SYSTEM REQUIREMENT SPECIFICATION

BROKEN FACILITY SYSTEM

FOR INFORMATICS DEPARTMENT

Prepared by:

5114100070 Prasetyo Nugrohadi

5114100158 Muhammad Ghazian

Informatics Department - Institut Teknologi Sepuluh Nopember

Kampus ITS Keputih Sukolilo Surabaya

CHANGE LIST

Revision	Description
A	
B	
C	
D	
E	
F	
G	

DATE INDEX	-	A	B	C	D	E	F	G
Written by								
Checked by								
Verified by								

Page Revision List

Page	Revision	Page	Revision

Table of Contents

Table List

Figure List

1 Preliminary

1.1 Document Aim

This document contains Software Requirement Specification for Broken Facility System. The creation of this document is meant for giving proper documentation about the software artifact that will be built. The system may be explained using general explanation denoted as figures, or in thorough detail.

This document will be used as starting point in developing the system and will be used as the evaluation material during developing the software and by the end of the development. By the existence of this SRS, there is high hope that the software development process would be more structured and focused, and also to prevent ambiguity for the software developer.

1.2 Scope of Problem

The software that will be built is Broken Facility System for Informatics Department ITS. It is a software in form of desktop system information. The system can be used to facilitate the report regarding any broken facility in Infomatics Department. This system is aimed to do several activities:

1) To facilitate the civitas academica of Informatics Department to resolve the problem about unsatisfactory condition of the facility.
2) As the media for the staff of Informatics Department to control the health of their facility.

It is hoped that with this system, the overall condition of facilities in Informatics Department can be improved and maintained.

1.3 Glossary

The following list is the term used for the rest of this document:

- SRS : Software Requirements Specification, or
  SKPL : *Spesifikasi Kebutuhan Perangkat Lunak*
  The document containing the software specification.
- IEEE : Institute of Electrical and Electronics Engineering
  International Standard for developing and designing product.
- ANSI : American National Standard Institute
  American standardization institute.
- TBD : To Be Defined
- LAN : Local Area Network

1.4 Naming and Numbering Nomenclature

The writing of this SRS uses different naming and numbering rule for different section. The rule for naming and numbering that will be used is contained within Table 1 below.

Table 1 Naming and Numbering Nomenclature

Section/Subject	Naming/Numbering Rule
Functional Requirement	SKPL-FXX : Means the XXth functional requirement
Non Functional Requirement	SKPL-NFXX : Means the XXth non-functional requirement
Functional Requirement Summary	SKPL-Fxxx where xxx is the last 3 numeric digit starting from 000
Non-functional Requirement Summary	SKPL-NFxxx where xxx is the last 3 numeric digit starting from 000

1.5 Reference

The documents that will be used as reference of making this SRS is as follows:

1) *Dokumen* Software Requirement Spesification (SRS) – IEEE year 1999 by Karl E. Wiegers.
2) *Panduan Penggunaan dan Pengisian Spesifikasi Perangkat Lunak (SKPL),* Informatics Department, Institut Teknologi Sepuluh November.
3) *Panduan Penggunaan dan Pengisian Spesifikasi Perangkat Lunak (SKPL),* Informatics Department, Institut Teknologi Bandung.

1.6 Document Summary

In general, this document consists of 3 section:

- Section 1: Preliminary. This part is the introductory section of this SRS, containing the aim of the document, scope of the problem, also includes the definition and term that will be used. This part also explains about the general description about the SRS.
- Section 2: Software's Global Description. This part defines the perspective of the software along with the assumptions in consideration and any dependencies that will be used during the development of the system
- Section 3: Detailed Requirement Description. This part describes the specific needs that aroused by the system, including the external interface, functional requirement, performance requirement, development boundary, software attributes, and other requirement of Broken Facility System

2 Software General Description

2.1 System General Description

The broken facility system is a system that monitors the condition of the facility in the Informatics Department of ITS. This system stores any report submitted by the privileged user. This system is mainly driven by two users, i.e. the civitas academica of Informatics Department of ITS, and the staff of Informatics Department of ITS. The civitas academica of Informatics can report whichever equipment in the campus that is not in good shape. The report then will be sent to the staff. Later on, the staff will take care of the report and check the problematic facility. In the end of the maintenance of the facility, the staff provides feedback to the report as solved, which the submitter will be able to see.

