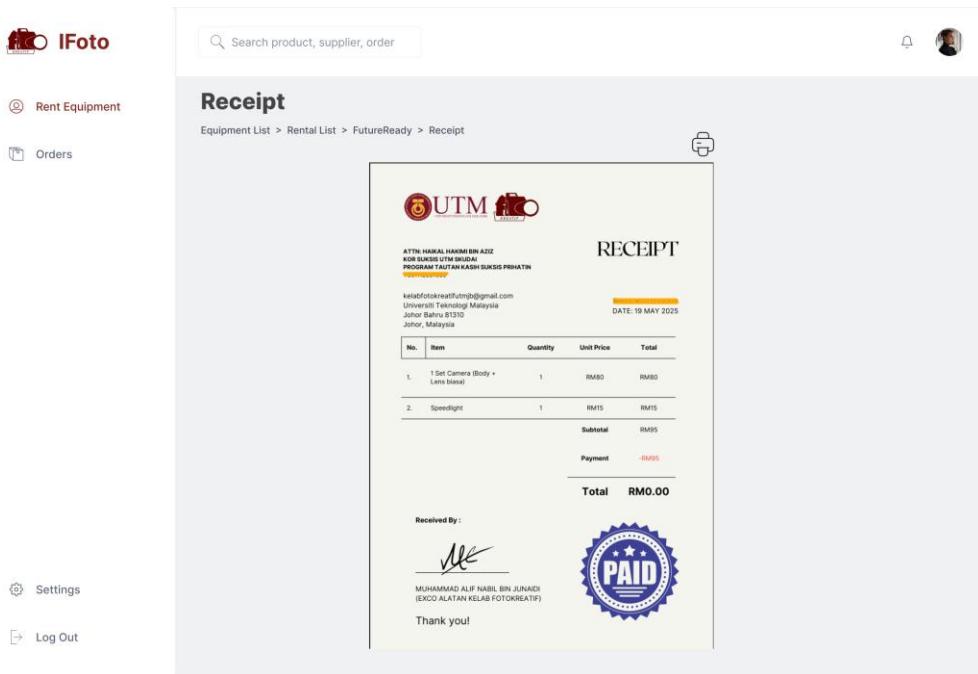


Figure 5.7 Interface of Receipt (Rent Equipment)

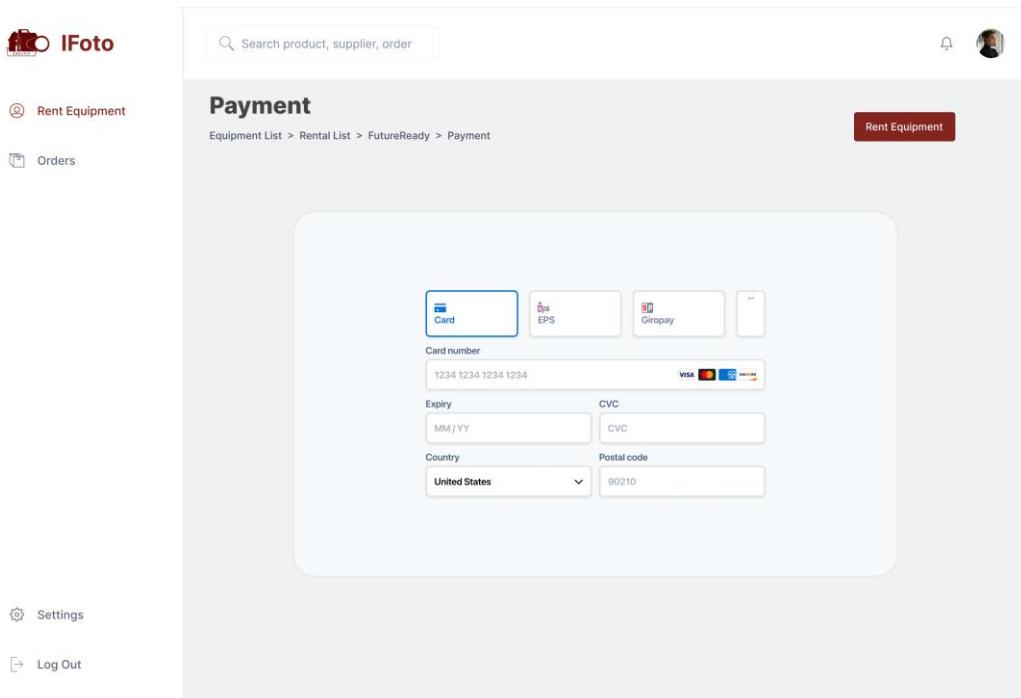


Figure 5.8 Interface of Payment Page (Rent Equipment)

The screenshot shows the 'FutureReady details' page within the iFoto Rent Equipment application. At the top, there's a search bar and a user profile icon. Below the header, a breadcrumb navigation shows 'Equipment List > Rental List > FutureReady'. The main content area displays a summary for 'FutureReady (ID123124123123)'. It includes the address '1A/Krihnarajapuram, 3 rd street sulur', total amount 'RM 75.00', contact 'Muhammad Taufiq', rental period '20/6/2025 - 25/6/2025', and status 'Approved'. A 'Confirm Return' button is visible. Below this, two tables show equipment inventories: 'Camera' and 'Camera Lens', both with columns for No., Brand, Model, Condition, Status, Notes, and Action (Edit, Remove). On the left sidebar, there are links for 'Rent Equipment', 'Orders', 'Settings', and 'Log Out'.

Figure 5.9 Interface of Equipment Return Page (Rent Equipment)

The screenshot shows the 'Cart' page within the iFoto Rent Equipment application. The title 'Cart' is at the top, along with a 'Rental Form' button. A breadcrumb navigation shows 'Equipment List > Cart'. The main content area displays a 'Camera' table with three items. A modal window is open in the center, titled 'SUCCESS' with a green checkmark icon. The message inside the modal reads: 'You have successfully submitted a Equipment Rental Form. Please wait approval before proceed with payment.' In the background, a 'Camera Lens' table is partially visible. On the left sidebar, there are links for 'Rent Equipment', 'Orders', 'Settings', and 'Log Out'.

