# Software Architecture Documentation

Enterprise Resource Planning System
Bicycle Manufacturing
Soen 390
Winter 2021

# Contents

# 1 Introduction

The purpose of this document is to give detailed architecture design and it's constraint of the ERP bicycle manufacturing system. Software architect documents of ERP bicycle manufacturing systems will be evaluated based on various quality attributes such as usability, maintainability, availability and testability.

The objective of the software document is to give ideas for the development team how the system will be structured at the high level. Also project evaluators can use this document for validating development team implementation.

## 1.2 Supplementary information

ERP bicycle manufacturing system is a software which helps business processes by providing functionality like accounting, sales, production, inventory, procurement, packaging and many more.

Manufacturing ERP systems provide support for production planning, machine workload, scheduling by analyzing various data in the system. There are different type of ERP software:

- On-premise ERP: ERP software deployed directly in-site devices.
- Open source ERP: Same as other open source software which can be modifiable.
- Cloud-based ERP: ERP software-as-a-service(SaaS) which is available over cloud-hosting service not on-site.

Benefit of ERP manufacturing:

- Workflow visibility.
- Data analytics.
- Cross-Department collaboration.
- Risk management.
- Data Security.

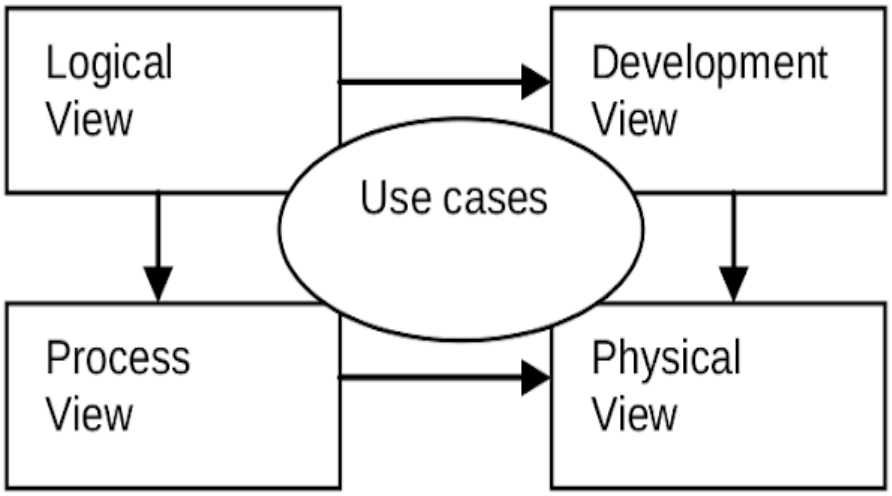
## 1.3 Other information

**Revision History**

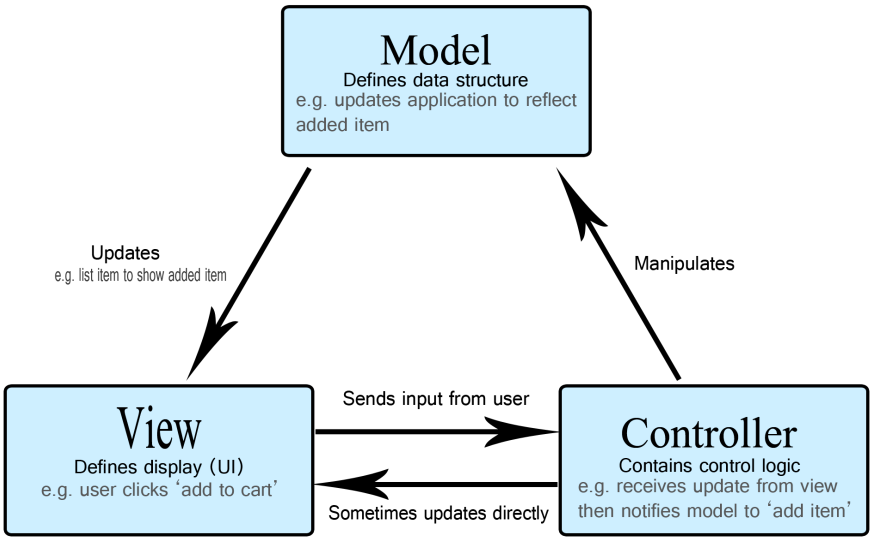
| Date | Revision | Description | Author(s) |
|---|---|---|---|
| 31-01-2021 | 1.0 | Initial version | Muhammad Shah Newaz, Julien Xu |
| 24-02-2021 | 1.1 | Updated domain model, added class diagram, added ER diagram. | Muhammad Shah Newaz, Julien Xu |
| 15-03-2021 | 1.2 | Updated class diagram, ER diagram, use case diagram, sequence diagram | Muhammad Shah Newaz, Julien Xu |
| 07-04-2021 | 1.3 | Updated class diagram, use case diagram, ER diagram | Muhammad Shah Newaz, Julien Xu |

### 1.3.1 Overview

The ERP system architect document has been analyzed using Kruchten's 4+1 architectural view model.




ERP bicycle manufacturing system is built using Model-View-Controller architectural design pattern.




### 1.3.2 Architecture evaluations

N/A

### 1.3.3 Rationale for key decisions

Key Decision:

ERP bicycle manufacturing system shall be built using MVC(Model-View-Controller) architectural design pattern.

Owner of the decision:

Development team of the system.

Affecting Key Stakeholder:

System developer, System maintenance team, System testing team.

Rationale for the key decision:

- MVC is ideal for developing complex applications.
- It is an extensible and pluggable framework which can be easily replaced and customized.
- Uses component based design of the ERP application which is logically divided into Model, View and Controller components.

# 2 Stakeholders and concerns

## 2.1 Stakeholders

Users
The users of the ERP system consist of company employees that are in charge of managing and supervising the system. There are different levels of users, the highest one being the admin. They are an internal stakeholder.

Operators
The operators consist of HR users who are in charge of planning and scheduling. They are an internal stakeholder.

Acquirers
The acquirers are the purchasers of the product. They consist mostly of consumers. They are an external stakeholder.

Owners
The owners are the group of people who possess the rights to the manufactured product, represented by the Product Owner. They are an external stakeholder.