The developed software system has several components based on the user. The said components are as follows :

1) From the user's perspective (i.e. civitas academica of Informatics), the system facilitates them to report any damaged equipment or any other facility which not in good condition. This helps them to reach to the management such that the matter is to be resolved quickly. The benefit of this system is that the user does not have to go to the staff office directly only to inform the facility's condition. The user also may see whether the matter has been resolved or not. Since the system needs authentication to limit the report only comes from the civitas academica of Informatics Department, the user needs to have an account, and the system supports the user to request for account creation.
2) From the staff's perspective, the system helps them to monitor the overall condition of the campus's facilities. By facilitating the user to file in report about unsatisfactory condition of the facility, the workload to manage the facility decreases. Also, the staff may create other account, may that be for another staff or for the user. In case there are any incoming account creation request from the user, the staff can validate which request are to be processed further, and which are discarded

2.2 Product's Function

The broken facility system has several main functions, that is:

1. (SKPL-F1) Both user and staff can log into the system.
2. (SKPL-F2) Department staff possess overview about conditions of the department's equipment.
3. (SKPL-F3) User can submit a report about broken equipment.

2.3 User Characteristics

The characteristics of the user related to the broken facility system is explained in the following table :

No	User Category	Task	Privilege	Preliminary Skill
1.	Administration Staff	Manage the information system	May process the report or delete it. Also processes the account creation	1.Able to operate computer 2.Able to operate web and database
2.	Civitas Academica	Access the information system	May file in report and check whether the report is resolved or not	1.Able to operate computer 2.Able to use internet

2.4 Boundary

The development of the information system has several limitation as follows:

1. The Broken Facility System uses website, hence the development will use HTML, PHP, CSS, and related framework that supports the development.
2. Interface is simplistic.
3. The supporting software that will be used is XAMPP, Sublime, Notepad++, and Visual Studio Code.

2.5 Operating Environment

The application will operate in environment with several specification:

Platform sistem operasi : Microsoft Windows
Versi sistem operasi : Windows 7/8/10
DBMS : MySQL
Kerangka kerja : Laravel or CodeIgniter

3 Requirement General Description

3.1 External interface requirement

3.1.1 User interface

Broken Facility System uses graphical user interface (GUI). User can enter the input using keyboard and mouse, and operate it using Windows operation system.

3.1.2 Hardware interface

System runs on a computer server. Any computer that will be connected to the system must connected with LAN.

3.1.3 Software Interface

Broken Facility System will be built using HTML, CSS, PHP and MySQL.

3.1.4 Communication Interface

Broken Facility System is a system that connected to internet connection.