Figure 5.10 Example of Interface for Successful Action (Rent Equipment)

Figure 5.11 Interface of List of Event (Request Equipment)

Figure 5.12 Interface of List of Equipment (Request Equipment)

Figure 5.13 Interface of Cart Page (Request Equipment)

Figure 5.14 Interface of Equipment Request Form (Request Equipment)

The screenshot shows the iFoto software interface. At the top left is the logo 'iFoto'. To its right is a search bar with the placeholder 'Search product, supplier, order'. Further right are a notification bell icon and a user profile picture.

On the left side, there is a vertical sidebar with navigation links: 'Rent Equipment' (selected), 'Orders', 'Settings', and 'Log Out'.

The main content area has a title 'KBV details' and a breadcrumb trail: 'Equipment List > Rental List > FutureReady'. Below this is a section titled 'KBV' containing a summary box:

KBV (ID123124123123)	Return Equipment
1A/Krighnarajapuram, 3 rd street sulur	
Ahmad Saifudin	
20/6/2025 - 25/6/2025	Status: Ongoing

Below this is a table titled 'Camera' with the following data:

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	5D Mark IV	Good	Available	-	Edit Remove
2	Canon	5D Mark IV	Good	Available	-	Edit Remove
3	Canon	5D Mark II	Good	Available	EU	Edit Remove

Below the camera table is another table titled 'Camera Lens' with the following data:

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Edit Remove

Figure 5.15 Interface of Specific Equipment Request Details (Request Equipment)

5.2 User Interface for Equipment Management Module

The screenshot shows the 'Club's Equipment List' interface. On the left sidebar, there are links for Rent Equipment, Manage Equipment (which is selected), Equipment Requests, Equipment Returns, Orders, Settings, and Log Out. The main area displays two tables:

Camera							
No.	Brand	Model	Serial Number	Condition	Status	Notes	Action
1	Canon	5D Mark IV	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
2	Canon	5D Mark IV	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
3	Canon	5D Mark II	XXX12390	Broken	Unavailable	EU	<button>Edit Equipment</button> <button>Remove</button>
4	Canon	5D Mark II	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
5	Canon	60D	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
6	Nikon	D90	XXX12390	Broken	Unavailable	KBV	<button>Edit Equipment</button> <button>Remove</button>
7	Nikon	D90	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
8	Nikon	D7000	XXX12390	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>

Camera Lens							
No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>
2	Canon	Normal	Canon EF17-40mm USM	Good	Available	-	<button>Edit Equipment</button> <button>Remove</button>

Figure 5.16 Interface of List of Club's Equipment (Manage Equipment)

The screenshot shows the 'Update Equipment Form' dialog box over the 'Club's Equipment List' interface. The dialog has fields for Brand, Model, Serial Number, Condition, Status, and Notes. It also includes a placeholder for an image and a 'Save Equipment' button. The background shows the same equipment lists as Figure 5.16.

Figure 5.17 Interface of Update Equipment Form (Manage Equipment)

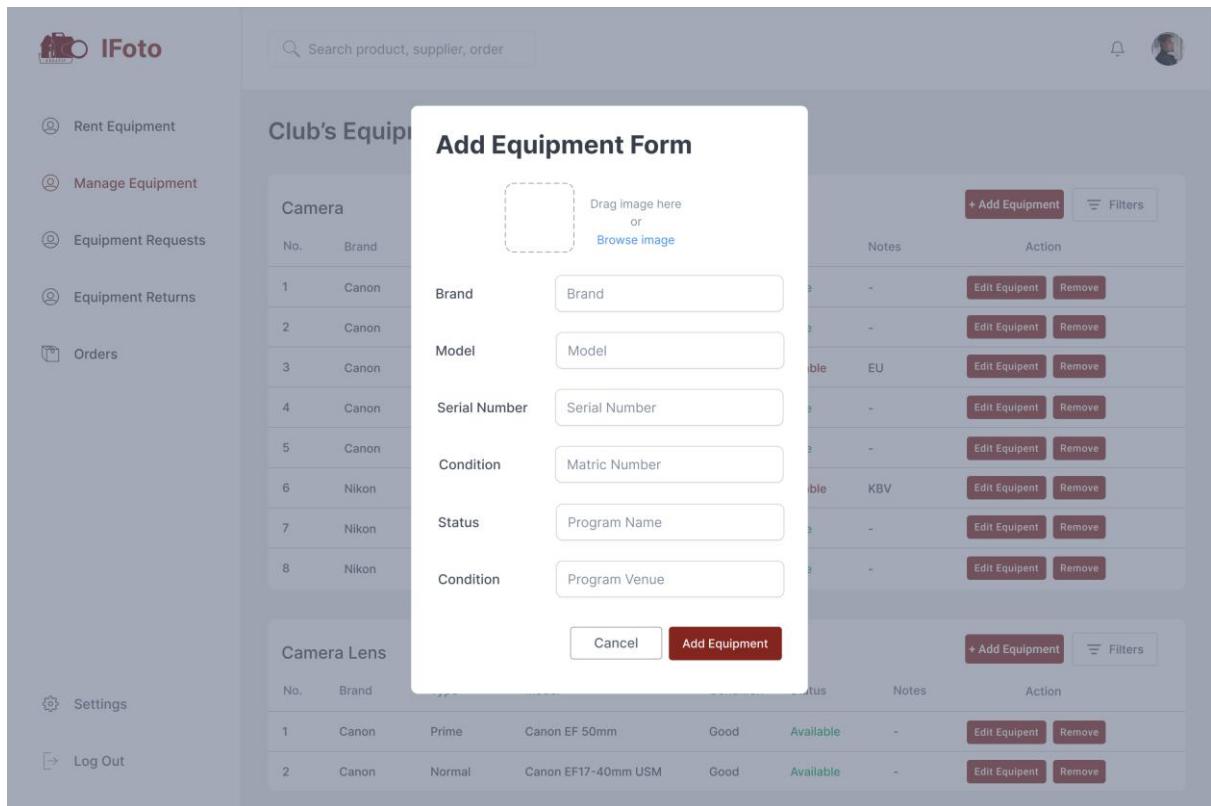


Figure 5.18 Interface of Add Equipment Form (Manage Equipment)

