



*Operating Lease Support System Development*

**For**

******

***Functional Specification***

***A.2 Maintenance Module***

Revision 1.0

**Prepared by:**

**PT. Berlian Sistem Informasi**

KTB Annex Bld. 4th Floor, Jl. Jend. A. Yani. Pulomas,

Tel. +62 21 4786 7575 (Hunting) Fax. +62 21 4714964 [www.bsi.co.id](http://www.bsi.co.id)

Document Control

|  |  |
| --- | --- |
| **Author** | Deny Prasetia |
| **File Name** | TM14402A - DSF OLSS - FS A.1 Marketing Module Rev.1.0 |
| **Path** | N/A |
| **Create Date** | 25/11/2014 11:02:00 |
| **Last Edited** | 06/01/2015 15:19:00 |
| **Number of Pages** | 15 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 25-Nov-2014 | 0.1 | Initial creation | Deny Prasetia |
| 10-Dec-2014 | 1.0 | Baseline | Deny Prasetia |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Organization** | **Title** |
| Mr. Jin Nishimura | Dipo Star Finance, PT | DSF Project Director |
| Mr. Abdul Somad | Dipo Star Finance, PT | DSF Project Manager |
| Mr. Satoshi Koibuchi | Berlian Sistem Informasi, PT | BSI Project Director |
| Mr. Deny Prasetia | Berlian Sistem Informasi, PT | BSI Project Manager |

Document Approval

By signing this document I acknowledge I have read the document and give the Project Management Team approval to proceed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature** | **Date** |
| DSF Project Director | Mr. Jin Nishimura |  |  |
| DSF Project Manager | Mr. Abdul Somad |  |  |
| BSI Project Director | Mr. Satoshi Koibuchi |  |  |
| BSI Project Manager | Mr. Deny Prasetia |  |  |

Table of Contents

Document Control 2

Revision History 2

Distribution List 2

Document Approval 2

1 Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Acronyms and abbreviations 5

1.4 References 5

2 Detail Specification 6

2.1 Service History Inquiry 6

**2.1.1** **Use case** 6

**2.1.2** **Operations and scenarios** 6

**2.1.3** **Status management and roles** 9

**2.1.4** **Sitemap design** 9

**2.1.5** **Screen design** 10

**2.1.5.1** **Service history inquiry** 10

**2.1.5.2** **Add new service history records** 11

**2.1.5.3** **Update service history records** 12

**2.1.6** **Screen features** 13

**2.1.7** **Data structure** 13

3 Appendix 15

3.1 Use Case Terminology 15

1. Introduction
   1. Purpose

Functional specification is to explain some of the following:

1. Describe who uses your application or system,
2. Describe the order in which steps or events or on the scenario are performed,
3. Show user interface design, and
4. What kind of data that will be maintained by functions.
   1. Scope

Marketing module described about how to maintain data transaction on the marketing process which is utilized by functions in the system. On this module there’s no integration with another system (such as ACA and MFAPPL). This modules deal in the “to-be design” for developing the new system.

To keep the accuracy of entire design of the system we are using modeling approach for design and analysis. ***To define a model of functions we will determine use cases. In the use case, it can summarize who uses your application or system, and what they can do with it.***

Within a use case we can determine which use cases are supported by application. We can create use case diagram, activity diagram, use case scenario, data structure, and interface design.

A data structure can be specified what kind of information that will be required for the system. Data structure is represented database design of the system. A user interface design can imagine the user about the system as visual. User interface is designed based on prototypes which have been already created. And, based on these things, we will develop the system.

The following is functionality of Maintenance module:

* Service History Inquiry

1. Add new service history records

To input a new service history records by fill out the form. This feature allows a user to save a data into database, after input on the service history record form.

1. Update details service history records

To update a detail service history records by fill out the details form. This feature allows a user to changes any data, and then saves into database.

1. Filter and sort agreement information

On this feature the record can be filtered and sorted based on columns on top of the list.

1. Delete service history records

Delete a record of service history on the list. Only status draft can be deleted.

1. Validate service history records

After the records was confirm, the record can be to validate.

1. Set to draft service history records

If the data still need to be update by user who initiate to create the agreement, the records can be set to draft.

* 1. Acronyms and abbreviations
* ACA : Auto Credit Approval System
* MFAPPL : Multi Finance Application
  1. References

This functional scope refers “to be design” in the section A.1.Marketing Module.