Suppliers
The suppliers are the group of people who sell the company the necessary product parts and materials. They consist mostly of vendors. They are an external stakeholder.

Builders
The builders are the group of people in charge of creating and managing the structure of the ERP system. They are an internal stakeholder.

Developers
The developers are the group of people in charge of creating the software for the ERP system. They are also the one who design the UI and implement new features to the existing software. They are an internal stakeholder.

Maintainers
The maintainers represent the group of people that fixes bugs and runs maintenance on the software. They are the ones who update the software to the latest version during deployment. They are an internal stakeholder.

## 2.2 Concerns

1. Does the ERP system use encapsulation in order to store information safely?
2. Is the ERP system reliable?
3. How is the ERP system to be maintained and evolved?
4. What is the suitability of the architecture for achieving the ERP system's purposes?
5. How feasible is it to construct and deploy the ERP system?
6. What are the potential risks and impacts of the ERP system to its stakeholders throughout its life cycle?

## 2.3 Concern–Stakeholder Traceability

| | Users | Operators | Acquirers | Owners | Suppliers | Builders | Developers | Maintainers |
|---|---|---|---|---|---|---|---|---|
| Concern 1 | x | x | | | | | | |
| Concern 2 | x | x | x | x | | | | |
| Concern 3 | | | | | | | x | x |
| Concern 4 | | | | | | x | | |
| Concern 5 | | | | | | x | | x |
| Concern 6 | | | x | x | x | | | |

# 3 Viewpoints +

### 3.1.1 Software Developer

### 3.1.2 Overview

Software developer task to build ERP systems which is easily maintainable, testable and available all the time. Also software developers need to build a system which meets stakeholder requirements. In order to build an efficient ERP manufacturing system software developer needs to understand and analyze the requirements. Software developer viewpoints have multiple views which helps them understand their requirements and also visualize the system to build.

### 3.1.3 Concerns and stakeholders

#### 3.1.3.1 Concerns

- Understand the difficulty of the system.
- What kind of architecture should deal with the system.
- What are the conceptual classes of the system?
- Finding various objects in the system.
- Finding relation and multiplicity between different conceptual classes.
- Maintainability of the software architecture.
- Testability of the software system.

#### 3.1.3.2 Typical stakeholders

- Developer or programmer of the ERP system.
- Tester of the ERP system.
- Maintainer of the ERP system.
- System engineers of the ERP system.

## 3.1.4 Model kinds +

## 3.1.5 UML Structural Diagrams.

#### 3.1.5.1 UML structural diagrams conventions

- Object or conceptual class identified by noun phrase.
- Object relation is shown by lines, arrow lines and dotted lines.
- Uses actors of the system.
- Multiplicity of the object relation.
- Relations are described by verb phrases.

#### 3.1.5.2 correspondence rules

- Class diagram.
- Domain model.
- Component diagram.
- Deployment diagram.
- Package diagram,

## 3.1.6 Operations on views

- Object relations are created using domain models.
- Creates object structure.
- Define relation between various objects.
- Define type of objects in the ERP system.
- Define architure of the ERP system.
- Define data processing mechanism.

## 3.1.7 Correspondence rules

- Logical or Conceptual view is described by domain model, class diagram.
- Physical view is described by the Deployment diagram.
- Development view describe Component and package diagram

## 3.1.8 Sources


- Software System Architecture,Second Edition, Nick Rozanski, Eion Woods.
- Kruchten's 4+1 View Model.

### 3.2.1 End Users

### 3.2.2 Overview

ERP manufacturing system is used by various users in the company in order to accomplish different tasks. In the end, using a viewpoint helps to define tasks that can be done using different features of the system. In this viewpoint, the process of the task to be accomplished is also shown in diagrams.

### 3.2.3 Concerns and stakeholders

#### 3.2.3.1 Concerns

- Understand the difficulty of the system.
- Which feature need to use for desired task
- What are the conceptual classes of the system?
- System can create new accounts efficiently.
- After entering new data into the system is updated in the correspondence features.

#### 3.2.3.2 Typical stakeholders

- IT specialist.
- HR manager.
- Production supervisor.
- Sales manager.
- Product planning personal.

## 3.2.4 Model kinds +

### 3.2.5 UML Behavioral Diagrams.

#### 3.2.5.1 UML behavioral diagrams conventions

- Primary actor.
- Secondary actor.
- System boundary.
- <<include>> relations.
- <<exclude>> relations.

#### 3.2.5.2 correspondence rules

- Use-Case diagram.
- State Machine diagram.
- Activity Diagram.

### 3.2.6 Operations on views

- Define actor goals.
- Define actors not included in the relationship.
- Provide guidance of the system.
- Define multiple paths in the activity.

### 3.2.7 Correspondence rules

- Use case view described by the use case diagram.

### 3.2.8 Sources

- Software System Architecture,Second Edition, Nick Rozanski,Eion Woods.
- Kruchten's 4+1 View Model.