Figure 5.19 Interface of List of Equipment Request and Rentals (Manage Equipment Requests and Rentals)

The screenshot shows the 'Manage Equipment Requests and Rentals' section of the IFoto software. On the left sidebar, there are several navigation items: Rent Equipment, Manage Equipment, Equipment Requests (which is currently selected and highlighted in red), Equipment Returns, and Orders. The main content area has a search bar at the top. Below it, a title 'Manage Equipment Requests and Rentals' is followed by a breadcrumb 'Requests and Rentals List > FutureReady'. A sub-section titled 'FutureReady' displays a request card for 'FutureReady (ID123124123123)'. The card includes details: Address (1A/Krihnarajapuram, 3 rd street sulur), Total Amount (RM 75.00), Contact (Muhammad Taufiq), Date Range (20/6/2025 - 25/6/2025), and Status (Approved). Action buttons for 'Approve' (green) and 'Reject' (red) are shown, along with links to 'View Invoice' and 'View Receipt'. Below this, there are two tables: 'Camera' and 'Camera Lens', each with columns for No., Brand, Model, Condition, Status, Notes, and Action. The 'Camera' table lists three items: Canon 5D Mark IV, Canon 5D Mark IV, and Canon 5D Mark II. The 'Camera Lens' table lists one item: Canon Prime EF 50mm. Both tables include 'Manage' and 'Remove' buttons for each row.

Figure 5.20 Interface of Managing Equipment Request and Rentals (Manage Equipment Requests and Rentals)

The screenshot shows the 'Manage Equipment Returns' section of the IFoto software. The left sidebar includes the same navigation items as Figure 5.20, with 'Equipment Returns' being the active item. The main content area features a search bar. Below it, a title 'Manage Equipment Returns' and a subtitle 'Equipment Return List'. A sub-section titled 'Manage Equipment Rent Return' displays a card for 'FutureReady (ID123124123123)'. The card shows the same details as in Figure 5.20, but the status is now 'Pending Return' (orange). An 'Manage' button is present. Another card for 'CONVOCATION - 68 (ID123124123123)' is shown below, with a status of 'Returned' (green) and an 'Manage' button. A third sub-section titled 'Manage Equipment Request Return' displays a card for 'KBV (ID123124123123)', which is currently approved (Status: Approved, green). An 'Manage' button is also present here. The bottom of the sidebar includes 'Settings' and 'Log Out' options.

Figure 5.21 Interface of List of Equipment Returns (Manage Equipment Returns)

Manage Equipment Returns

Equipment Return List > FutureReady

FutureReady

FutureReady (ID123124123123)

1A/Krighnarajapuram, 3 rd street sulur
Muhammad Taufiq
20/6/2025 - 25/6/2025 Status: Pending Return

Approve **Reject**

No.	Brand	Model	Condition	Status	Notes	Action
1	Canon	5D Mark IV	Good	Available	-	Manage Remove
2	Canon	5D Mark IV	Good	Available	-	Manage Remove
3	Canon	5D Mark II	Good	Available	EU	Change Remove

No.	Brand	Type	Model	Condition	Status	Notes	Action
1	Canon	Prime	Canon EF 50mm	Good	Available	-	Manage Remove

Camera

Camera Lens

Settings

Log Out

Figure 5.22 Interface of Managing Equipment Returns (Manage Equipment Returns)

5.3 User Interface for Role and Event Management Module

The screenshot shows the 'Event List' page. At the top right is a search bar with placeholder text 'Search product, supplier, order'. On the far right is a user profile icon. Below the search bar is a button '+ Add Event'. The main area is titled 'Event List' and contains a table with three rows of event data:

No.	Name	Description	Location	Status	Start Date	End Date	Action
1	KBV	Video Lesson	SUB	Ongoing	23/5/2025	28/5/2025	<button>Manage Event</button> <button>Manage Member</button>
2	MMM	Minggu Mesra Mahasiswa	DSI	Ongoing	23/5/2025	28/5/2025	<button>Manage Event</button> <button>Manage Member</button>
3	Wedding	Perkahwinan Adam	Pondok	Upcoming	23/6/2025	28/6/2025	<button>Manage Event</button> <button>Manage Member</button>

The left sidebar includes links for Rent Equipment, Manage Event, Orders, Settings, and Log Out. The top left corner has a logo 'iFoto' with a camera icon.

Figure 5.23 Interface of List of Event

The screenshot shows the 'Manage Member' page. At the top right is a search bar with placeholder text 'Search product, supplier, order'. On the far right is a user profile icon. Below the search bar is a breadcrumb 'Event List > Manage Equipment Committee Member'. The main area is titled 'Manage Member' and contains two tables:

Assign Equipment Committee Member to KBV

No.	Name	email	Phone No.	Matric No.	Roles	Action
1	Muhammad Taufiq	example@gmail.com	011-XXXXXXX	A22ECXXX	<button>View Roles</button>	<button>+ Add</button>
2	Muhammad Nabil	example@gmail.com	011-XXXXXXX	A22EMXXX	<button>View Roles</button>	<button>+ Add</button>
3	Ahmad Saifudin	example@gmail.com	011-XXXXXXX	A23ESXXX	<button>View Roles</button>	<button>+ Add</button>

Assigned Equipment Committee Member to KBV

No.	Name	email	Phone No.	Matric No.	Roles	Action
1	Muhammad Alif	example@gmail.com	011-XXXXXXX	A22ECXXX	<button>View Roles</button>	<button>Remove</button>
2	Ahmad Hazim	example@gmail.com	011-XXXXXXX	A22EMXXX	<button>View Roles</button>	<button>Remove</button>

The left sidebar includes links for Rent Equipment, Manage Event, Orders, Settings, and Log Out. The top left corner has a logo 'iFoto' with a camera icon.