1. Detail Specification
   1. Service History Inquiry
      1. **Use case**

The figure below is summarizing who uses features of function, and what they can do with it.

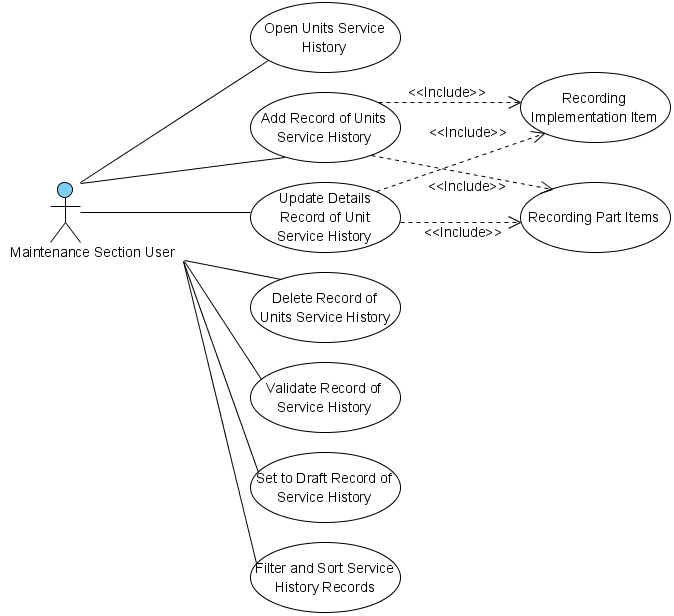


Figure 1 – Use case diagram of Service History Inquiry

* + 1. **Operations and scenarios**

The figure below is describing operations or steps performed in a function interact with people (flow of work between actors and the system).