# 4 Views+

## 4.1 View: Use Case

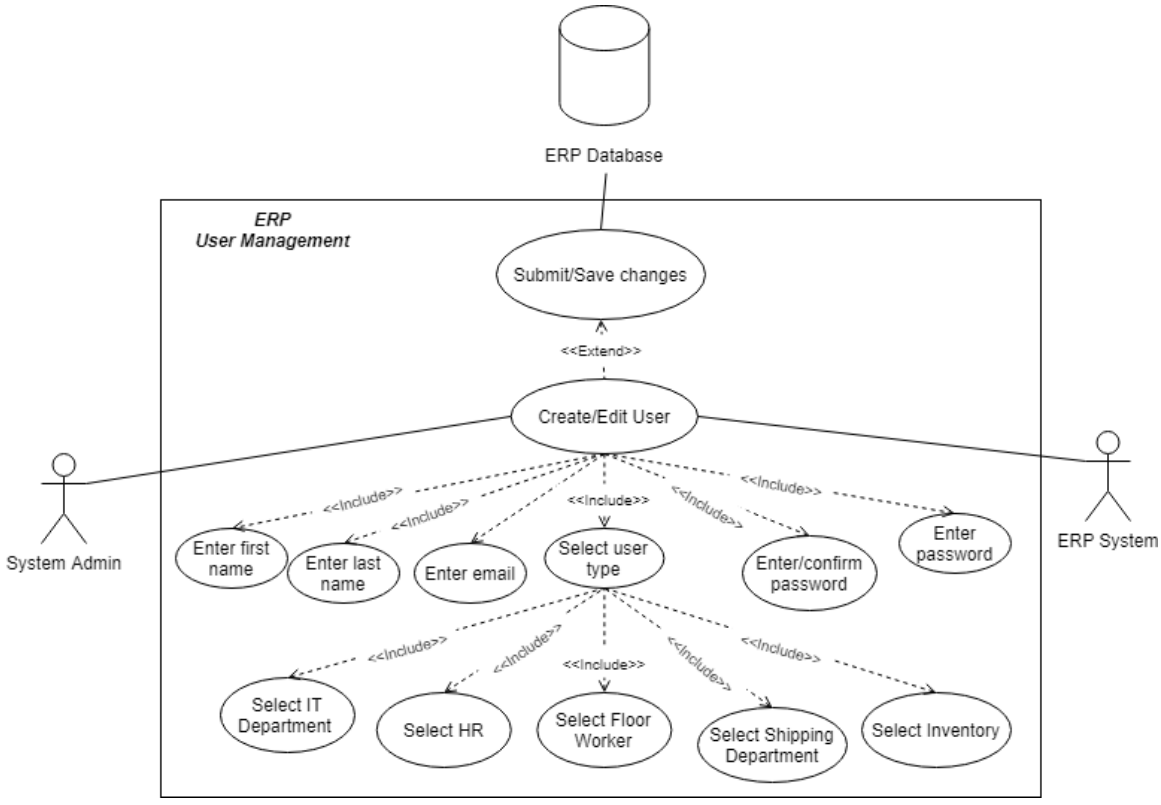
Use Case view describes the relation between end-users and customers of the ERP system. It also describes connection between functionalities with their internal or external controller. Controllers are known as the actors of the system in the Use case view.

### 4.1.1 Models +

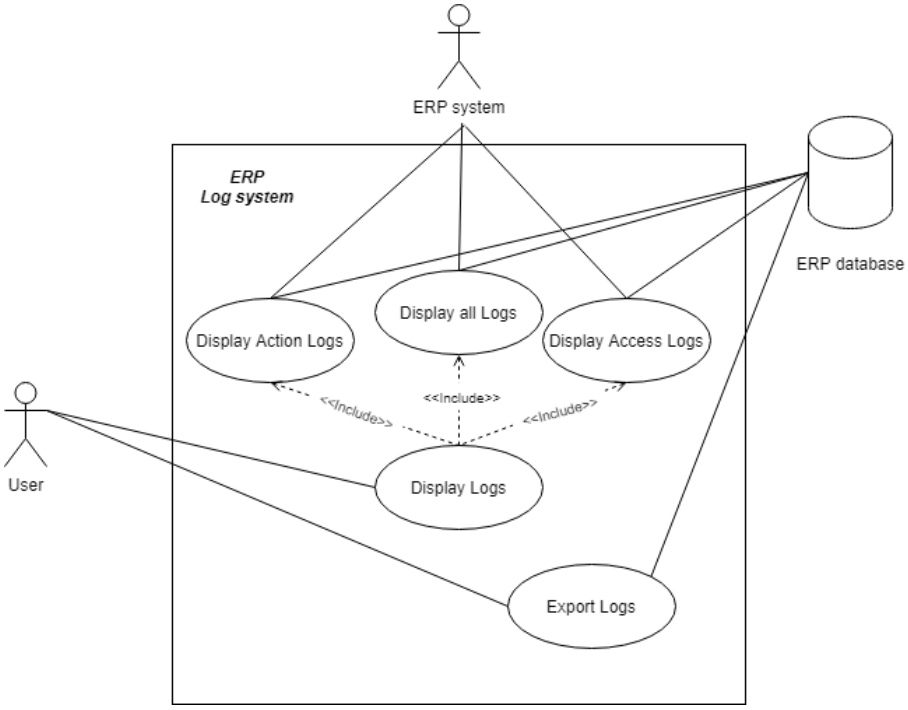
- Use Case diagram

### 4.1.2 Use case model

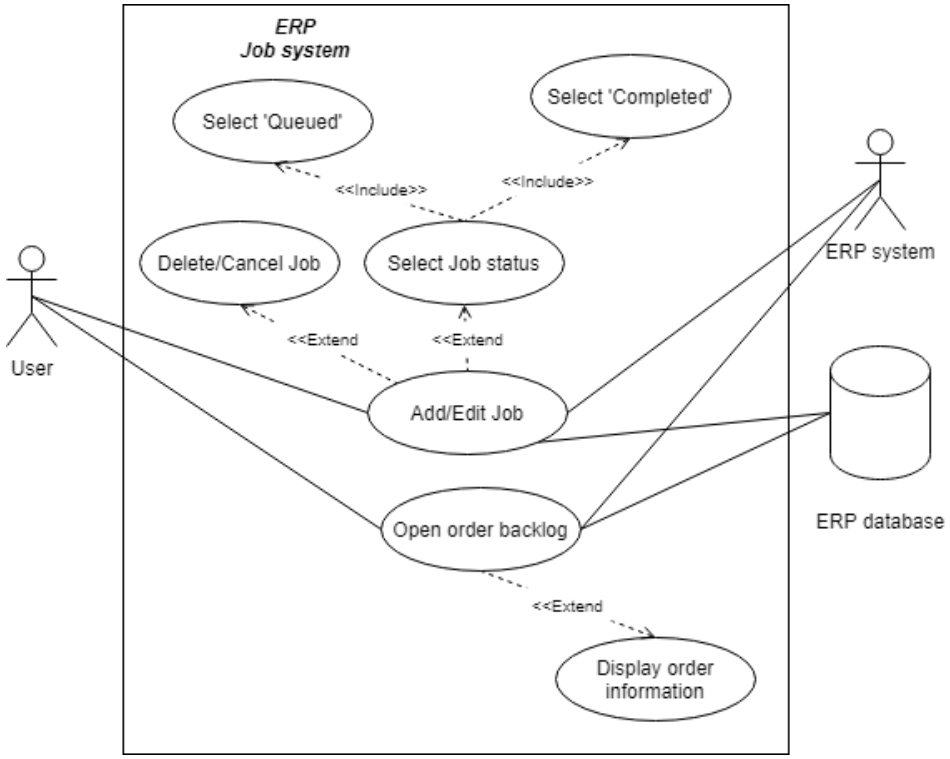
User Management:



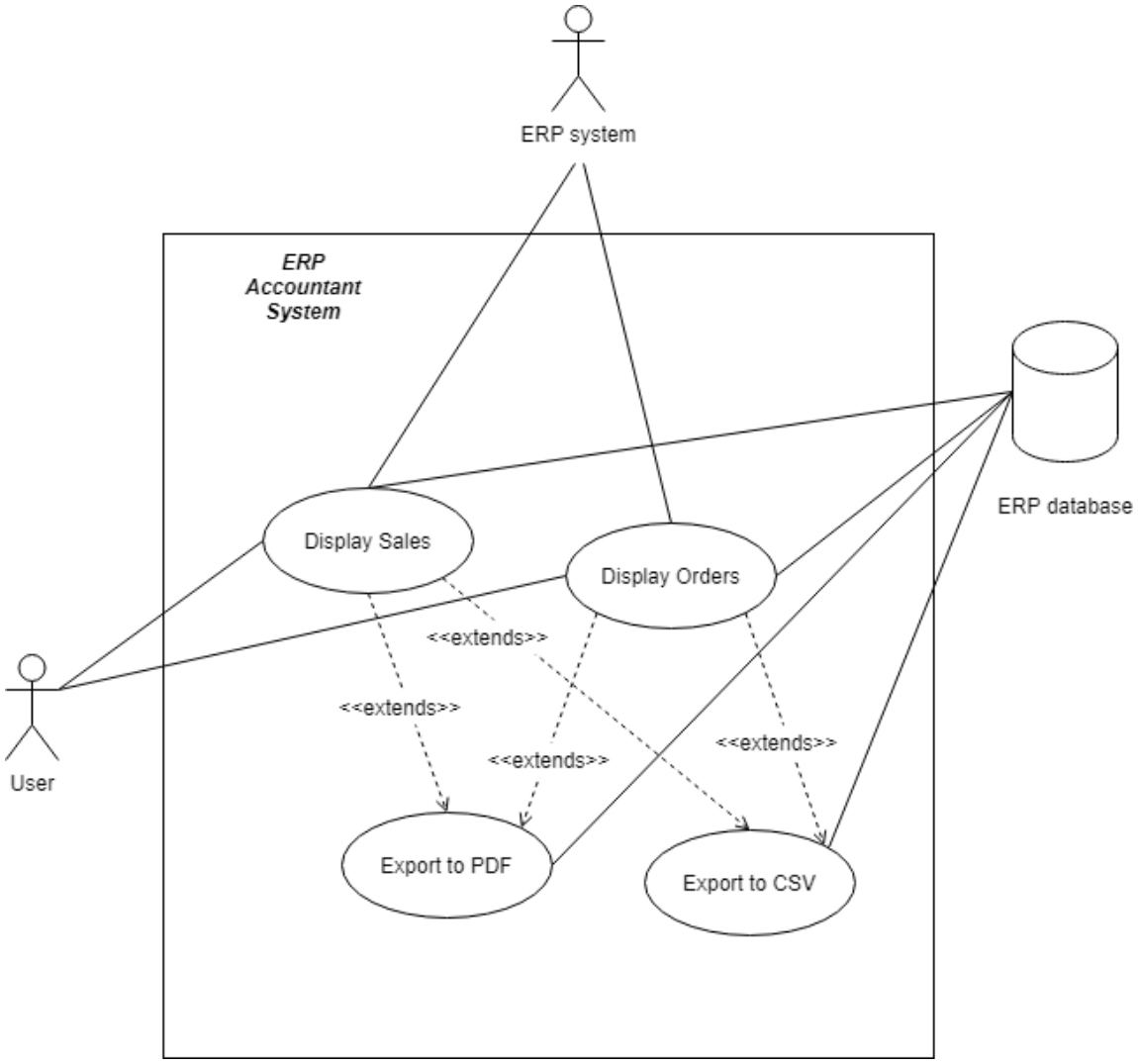
Logs:


ERP system
ERP
Log system
ERP database
Display all Logs
Display Action Logs
Display Access Logs
<<Include>>
<<Include>>
<<Include>>
User
Display Logs
Export Logs

Jobs:


ERP
Job system
Select 'Queued'
Select 'Completed'
<<Include>>
<<Include>>
ERP system
Delete/Cancel Job
Select Job status
<<Extend
<<Extend
User
Add/Edit Job
ERP database
Open order backlog
<<Extend
Display order
information

Sales:




### 4.1.3 Known Issues with View

N/A

## 4.2 View: Logical/ Conceptual

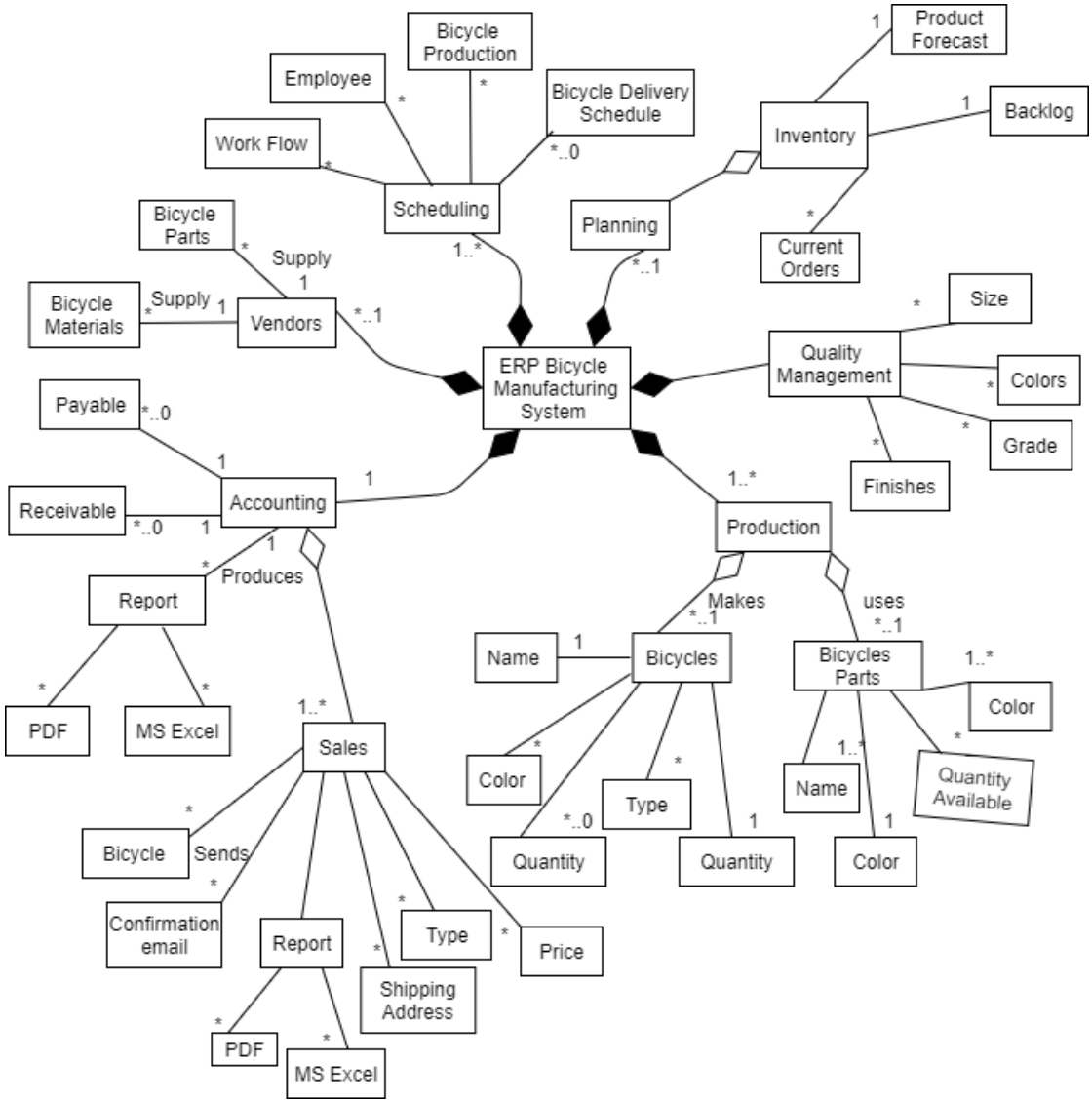
Logical view describes the relation between objects and structure of the object. It also defines different types of objects in the system and their role in the system. Logical view deals with the functional requirements of the system.

### 4.2.1 Models +

- Domain model
- Class diagram

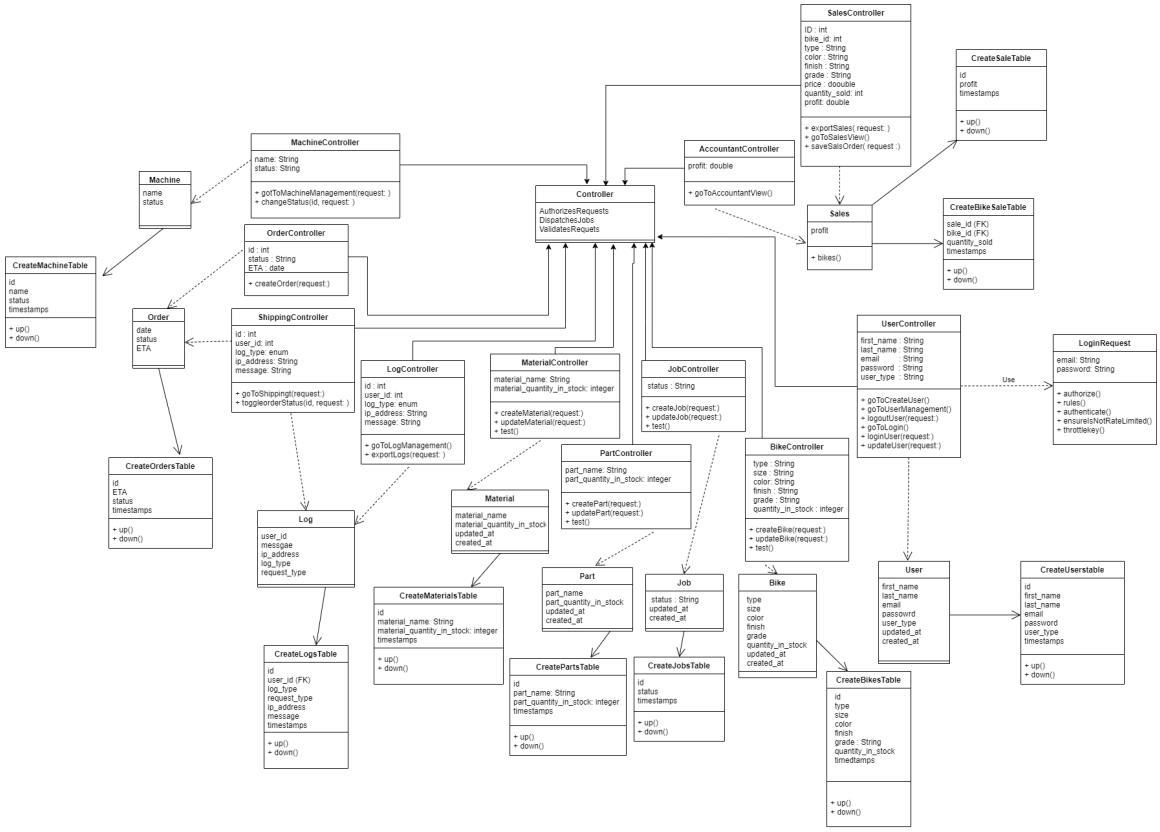
### 4.2.2 Domain model

Domain model diagram:



### 4.2.3 Class model

Class diagram:




### 4.2.4 Known Issues with View

N/A

## 4.3 View: Physical

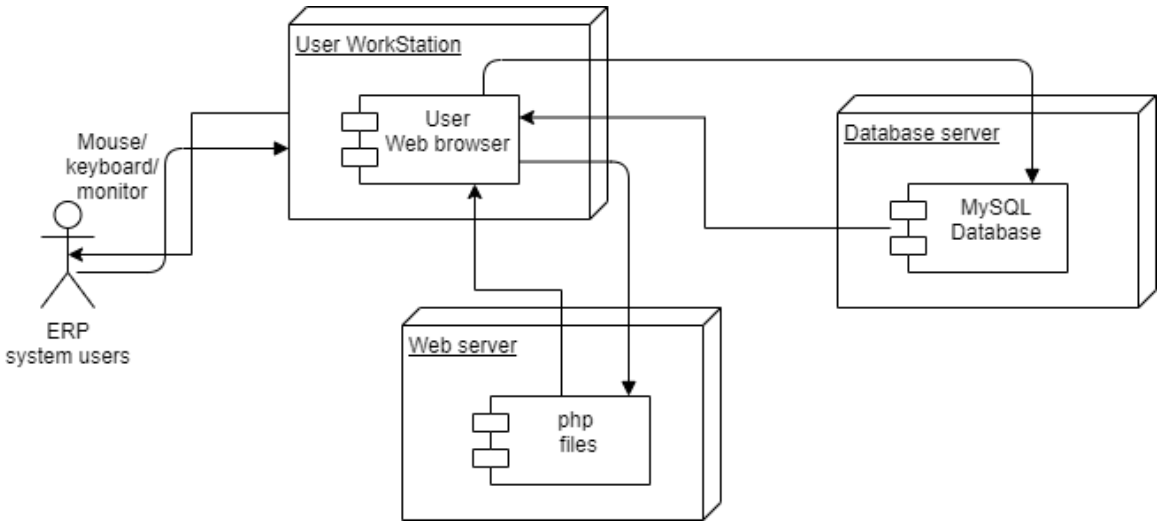
Physical view deals with the physical component of the system and intere connection between the components. This view describes a system from the point of view of a system engineer's view. Physical view is also known as the deployment view.

### 4.3.1 Models +

- Deployment diagram.

### 4.3.2 Deployment model.

Deployment Diagram:




### 4.3.3 Known Issues with View

N/A

## 4.4 View: Development

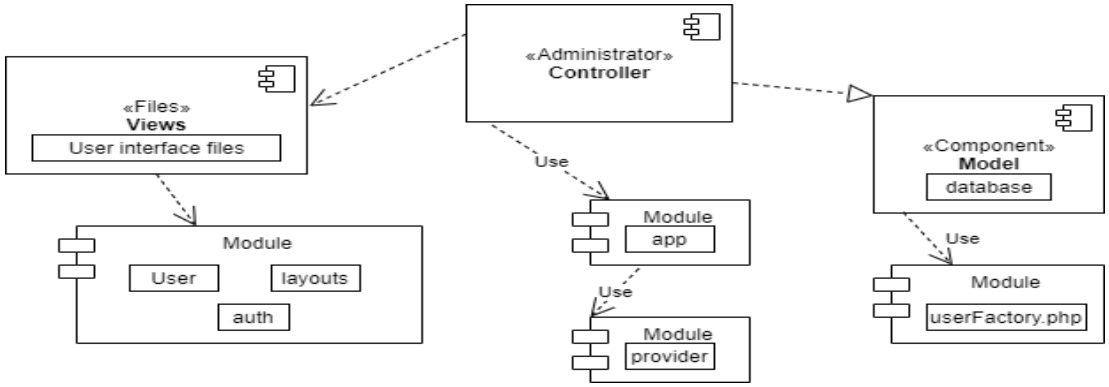
This view describes the perspective of developer or programmer viewpoint. Development view is also known as implementation view. It describes packages, files of the system.

### 4.4.1 Models +

- Component diagram.
- Package diagram.

### 4.4.2 Component Model

Component Diagram:



### 4.4.3 Known Issues with View

N/A.

## 4.5 View: Process View

Process view described in the developer or programmer viewpoint. This view focuses on dynamic aspects of the system. This view explains different processes at the runtime of ERP systems.

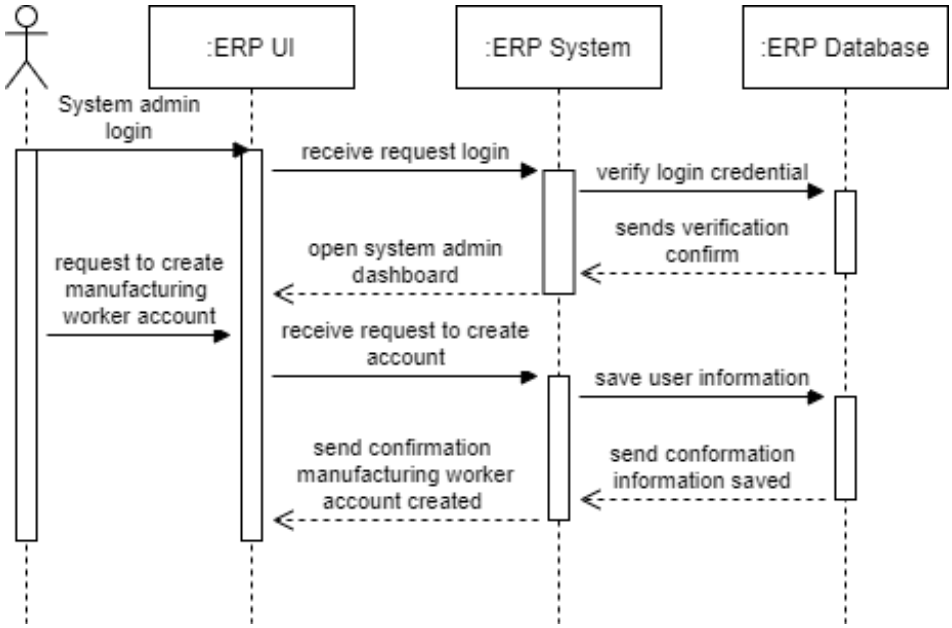
### 4.5.1 Models +

- Sequence diagram.