Figure 5.24 Interface of Manage Club Members

The screenshot shows the iFoto software interface. At the top left is the logo 'iFoto'. To the right is a search bar with placeholder text 'Search product, supplier, order'. Further right are a notification bell icon and a user profile picture.

The main area features a central 'Event List' dialog box with a white background. At the top of this dialog is the title 'Event List' and a red button '+ Add Event'. Below the title is a section titled 'Add Event Form'.

The form contains several input fields:

- 'Event Name': A text input field containing 'Renter Name'.
- 'Description': A text input field containing 'Phone Number'.
- 'Location': A text input field containing 'Email'.
- 'Matric No.': A text input field containing 'Matric Number'.
- 'Start Date': A date input field showing 'dd/mm/yy'.
- 'End Date': A date input field showing 'dd/mm/yy'.

Below these fields is a section titled 'Assign Equipment Committee Member' with a red '+ Add' button.

At the bottom of the dialog are two buttons: 'Cancel' and 'Create Event'.

On the left side of the interface, there is a sidebar with the following menu items:

- Rent Equipment
- Manage Event
- Orders
- Settings

On the right side, there is a table titled 'Event List' with columns: Start Date, End Date, and Action. It displays three rows of event data:

Start Date	End Date	Action
5/5/2025	28/5/2025	Manage Event Manage Member
5/5/2025	28/5/2025	Manage Event Manage Member
6/6/2025	28/6/2025	Manage Event Manage Member

At the bottom left is a 'Log Out' link.

Figure 5.25 Interface of Add Event Form

Appendix D Software Test Documentation (STD)



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECJ 3032: Software Engineering FYP1
Semester 02, 2024/2025

Software Test Documentation

(STD)

Inventory Management System for Kelab
Fotokreatif (IFoto)

Version 1.0

21/06/2025

Prepared by: Lio Kock Hock

Revision Page

1.1 Overview

This current document contains the introduction of Inventory Management System for Kelab Fotokreatif (IFoto), overall description of the system, and details of test cases for each use case.

1.2 Target Audience

The target audience are Kelab Fotokreatif members, developers, project managers and software testers.

1.3 Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Lio Kock Hock	Completed STD version 1.0	21/06/2025

Table of Contents

1	Introduction	263
1.1	Purpose.....	263
1.2	Scope.....	263
1.3	Definitions, Acronyms and Abbreviation	264
1.4	References.....	264
1.5	Overview	264
2	Test Cases, Data and Expected Results.....	266
2.1	Authentication and Authorization Module.....	266
2.1.1	Test TC001: Register an Account.....	266
2.1.2	Test TC002: Login	269
2.1.3	Test TC003: Reset Password	271
2.2	Equipment Request and Rental Module	275
2.2.1	Test TC004: View Equipment List.....	275
2.2.2	Test TC005: Rent Equipment	276
2.2.3	Test TC006: Return Equipment	278
2.2.4	Test TC007: Make Payment	279
2.2.5	Test TC008: View Generated Invoice and Receipt.....	280
2.2.6	Test TC009: Request Equipment.....	282
2.3	Equipment Management Module	283
2.3.1	Test TC010: Monitor Equipment	283
2.3.2	Test TC011: Update Equipment	285
2.3.3	Test TC012: Manage Equipment Requests and Rentals	287
2.3.4	Test TC013: Confirm Equipment Return.....	290
2.4	Role and Event Management Module	292
2.4.1	Test TC014: Manage Members Roles	292
2.4.2	Test TC015: Manage Event.....	295

1 Introduction

1.1 Purpose

Software Test Document (STD) is an exhaustive document that is used to define the scope, objective and testing strategy of IFoto Inventory Management System to be tested to produce successful results. It is aimed mainly at the validation of all the functional and non-functional requirements provided in SRS with the help of clearly stated test cases. This document gives good guidelines to testers details with test cases, input and expected results along with pass and failure conditions. It serves as a basis of developers, quality assurance engineers and project stakeholders to verify the behaviour of the systems under specific conditions, identify defects at an early stage and the final product so that it behaves as per the expectations intended by the Kelab Fotokreatif. Finally, the STD facilitates the systematic and regular testing efforts during software development lifecycle.

1.2 Scope

This Software Test Document (STD) will cover all the activities of testing undertaken about the IFoto Inventory Management System designed to support Kelab Fotokreatif. In this document, the testing strategy will be introduced and the test cases needed to cover the functionality, performance and reliability of the system would be defined as per the requirements given in Software Requirement Specification (SRS).

The STD presents unit test, integration test, system test and acceptance test of all modules of an application i.e.:

- Authentication and Authorization Module
- Equipment Management Module
- Equipment Request and Rent Module
- Role and Event Management Module

This document as well discusses positive and negative test cases of critical functions like user registration, equipment renting and returning, payment processing,

role assignment and events management. It states the input data, expected outputs, test conditions and over which conditions the system should pass or fail so that it is at its intended purpose, and works correctly with a range of conditions.

Testing will then be done in an environmentally-controlled development and staging environment, as it would be used in a real world. This will guarantee that the final output given to Kelab Fotokreatif will work well, safely and effectively.

1.3 Definitions, Acronyms and Abbreviation

Definitions of all terms, acronyms and abbreviation used are:

SRS – Software Requirements Specifications

STD – Software Testing Document

UML – Unified Modelling Language

1.4 References

IEEE. (1998). *IEEE Standard for Software Test Documentation (IEEE 829-1998)*.

Institute of Electrical and Electronics Engineers.

<https://doi.org/10.1109/IEEESTD.1998.88286>

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

1.5 Overview

This Software Test Document (STD) is a description of the testing strategy and individual test cases of Inventory Management System at Kelab Fotokreatif (IFoto). It makes sure that all functional and non-functional requirements included in Software Requirements Specification (SRS) are tested using planned out tests.

Firstly, STD contains test case specification. Every test case has properties like test ID, test description, test conditions, input data, expected results and actual results. The test cases shall also be set to test the positive cases and the negative ones so that the behaviour of the system is tested well (IEEE 829-1998 Standard for Software Test Documentation).