3.2 Functional Description

3.2.1 Use Case Diagram

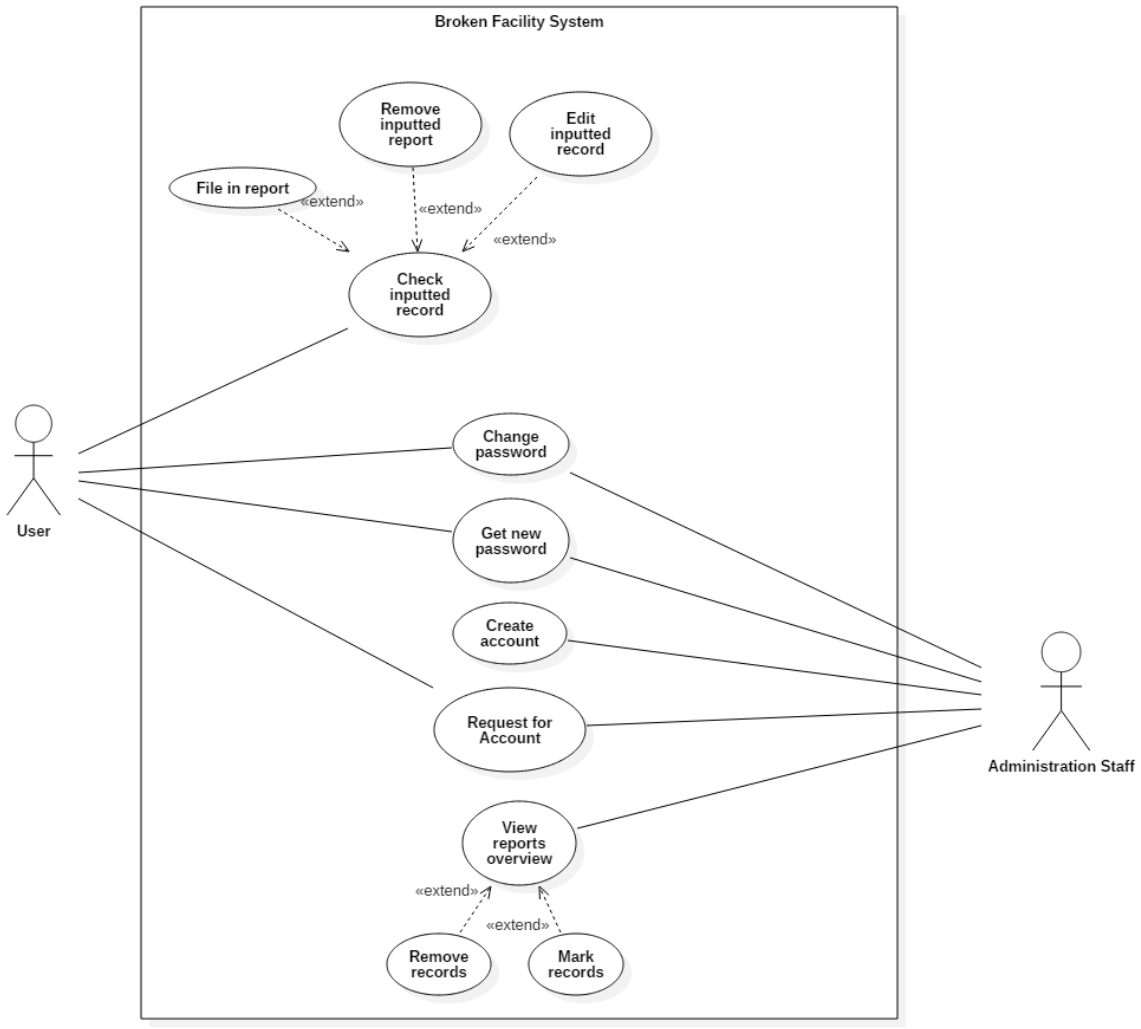


Figure 1. Use Case Diagram

3.2.2 Function 1: Create Account

3.2.2.1 Scenario: Create Account

Use Case ID	UC 1
Use Case Name	Create Account
Description	User create his account
Relationship	-
Related actor	User, Administrator
Pre-condition	• Has the application • Network connection is active
Post-condition	User account registered

Normal Flow	
Actor	System
1. User press create account 3. User fill the data requirement 5. Admin check the data 6. User's account created	2. System open create account page 4. System validate the inputted data
Alternate Flow	