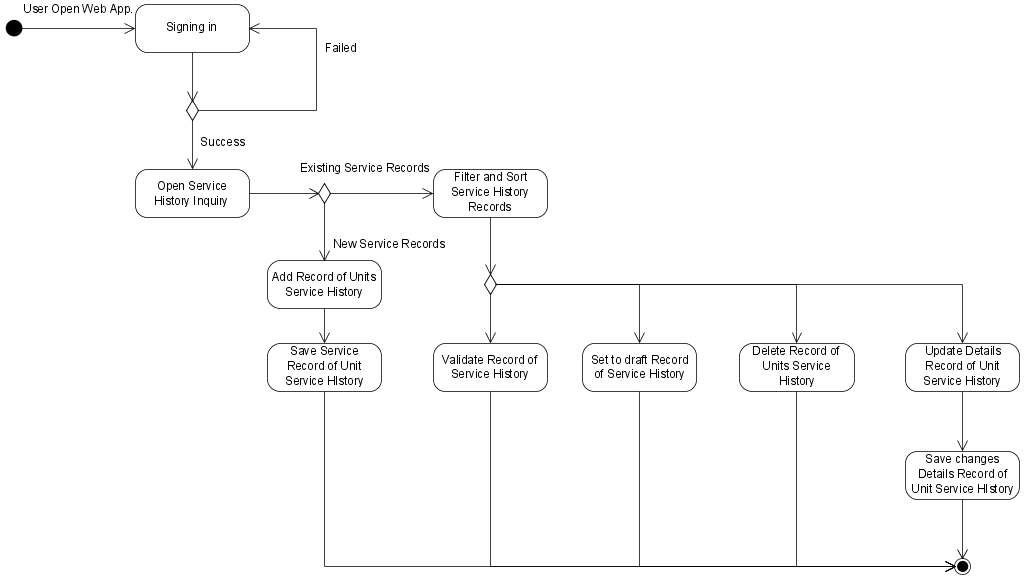


Figure 2 – Activity diagram of Service History Inquiry

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Steps/ Operations** | **Actor/ People** | **Scenario** |
| 1 | Signing in | Maintenance Section User | * Actor fills out sign in form by inputting user name and password. * Actor clicks [sign in] button. System will check the account of user; if user account is already registered user will enter into the system. If doesn’t user can return the step. |
| 2. | Open service history inquiry module | Maintenance Section User | * After login, actor can open service history inquiry module on the menu of system. After that, screen of service history inquiry will open. |
| 3. | Add new service history record | Maintenance Section User | * If service history screen opens, actor can determined the agreement number and police number that want to be add the service history record. * Upon clicking […] button on the agreement number fields, the agreement look-up screen will be displayed. Agreement number can be selected (only one record can be select). After it’s done, agreement information field will be filled automatically. * After determined agreement number, actor can determine the police number by clicking […] button on the police number field. The Units (Object Lease) look-up screen will be displayed. Police number can be selected (only one record can be select). After it’s done, unit’s information field will be filled automatically. * After determining agreement number and police number, user can click [Add] button to add new service history record. Once it done, ‘Add’ screen will be displayed. * On ‘Add’ screen open, general information of service can be views. To input service history details, actor should input several fields, such as; type of service, service date, actual mileage and workshop. Once it done, service record history can be saved as draft by clicking [Save as Draft] button. * If service history records already drafted, actor can add main implementation item and parts item by clicking [Add] button, and then input the data’s. * The mandatory fields cannot be blank. |
| 4. | Save service history records as draft | Maintenance Section User | * After filling out the fields, actor can click [save as draft] button, if want to make a records as **draft**. * After that, system will proceed to store a data into database and will automatically check a validity of data (such as; data type, mandatory data, and data length) * If done, actor can click [Back] button to return to the main screen. |
| 5. | Filter and sort service history records | Maintenance Section User | * Actor can filter and sort the record based on columns where displayed on the list. * To filter the records, actor can enter keywords into the filter fields. If matched data founds, the filtered result will be displayed on list. If doesn’t the list will be empty. * To sort the records, actor can click header of column on the list table. After that the records will be sorted by ascending or descending, alternately. * Upon select [record number filter] drop down list, actor can select a records number to display on the list. At least only one item can be selected. |
| 6. | Update service history record | Maintenance Section User | * If service history screen opens, actor can determined the agreement number and police number that want to be display in the service history. * Upon clicking […] button on the agreement number fields, the agreement look-up screen will be displayed. Agreement number can be selected (only one record can be select). After it’s done, agreement information field will be filled automatically. * After determined agreement number, actor can determine the police number by clicking […] button on the police number field. The Units (Object Lease) look-up screen will be displayed. Police number can be selected (only one record can be select). After it’s done, unit’s information field will be filled automatically. * After determining agreement number and police number, user can click [Details] button to edit the details of service history record. Once it done, ‘Details’ screen will be displayed. * If ‘Details’ screen opens, actor can edit a record by inputting a data on each of the fields. * The mandatory fields cannot be blank. |
| 7. | Delete a service history record | Maintenance Section User | * Actor can delete the record of service history that display on the list. Only data with status draft can be deleted. |
| 8. | Validate service history record | Maintenance Section User | * Once data is saved and data has been confirmed, actor can validate the record by clicking [Validate] button. * After that system will set a record as **validate (or not draft)**. * If done, actor can click back button to return to the list of service history. |
| 9. | Set to draft of service history | Maintenance Section User | * To do this process, actor should open detail screen by clicking [Details] button on the list. Once it open, actor can set to draft by clicking [set to draft] button. If the records still need to revise and status still not valid yet, actor can change the record status from **validated** to **draft**. * After that, system will proceed automatically to change the status, and actor can click [Back] button to return to the list. |

* + 1. **Status management and roles**

The figure below is describing a role matrix (including relation with Status) of actor that already identified in a function.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role of Actor** | **Filter and sort** | **Save as draft** | **Save changes**  **(draft)** | **validate** | **Validate** |
| Maintenance user | Yes | Yes | Yes | Yes | Yes |

The figure below is describing transition of Status that performed when people interact in a function. In this features, drafting process triggered on the agreement module. If service schedule already generated on the agreement module, so the status of service records will be draft too, and vice versa.