Moreover, the STD can be used to support regression testing and user acceptance testing (UAT) so that the stakeholders can decide whether they may start deploying the system. It also correlates with software quality practices in establishing the software against its own pre-determined requirements, which helps to make the software reliable, maintainable, and useful (Pressman, R.S., & Maxim, B.R., 2020).

It is an important document that can be used as a method of communication among testers, developers, and stakeholders to ascertain that the system provided to them is correctly running, reliable, and matches the expectations of Kelab Fotokreatif.

2 Test Cases, Data and Expected Results

2.1 Authentication and Authorization Module

2.1.1 Test TC001: Register an Account

2.1.1.1 TC001_01: Successful Registration with Valid Inputs

Table 2.1 Test Case for TC001_01: Successful Registration with Valid Inputs

Test Case ID	TC001_01	Test Case Description	Verifies that a user can register successfully with valid and complete input data		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	User is on the registration page		1	Email: user@example.com	
			2	Password: Pass1234	
			3	Name: Ali	
			4	Matric No: A22RT0123	
Test Conditions	All fields are valid and not empty				
Step #	Step Details	Expected Result			
1	Enter valid data in all fields				
2	Click "Register"	User account is created and redirected to login			

2.1.1.2 TC001_02: Register with Already Registered Email

Table 2.2 Test Case for TC001_02: Register with Already Registered Email

Test Case ID	TC001_02	Test Case Description	Ensures that the system does not allow one to register on an email address that has already been taken up		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	User is on the user registration page		1	Email: user@example.com	
2	The email ali@example.com is already registered		2	Password: Pass1234	
			3	Name: Ali	
			4	Matric No: A22RT0123	
Test Conditions	System checks for existing email in DB				
Step #	Step Details	Expected Result			
1	Enter duplicate email				
2	Click "Register"	System displays error message showing: Email already exists			

2.1.1.3 TC001_03: Register with Password Mismatch

Table 2.3 Test Case for TC001_03: Register with Password Mismatch

Test Case ID	TC001_03	Test Case Description	Check that behaviour in case of unmatched password and confirm password.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	User is on the user registration page		1	Email: user@example.com	
			2	Password: Pass1234	
			4	Confirm Password: Pass5674	
Test Conditions	Password and confirm password input are different				
Step #	Step Details	Expected Result			
1	Enter valid password and different confirm password				
2	Click “Register”	System displays error message showing: Passwords do not match			

2.1.2 Test TC002: Login

2.1.2.1 TC002_01: Successful Login

Table 2.4 Test Case for TC002_01: Successful Login

Test Case ID	TC002_01	Test Case Description	Ensure that user is able to enter a correct user name and password and log in.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is on the user login page		1	Email: user@example.com
2	Account exists and active.		2	Password: Pass1234
Test Conditions	User able to login into their account with valid credentials			
Step #	Step Details	Expected Result		
1	Enter correct email and password			
2	Click "Login"	User redirected to the right dashboard		

2.1.2.2 TC002_02: Invalid Password

Table 2.5 Test Case for TC002_02: Invalid Password

Test Case ID	TC002_02	Test Case Description	Ensure that system rejects wrong passwords.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	User is on the user login page		1	Email: user@example.com	
2	Account exists and active.		2	Password: WrongPass1234	
Test Conditions	User unable to login into their account with invalid password				
Step #	Step Details	Expected Result			
1	Enter valid email and wrong password				
2	Click “Login”	System displays error message showing: Passwords do not match			

2.1.3 Test TC003: Reset Password

2.1.3.1 TC003_01: Reset with Registered Email

Table 2.6 Test Case for TC003_01: Reset with Registered Email

Test Case ID	TC003_01	Test Case Description	Ensure that reset link is sent to correct and existing email.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is on the user login page		1	Email: user@example.com
2	Email of an Account is registered			
Test Conditions	User will receive Password Reset Link through Email.			
Step #	Step Details	Expected Result		
1	Click "Reset Password"			
2	Enter valid email.			
3	Click "Send Reset Link"	System sends reset link and confirmation message shown		

2.1.3.2 TC003_02: Request Reset with Unregistered Email

Table 2.7 Test Case for TC003_02: Request Reset with Unregistered Email

Test Case ID	TC003_02	Test Case Description	Ensure that reset link is not sent to unregistered Email	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is on the user login page		1	Email: user@example.com
2	Email is not registered to any account			
Test Conditions	User unable to Reset Password of Unregistered Email.			
Step #	Step Details	Expected Result		
1	Click “Reset Password”			
2	Enter Unregistered email.			
3	Click “Send Reset Link”	System displays error message: <i>Email not found</i>		

2.1.3.3 TC003_03: Reset with Valid Token and Passwords

Table 2.8 Test Case for TC003_03: Reset with Valid Token and Passwords

Test Case ID	TC003_03	Test Case Description	Ensure that password has been updated with valid token.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User clicked on the reset link		1	Password: NewPass1234
Test Conditions	User able to update password with valid token			
Step #	Step Details	Expected Result		
1	Enter new password and confirm			
2	Submit	System updates the user password and redirect the user to login page.		

2.1.3.4 TC003_04: Reset with Expired Token

Table 2.9 Test Case for TC003_04: Reset with Expired Token

Test Case ID	TC003_04	Test Case Description	Ensure that password cannot be updated with expired token.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User has clicked on the reset password link.		1	Password: NewPass1234
Test Conditions	User unable to update password with expired token.			
Step #	Step Details	Expected Result		
1	Enter Unregistered email.			
2	Click “Send Reset Link”	System displays error message: Token has expired. Try resend Reset Password link.		

2.2 Equipment Request and Rental Module

2.2.1 Test TC004: View Equipment List

2.2.1.1 TC004_01: Successfully Load Equipment List

Table 2.10 Test Case for TC004_01: Successfully Load Equipment List

Test Case ID	TC004_01	Test Case Description	Check that all Main Equipment and Sub Equipment are displayed and listed accordingly.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	User has signed into an account.				
2	Equipment data exist in database				
Test Conditions	System displays existing Equipment in the database.				
Step #	Step Details	Expected Result			
1	Navigate to "View Equipment" page	System displays List of equipment with its details accordingly.			