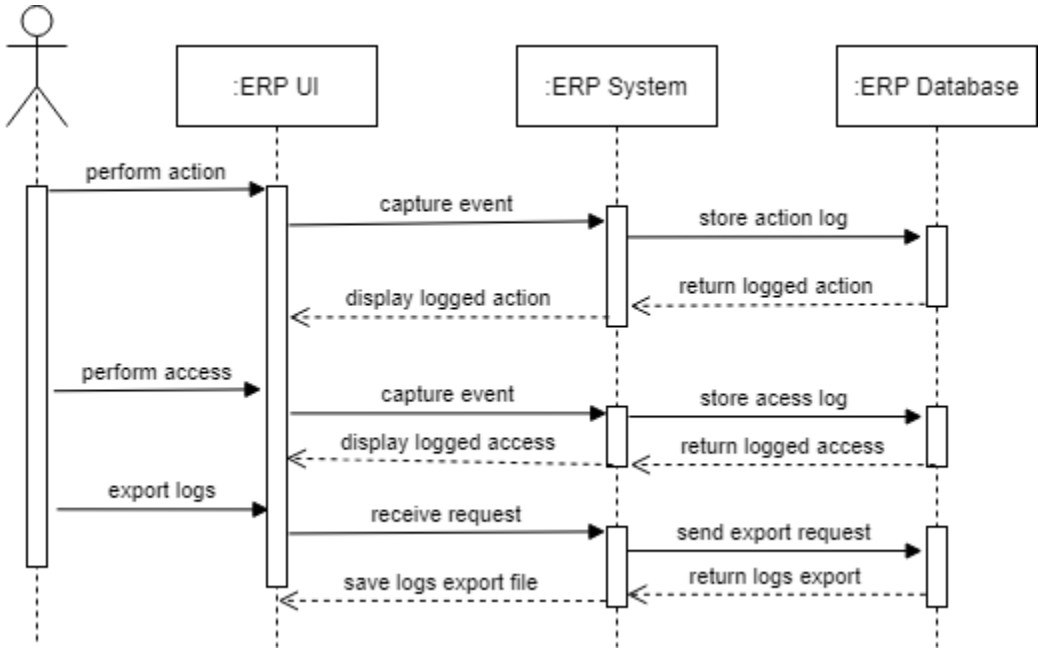
### 4.5.2 Sequence model

Sequence Diagrams:

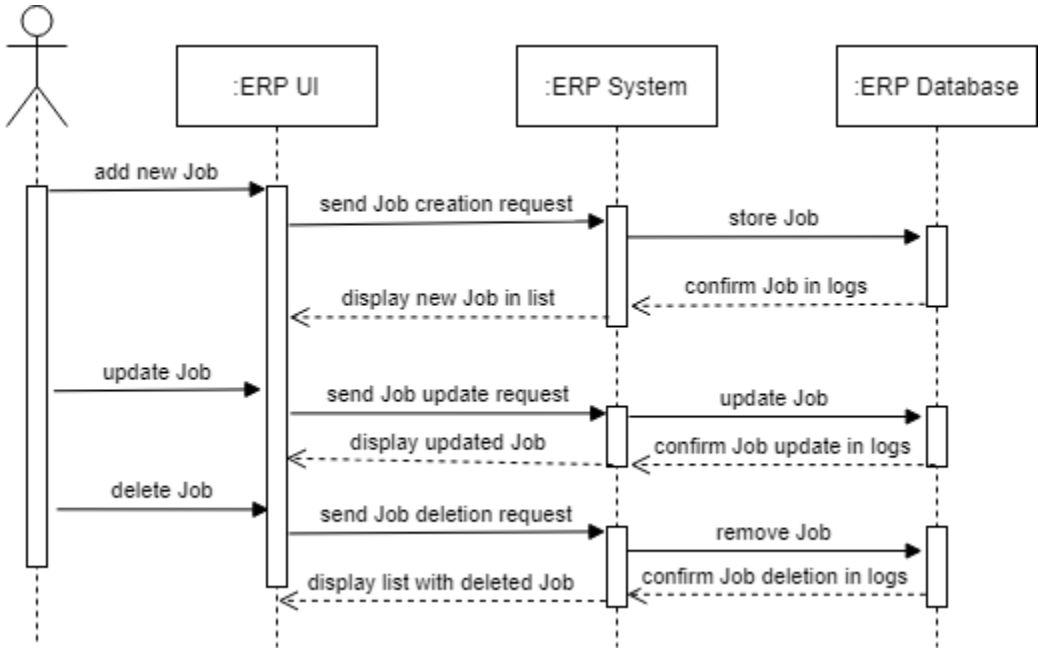
User login/creation:



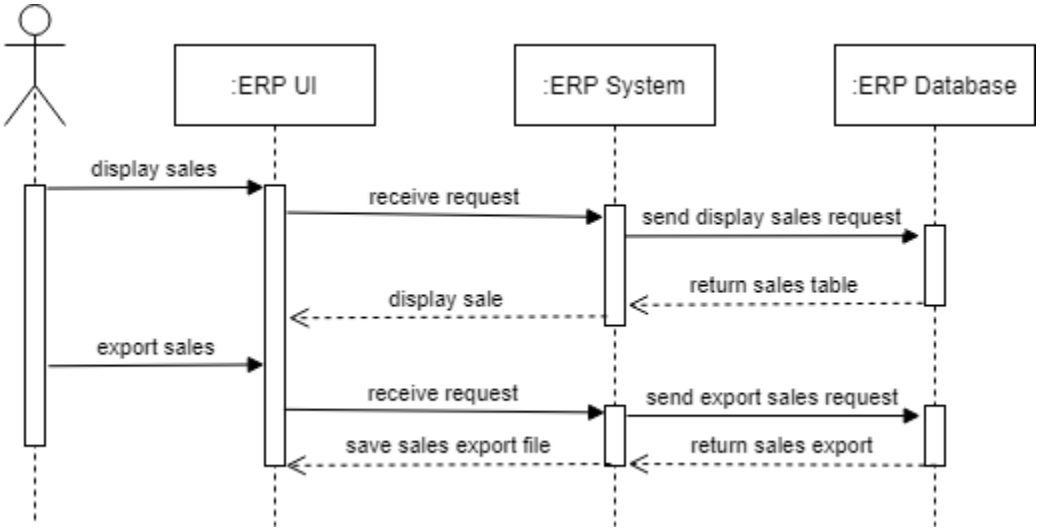
Logs:




Create/update/delete job:



Sales:




### 4.5.3 Known Issues with View

N/A

## 4.6 View: Data View

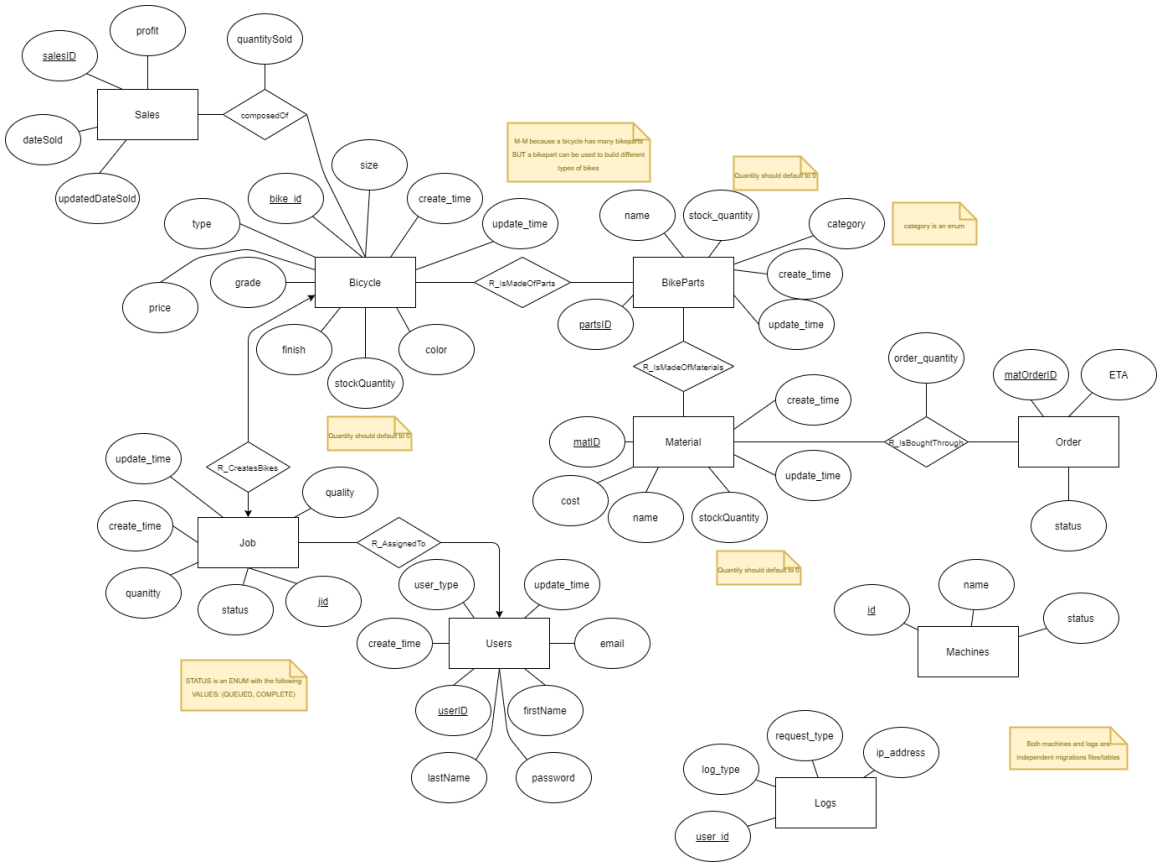
Data View is the structure of the data or visual representation of the data which is different from the physical representation of the data. This view helps make data more readable.

### 4.6.1 Model+

- ER diagram.

### 4.6.2 Data Model

ER diagram:




### 4.6.3 Known Issues with View N/A.

# 5 Consistency and correspondences

This chapter describes consistency requirements, recording of known inconsistencies in an AD, and the use and documentation of correspondences and correspondence rules.

## 5.1 Known inconsistencies

Due to early development, there have been some inconsistencies in the AD, mainly in the stakeholder concerns. The concerns are mainly speculatory as well as the effects on the stakeholders. In order to obtain a better projection of the concerns, the ERP system will need to go through more phases of development and testing.

The Logical/Conceptual view is more developed than other views because there isn't enough information given at the current time to generate the specific details of the way the features will be implemented. Every view describes an early concept of the ERP system and may be subject to changes in the future.

## 5.2 Correspondences in the AD

The Logical/Conceptual view of the architecture is a view that showcases the features and requirements of the ERP system. It is a view typically used by stakeholders such as owners and builders of the system and is represented by a domain model or a class diagram. A domain model showcases the relations between each module in the system while the class diagram showcases the relations between each class in the code.

The Physical view of the architecture is a view that represents the operations of the ERP system in a real and concrete way. It is a view typically used by stakeholders such as users and operators of the system and is represented by a Deployment diagram.

The Development view of the architecture is a view that represents the operations of the ERP system in a software oriented way. It is a view typically used by stakeholders such as developers and maintainers of the system and is represented by a Component and Package diagram.

The Use Case view of the architecture is a view that represents the front-end and back-end relations of the ERP system. It is a view typically used by stakeholders such as users and operators of the system and is represented by a Use Case diagram.

The process view of the architecture is a view that represents the actions taken by certain modules when the ERP system is performing actions. It is a view typically used by stakeholders such as developers and maintainers of the system and is represented by a Sequence diagram.

The data view of the architecture is a view that represents the relations between the tables in the database of the system. It is a view typically used by stakeholders such as developers and maintainers of the system and is represented by an ER diagram.

## 5.3 Correspondence rules

1. Logical or Conceptual view is described by Domain Model, Class diagram. (satisfied)
2. Physical view is described by the Deployment diagram. (satisfied)
3. Development view is described by the Component and Package diagram (satisfied)
4. Use case view described by the Use Case diagram. (satisfied)
5. Process view described by the Sequence diagram. (satisfied)
6. Data view described by the ER diagram. (satisfied)