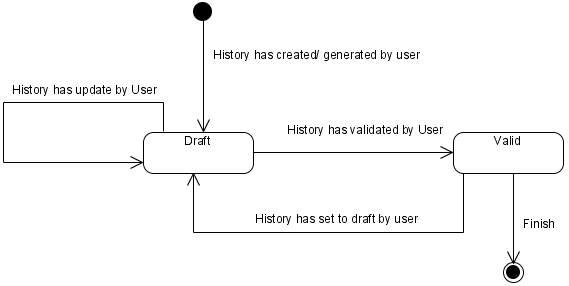
****

Figure 3 – Status diagram of Service History Inquiry

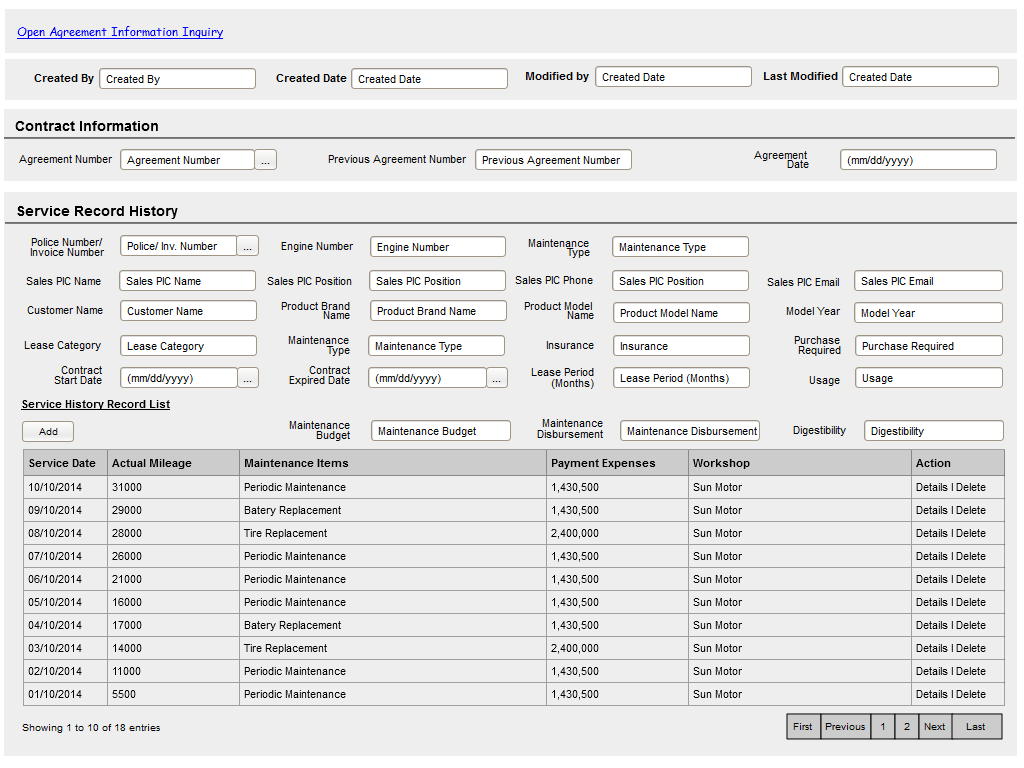
* + 1. **Sitemap design**

Describe a web page that lists the pages on a web site to users. Usually sitemap organized in hierarchical style.

Figure 4 – Sitemap design of Service History Inquiry

* + 1. **Screen design**
       1. **Service history inquiry**

This screen is designed to service history inquiry. On this screen actor can open service records. And also, actor can open another screen to add and/or update service history record.