2.2.2 Test TC005: Rent Equipment

2.2.2.1 TC005_01: Successfully Submit Equipment Rent

Table 2.11 Test Case for TC005_01: Successfully Submit Rental Request

Test Case ID	TC005_01	Test Case Description	Verify that a user can rent equipment with valid event details and available equipment.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	User has signed into an account.		1	Program: UTM Carnival	
2	Equipment data exist in database.		2	Location: UTM, Skudai, Johor	
			3	Dates: 5–7 July	
			4	Main Equipment: 101	
			5	Sub Equipment: 301, Quantity: 2	
Test Conditions	User can rent equipment with available equipment and valid form fields.				
Step #	Step Details	Expected Result			
1	Go to “Rent Equipment”	System display List of equipment with its details accordingly.			
2	Fill form (program, venue, dates)				
3	Select Equipment				
4	Agree on terms and conditions				
5	Upload signature				
6	Click “Submit” button	System creates new Equipment Rent and displays successful message: Rental Request Submitted.			

2.2.2.2 TC005_02: Try Renting Unavailable Equipment

Table 2.12 Test Case for TC005_02: Try Renting Unavailable Equipment

Test Case ID	TC005_02	Test Case Description	Ensure that unavailable equipment is disabled from booking.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	Main Equipment is marked as Unavailable.		1	Program: UTM Carnival	
2	Sub Equipment has no quantity		2	Location: UTM, Skudai, Johor	
3	Equipment data exist in database		3	Dates: 5–7 July	
			4	Unavailable Main Equipment: 101	
			5	Sub Equipment: 301, Quantity: 0	
Test Conditions	User cannot add unavailable and zero quantity equipment.				
Step #	Step Details	Expected Result			
1	Go to “Rent Equipment”	System display List of equipment with its details accordingly.			
2	Fill form (program, venue, dates)				
3	Select Unavailable Equipment.	System disabled the “Add” button of unavailable equipment and zero quantity equipment.			

2.2.3 Test TC006: Return Equipment

2.2.3.1 TC006_01: Return Equipment Successfully

Table 2.13 Test Case for TC006_01: Try Renting Unavailable Equipment

Test Case ID	TC006_01	Test Case Description	Validate that the Equipment Return process should updates the equipment status, availability, and quantity.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	Equipment Rent has been in Approved.		1	Rent ID: 2001	
2	Payment of Equipment Rent has been made.		2	Main Equipment: 101	
			3	Sub Equipment: 301, Quantity Returned: 2	
Test Conditions	In case of good condition, Main Equipment status should be changed back to Available. Sub Equipment Available Quantity and Used Quantity must be re-calculated and restored.				
Step #	Step Details	Expected Result			
1	Navigate to “Return Equipment” page.	System display List of Equipment Rents with its details accordingly.			
2	Select rental record.				
3	Select and update equipment condition (if needed).				
4	Click “Confirm Return.”	System updates the Main Equipment status and restores the Sub Equipment quantities. System updates Equipment rent as Returned.			

2.2.4 Test TC007: Make Payment

2.2.4.1 TC007_01: Successful Stripe Payment

Table 2.14 Test Case for TC007_01: Successful Stripe Payment

Test Case ID	TC007_01	Test Case Description	Ensure that user make a successful payment.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Equipment Rent has been in Approved.		1	Rent ID: 2001
2	Invoice has been generated by the system.		2	Invoice ID: 4001
Test Conditions	User can make payment with valid payment method, bank details and required amount to correct equipment rent.			
Step #	Step Details	Expected Result		
1	Navigate to “Rental List” page.	System display List of Equipment Rents with its details accordingly.		
2	Click “Pay Now” on selected Rent.			
3	Select payment method			
4	Enter valid bank details			
4	Submit Payment.	System processes the payments and generates receipt.		

2.2.5 Test TC008: View Generated Invoice and Receipt

2.2.5.1 TC008_01: View Invoice After Approval

Table 2.15 Test Case for TC008_01: View Invoice After Approval

Test Case ID	TC008_01	Test Case Description	Ensure that user view Invoice after equipment rent approval.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Equipment Rent has been in Approved.		1	Rent ID: 2001
Test Conditions	User can view existing Invoice that is linked to Equipment Rent.			
Step #	Step Details	Expected Result		
1	Navigate to “Rental List” page.	System display List of Equipment Rents with its details accordingly.		
2	Click “View Invoice” on selected Rent.	System displays invoice.		

2.2.5.2 TC008_02: View Receipt After Payment

Table 2.16 Test Case for TC008_02: View Receipt After Payment

Test Case ID	TC008_02	Test Case Description	Ensure that user view Receipt after payment.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User has successfully completed the payment.		1	Rent ID: 2001
			2	Invoice ID: 4001
Test Conditions	User can view existing receipt that is linked to Equipment Rent and Invoice.			
Step #	Step Details	Expected Result		
1	Navigate to “Rental List” page.	System display List of Equipment Rents with its details accordingly.		
2	Click “View Receipt” on selected Rent.	System displays Receipt.		

2.2.6 Test TC009: Request Equipment

2.2.6.1 TC009_01: Request Equipment for Event

Table 2.17 Test Case for TC009_01: Request Equipment for Event

Test Case ID	TC009_01	Test Case Description	Ensure that Equipment Committee Member can request equipment.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Specific event must exist.		1	Event ID: 5001
2	User is assigned to event as Equipment Committee Member.		2	Main Equipment: 101
			3	Sub Equipment: 301, Quantity: 2
Test Conditions	Equipment Committee Member can request equipment by filling up request form for specific event with valid event selection and available Equipment.			
Step #	Step Details	Expected Result		
1	Navigate to "Request Equipment".	System display List of Event with its details accordingly.		
2	Select specific event.	System display List of Equipment with its details accordingly.		
3	Add specific equipment.			
4	Complete the request form			
5	Submit request form	System saves the equipment request with status "Pending Approval".		

2.3 Equipment Management Module

2.3.1 Test TC010: Monitor Equipment

2.3.1.1 TC010_01: Successfully Monitor All Equipment

Table 2.18 Test Case for TC010_01: Successfully Monitor All Equipment

Test Case ID	TC010_01	Test Case Description	Validate that the system can show all equipment including its rental and request details.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as Administrator or Equipment Executive Committee.			
Test Conditions	The actor can view list of Equipment along with rentals and requests details.			
Step #	Step Details	Expected Result		
1	Navigate to "Monitor Equipment".	System display List of Equipment with its rentals and requests details,		

2.3.1.2 TC010_02: Filter Equipment by Type or Status

Table 2.19 Test Case for TC010_02: Filter Equipment by Type or Status

Test Case ID	TC010_02	Test Case Description	Ensure that equipments can be filtered by equipment type or availability.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as Administrator or Equipment Executive Committee.			
Test Conditions	The actor can filter equipments by equipment type or availability.			
Step #	Step Details	Expected Result		
1	Navigate to "Monitor Equipment".	System display List of Equipment with its rentals and requests details,		
2	Apply a filter such as "Camera" or "Available" status.	System display List of Equipment that match the filters.		

2.3.2 Test TC011: Update Equipment

2.3.2.1 TC011_01: Successfully Update Equipment Details

Table 2.20 Test Case for TC011_01: Successfully Update Equipment Details

Test Case ID	TC011_01	Test Case Description	Check that the system allows Equipment Executive Committee to update equipment information.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as or Equipment Executive Committee.		1	Main Equipment: 101
2	Equipment data exist in database		2	Sub Equipment: 301
			3	Model: 60D, Condition: Good, Status: Available
Test Conditions	Equipment Executive Committee can update equipment information.			
Step #	Step Details	Expected Result		
1	Navigate to "Manage Equipment".	System display List of Equipment with its information such as brand, model, conditions, etc.		
2	Edit brand, model, condition, or status			
3	Save new Equipment Information	System saves the newly updated equipment data in database and displays successful message.		

2.3.2.2 TC011_02: Update Equipment with Invalid Input

Table 2.21 Test Case for TC011_02: Update Equipment with Invalid Input

Test Case ID	TC011_02	Test Case Description	Check that the system rejects Equipment Executive Committee to update equipment information with invalid input	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as or Equipment Executive Committee.		1	Main Equipment: 101
2	Equipment data exist in database		2	Sub Equipment: 301
			3	Model: asd@23, Condition: Ran@^23, Status: Ran&*23
Test Conditions	Equipment Executive Committee cannot update equipment information with empty required fields or invalid data types.			
Step #	Step Details	Expected Result		
1	Navigate to "Manage Equipment".	System display List of Equipment with its information such as brand, model, conditions, etc.		
2	Enter invalid model, condition, or status details and saves it	System displays error message: "Please enter appropriate Information."		
3	Left required fields empty and saves it	System displays error message: "Please fill in the required field."		

2.3.3 Test TC012: Manage Equipment Requests and Rentals

2.3.3.1 TC012_01: View All Requests and Rentals

Table 2.22 Test Case for TC012_01: View All Requests and Rentals

Test Case ID	TC012_01	Test Case Description	Ensure the system display the entire list of equipment request and rentals.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	User is logged in as or Equipment Executive Committee.				
2	Equipment Rents and Requests exist in database				
Test Conditions	Equipment Executive Committee can view the entire list of equipment requests and rentals.				
Step #	Step Details	Expected Result			
1	Navigate to the "Manage Requests and Rentals" page.	System load and display List of Equipment Request and Rents with its information.			

2.3.3.2 TC012_02: Approve Equipment Request

Table 2.23 Test Case for TC012_02: Approve Equipment Request

Test Case ID	TC012_02	Test Case Description	Ensure the system allows Equipment Executive Committee to approve Equipment Request.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as or Equipment Executive Committee.		1	Event ID: 5001
2	Equipment Requests exist in database		2	Main Equipment: 101
			3	Sub Equipment: 301, Quantity: 2
Test Conditions	Equipment Executive Committee can approve Equipment Request.			
Step #	Step Details	Expected Result		
1	Navigate to the "Manage Requests and Rentals" page.	System load and display List of Equipment Request and Rents with its information.		
2	Click "View More"	System displays more details on Equipment Request so that user can manage equipment requested.		
3	Click "Approve".	System updates the Equipment Request status, Main Equipment status, Sub Equipment Quantities and displays successful approval message.		

2.3.3.3 TC012_03: Approve Equipment Rental

Table 2.24 Test Case for TC012_03: Approve Equipment Rental

Test Case ID	TC012_03	Test Case Description	Ensure the system allows Equipment Executive Committee to approve Equipment Rent.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as or Equipment Executive Committee.		1	Program: UTM Carnival
2	Equipment Rent exist in database		2	Location: UTM, Skudai, Johor
			3	Dates: 5–7 July
			4	Main Equipment: 101
			5	Sub Equipment: 301, Quantity: 2
Test Conditions	Equipment Executive Committee can approve Equipment Rent.			
Step #	Step Details	Expected Result		
1	Navigate to the "Manage Requests and Rentals" page.	System load and display List of Equipment Request and Rents with its information.		
2	Click "View More"	System displays more details on Equipment Rent so that user can manage equipment requested.		
3	Click "Approve".	System updates the Equipment Rent status, Main Equipment status, Sub Equipment Quantities and displays successful approval message and generates Invoice for Renter.		

2.3.4 Test TC013: Confirm Equipment Return

2.3.4.1 TC013_01: Confirm Successful Return

Table 2.25 Test Case for TC013_01: Confirm Successful Return

Test Case ID	TC013_01	Test Case Description	Ensure the system allows Equipment Executive Committee to confirms all items returned in good condition.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as or Equipment Executive Committee.		1	Main Equipment: 101
2	Equipment Rent or Request exist in database		2	Sub Equipment: 301, Quantity: 2
3	Equipment Rental or Request has been approved			
Test Conditions	Equipment Executive Committee can confirm borrowed equipment return and system shall restore Main Equipment and Sub Equipment Inventory.			
Step #	Step Details	Expected Result		
1	Navigate to the "Manage Equipment Return" page.	System load and display List of Equipment Rents and Request and its returning equipment.		
2	Click "View More"	System displays more details on returning Equipment for each rent or request.		
3	Click "Approve".	System updates the Equipment Request or Rent status, Main Equipment status, Sub Equipment Quantities and displays successful message.		