To open details service history

General information of agreement

General information of service history/ maintenance condition

To view audit rails

To add new records

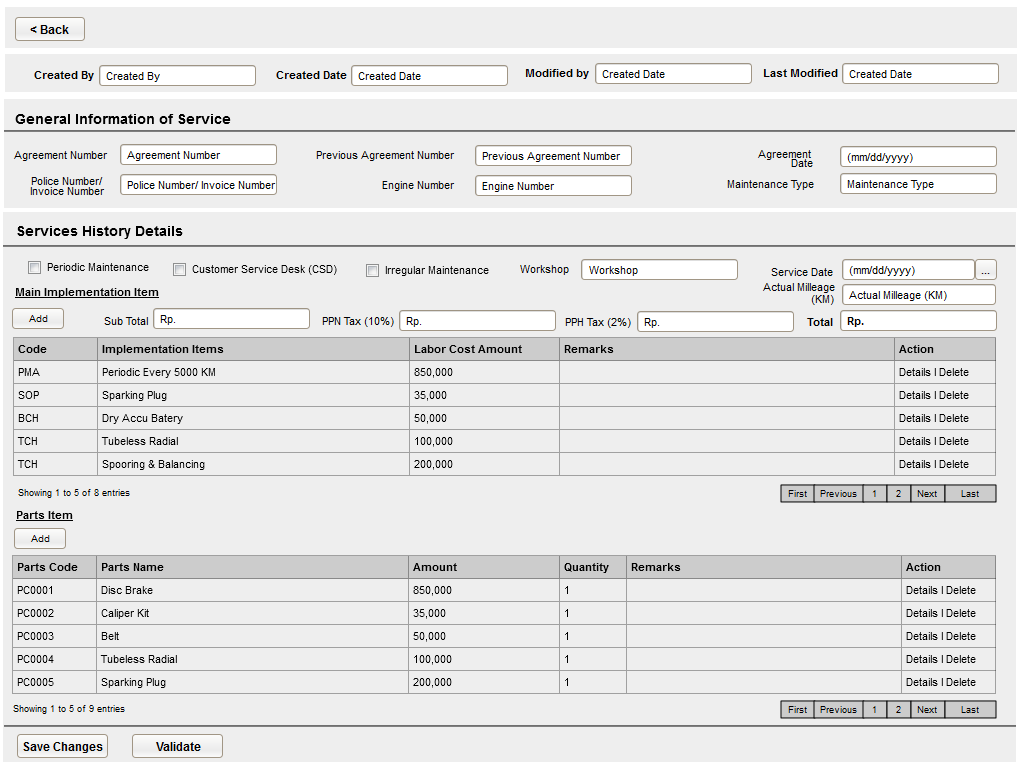
Data paging information

To open paging of records

Image 1 – Service History Inquiry Screen

* + - 1. **Add new service history records**

This screen is designed to add new service history record. After filling out necessary data on the screen, actor can store a data into the system by clicking the action button.



Buttons to save and validate the records

Details information of service history

To delete the items and open details screen

To delete the items and open details screen

To view audit rails

General Information of service history/ maintenance condition

To add Implementation Items

To add Part Items

Data paging information

To open paging of records

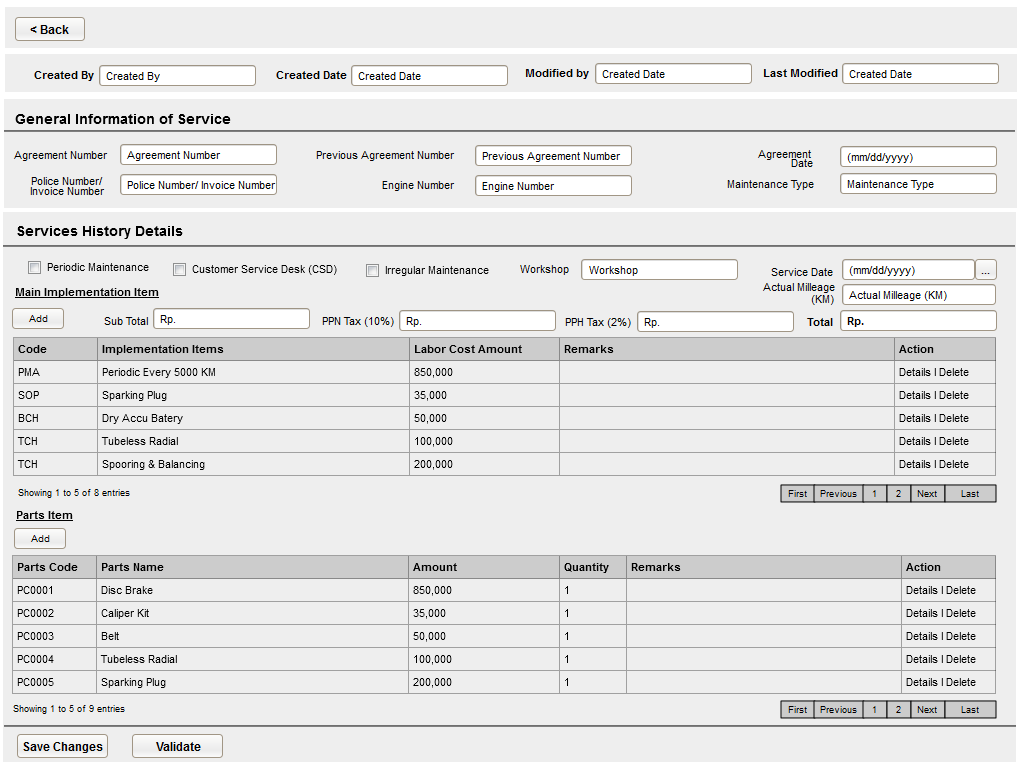
To open paging of records

Data paging information

Image 2 – Screen of Add New Service History Record

* + - 1. **Update service history records**

This screen is designed to update the service history records. After changing some data on the screen, actor can store a data into the system by clicking the action button.