2.3.4.2 TC013_02: Return with Missing Equipment for Rent

Table 2.26 Test Case for TC013_02: Return with Missing Equipment for Rent

Test Case ID	TC013_02	Test Case Description	Ensure the system allows Equipment Executive Committee to confirms missing items returned than rented.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	User is logged in as or Equipment Executive Committee.		1	Sub Equipment: 301, Quantity: 2
2	Equipment Rent exist in database			
3	Equipment Rent has been approved			
Test Conditions	Equipment Executive Committee can confirm missing items returned than rented and system shall calculate the punishment fee based on the missing items.			
Step #	Step Details	Expected Result		
1	Navigate to the "Manage Equipment Return" page.	System load and display List of Equipment Rents and Request and its returning equipment.		
2	Click "View More"	System displays more details on returning Equipment for each rent or request.		
3	Uncheck the missing items.	System calculates the punishment fee		
3	Click "Confirm".	System updates the Equipment Request or Rent status, Main Equipment status, Sub Equipment Quantities and displays successful message. System sends notifies renter to pay the punishment fee.		

2.4 Role and Event Management Module

2.4.1 Test TC014: Manage Members Roles

2.4.1.1 TC014_01: View All Users and Their Roles

Table 2.27 Test Case for TC014_01: View All Users and Their Roles

Test Case ID	TC014_01	Test Case Description	Ensure the system has display all the users including their current set roles.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Admin is logged into an account.		1	userId: 001, name: Ali, roles: [Administrator]
2	User and role data exists.		2	userId: 002, name: Sara, roles: [Equipment Committee Member]
Test Conditions	The system should identify and display users and roles assigned to the users correctly.			
Step #	Step Details	Expected Result		
1	Login as Admin (userId: 1).	User logged in as Administrator.		
2	Navigate to "Manage Member Roles".	System displays both users with their assigned role.		

2.4.1.2 TC014_02: Assign Role to a User

Table 2.28 Test Case for TC014_02: Assign Role to a User

Test Case ID	TC014_02	Test Case Description	Check that whether an admin can assign a user a new role.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Admin is logged into an account.		1	userId: 3, name: Amin, roles: [User]
2	User and role data exists.		2	Role to assign: Equipment Executive Committee
Test Conditions	User should have the role added to user's role list.			
Step #	Step Details	Expected Result		
1	Login as Admin (userId: 1).	User logged in as Administrator.		
2	Navigate to "Manage Member Roles".	System displays both users with their assigned role.		
3	Select user Amin.			
4	Select role: Equipment Executive Committee.			
5	Click "Assign".	System updates Amin role: [Member, Equipment Executive Committee].		

2.4.1.3 TC014_03: Remove Role from a User

Table 2.29 Test Case for TC014_03: Remove Role from a User

Test Case ID	TC014_03	Test Case Description	Ensure that an admin can delete a role of a user.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Administrator is logged into an account.		1	userId: 4, name: Farah, roles: [User, Equipment Committee Member]
2	User has multiple roles.		2	Role to remove: Equipment Committee Member
Test Conditions	Selected role should be removed from a user.			
Step #	Step Details	Expected Result		
1	Login as Admin (userId: 1).	User logged in as Administrator.		
2	Navigate to "Manage Member Roles".	System displays list of users with their assigned role.		
3	Select user Farah.			
4	Click "Remove" on Equipment Committee Member.	System updates Farah role: [User]		

2.4.2 Test TC015: Manage Event

2.4.2.1 TC015_01: Create a New Event

Table 2.30 Test Case for TC015_01: Create a New Event

Test Case ID	TC015_01	Test Case Description	Ensure that a user with a role of an administrator can create a new event.		
Created by:	Lio Kock Hock	Version:	1.0		
No	Prerequisites		No.	Test Data	
1	Administrator is logged into an account.		1	Event Name: Photography Workshop	
			2	Description: Basic DSLR Training	
			3	Start Date: 2025-08-15, End Date: 2025-08-17	
			4	Location: Dewan Sri Resak	
			5	Organizer: userId: 2	
Test Conditions	Administrator can create a new event.				
Step #	Step Details	Expected Result			
1	Navigate to "Manage Events".	System displays list of events.			
2	Click "Create Event".				
3	Fill in the event form and submit.	System saves newly created event and displays successful message.			

2.4.2.2 TC015_02: Edit an Existing Event

Table 2.31 Test Case for TC015_02: Edit an Existing Event

Test Case ID	TC015_02	Test Case Description	Ensure that an administrator has ability to edit a current event.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Administrator is logged into an account.		1	Event ID: 5001
2	Event already exists.		2	New Location: Dewan Kuliah 2
Test Conditions	Administrator can update an event and system save it in database.			
Step #	Step Details	Expected Result		
1	Navigate to "Manage Events".	System displays list of events.		
2	Click "Edit" for event ID 5001.			
3	Update location and save changes	System saves newly updated event and displays successful message.		

2.4.2.3 TC015_03: Assign Equipment Committee to Event

Table 2.32 Test Case for TC015_03: Assign Equipment Committee to Event

Test Case ID	TC015_03	Test Case Description	Ensure that it is possible to assign Equipment Committee Members for an event.	
Created by:	Lio Kock Hock	Version:	1.0	
No	Prerequisites		No.	Test Data
1	Administrator is logged into an account.		1	Event ID: 5002
2	Event data exists.		2	Equipment Committee Members: userId: 3, 4
3	Users' data exists.			
Test Conditions	Administrator can add new Equipment Committee member to an event.			
Step #	Step Details	Expected Result		
1	Navigate to "Manage Events".	System displays list of events.		
2	Click "Edit" for event ID 5002.			
3	Click "Assign Committee" and select users			
4	Click "save changes"	System saves newly added Equipment Committee Member and displays successful message.		