Buttons to save and validate the records

To open paging of records

Data paging information

To add Part Items

To delete the items and open details screen

Data paging information

To open paging of records

To add Implementation Items

To delete the items and open details screen

Details information of service history

To view audit rails

General Information of service history/ maintenance condition

Image 3 – Screen of Update Details Service History

* + 1. **Screen features**

|  |  |
| --- | --- |
| **Features** | **Description** |
| [Filter and sort] | To perform data filtering and data sorting of records. |
| [Add new] | To add new record by entering data on the ‘create’ form. |
| [Save as draft] or [Set to draft] | To save a record as a draft. |
| [Validate] | To validate a record as a final. |
| [Save changes] | To update changes when editing sorting record. |
| [Delete] | To delete the service history records. |

* + 1. **Data structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Data Type** | **Data Length** | **Format** | **Mandatory** |
| **Service History Summary** | | | | |
| MaintenanceBudget | Decimal | 18,0 | ###,###,###.## | *Yes* |
| MaintenanceDisburesement | Decimal | 18,0 | ###,###,###.## | *Yes* |
| Digestibility | Decimal | 18,0 | ###,###,###.## | *Yes* |
| **Service History Record** | | | | |
| MaintenanceType | Text | 100 | N/A | *Yes* |
| ServiceDate | Date | N/A | mm-dd-yyyy | *Yes* |
| MonthlyMileage | Numeric | N/A | N/A | *No* |
| MaintenanceItem | Text | 100 | N/A | *Yes* |
| PaymentExpense | Decimal | 18,0 | ###,###,###.## | *No* |
| SuplierCode | Text | 50 | N/A | *Yes* |
| IsPerodicMaintenance | Yes/No | N/A | N/A | *No* |
| IsCServiceDesk | Yes/No | N/A | N/A | *No* |
| IsIregularMaintenance | Yes/No | N/A | N/A | *No* |
| SubTotal | Decimal | 18,0 | ###,###,###.## | *No* |
| PPNTax | Decimal | 18,0 | ###,###,###.## | *No* |
| PPHTax | Decimal | 18,0 | ###,###,###.## | *No* |
| TotalIncludeTax | Decimal | 18,0 | ###,###,###.## | *No* |
| **Audit Rail** | | | | |
| Create By | Text | 100 | N/A | *Yes* |
| Create Date | Date time | N/A | mm-dd-yyyy:hh-mm-ss | *No* |
| Last Modified By | Text | 100 | N/A | *Yes* |
| Last Modified | Date | N/A | mm-dd-yyyy | *No* |
| **Status Management** | | | | |
| Is Deleted | Yes/No | N/A | N/A | *Yes* |
| Is Draft | Yes/No | N/A | N/A | *Yes* |
| Is Submitted | Yes/No | N/A | N/A | *Yes* |
| Is Valid | Yes/No | N/A | N/A | *Yes* |

1. Appendix
   1. Use Case Terminology

|  |
| --- |
| ***What is the use case?***   1. ***A use case is*** *a written description of how users will perform tasks on your website.  It outlines, from a user’s point of view, a system’s behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled. (*[*http://www.usability.gov/how-to-and-tools/methods/use-cases.html*](http://www.usability.gov/how-to-and-tools/methods/use-cases.html)*)* 2. ***A use case is*** *a list of steps, typically defining interactions between a role and a system. (*[*http://en.wikipedia.org/wiki/Use\_case*](http://en.wikipedia.org/wiki/Use_case)*)* 3. ***A use case is*** *a formal way of representing how a business interacts with its environment. it summarized into a single picture.* ([*http://romisatriawahono.net/lecture/sad/romi-sad-05-implementation-march2014.pptx*](http://romisatriawahono.net/lecture/sad/romi-sad-05-implementation-march2014.pptx)*)*   ***What are Benefits of use cases?***   1. ***Use cases*** *add value because they help explain how the system should behave and in the process, they also help brainstorm what could go wrong.  They provide a list of goals and this list can be used to establish the cost and complexity of the system. Project teams can then negotiate which functions become*[*requirements*](http://www.usability.gov/how-to-and-tools/methods/requirements.html)*and are built.*     *(*[*http://www.usability.gov/how-to-and-tools/methods/use-cases.html*](http://www.usability.gov/how-to-and-tools/methods/use-cases.html)*)*   1. *With the help of use case diagram, you can discuss and communicate:*  * *The scenarios in which your system or application interacts with people, organizations, or external systems.* * *The goals that it helps those actors achieve.* * *The scope of your system.*   *(*[*http://msdn.microsoft.com/en-us/library/dd409432.aspx*](http://msdn.microsoft.com/en-us/library/dd409432.aspx)*)* |