3.2.2.2 Activity Diagram: Create Account

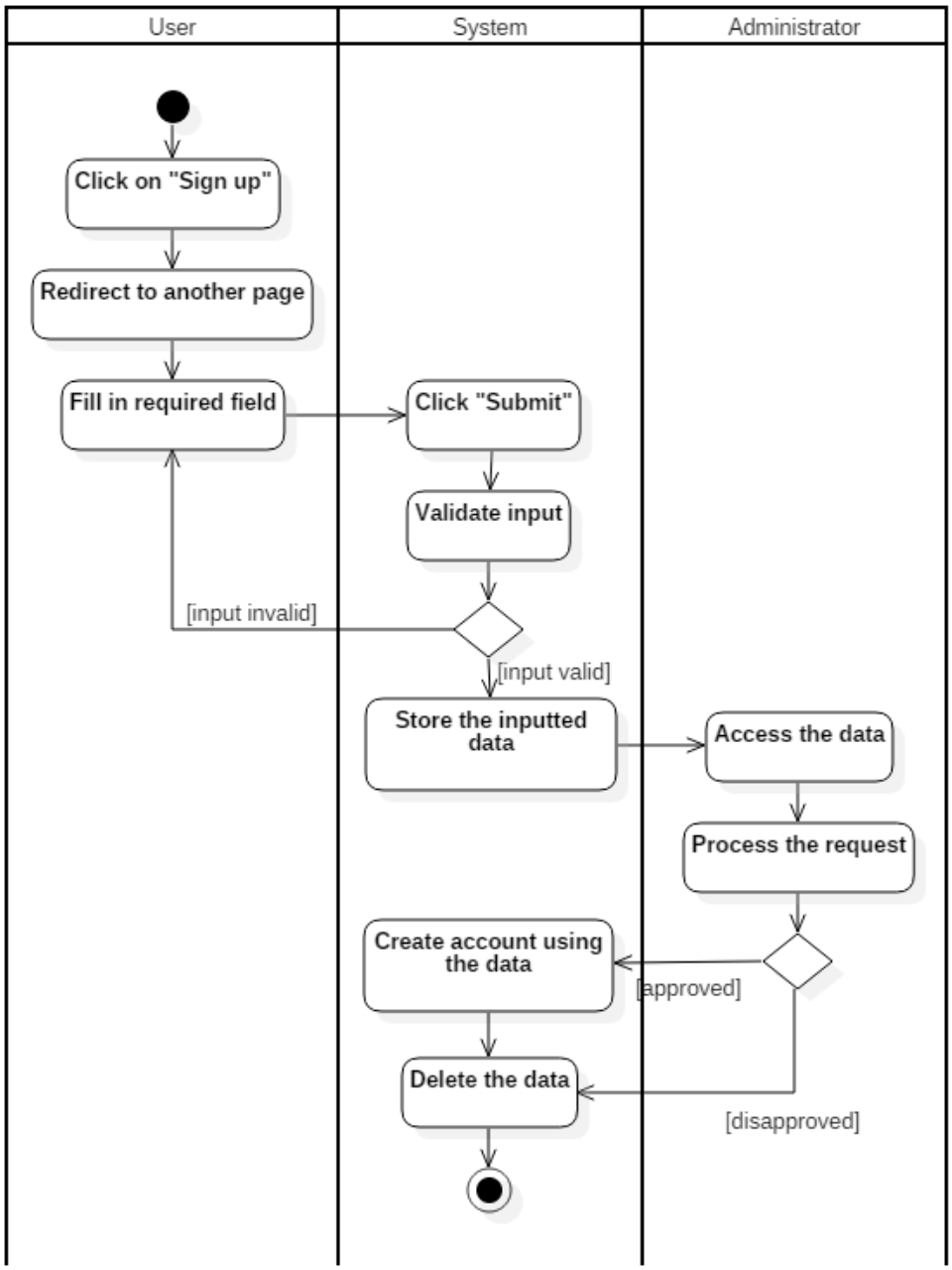


Figure 2. Activity Diagram "Create Account"

3.2.2.3 Sequence Diagram: Create Account

Figure 3. Sequence Diagram “Create Account”

3.2.2.4 Object Collaboration Diagram: Create Account

Figure 4. Object Collaboration Diagram “Create Account”

3.2.3 Function 2: Get New Password

3.2.3.1 Scenario: Get New Password

Use Case Id	UC 2
Use Case Name	Get New Password
Description	Retrieves new password generated by the system and send it to the email which registered for the corresponding account
Relationship	-
Related Actor	User, Administrator
Pre-condition	Actor is on the main page while not logged in
Post-condition	The password is sent to the actor’s email

Normal Flow	
Actor	System
1. Actor clicks ‘Forgot password’ link 3. Actor fills in the username of his account 4. Actor clicks ‘Submit’ button	2. System open the forgot password form 5. System check the inputted username 6. System generate new password 7. System send the new password to user’s email

Alternate Flow	
Alternate Flow	
Exception	4.a The username is not found. The flow ceases

Activity Diagram: Get New Password

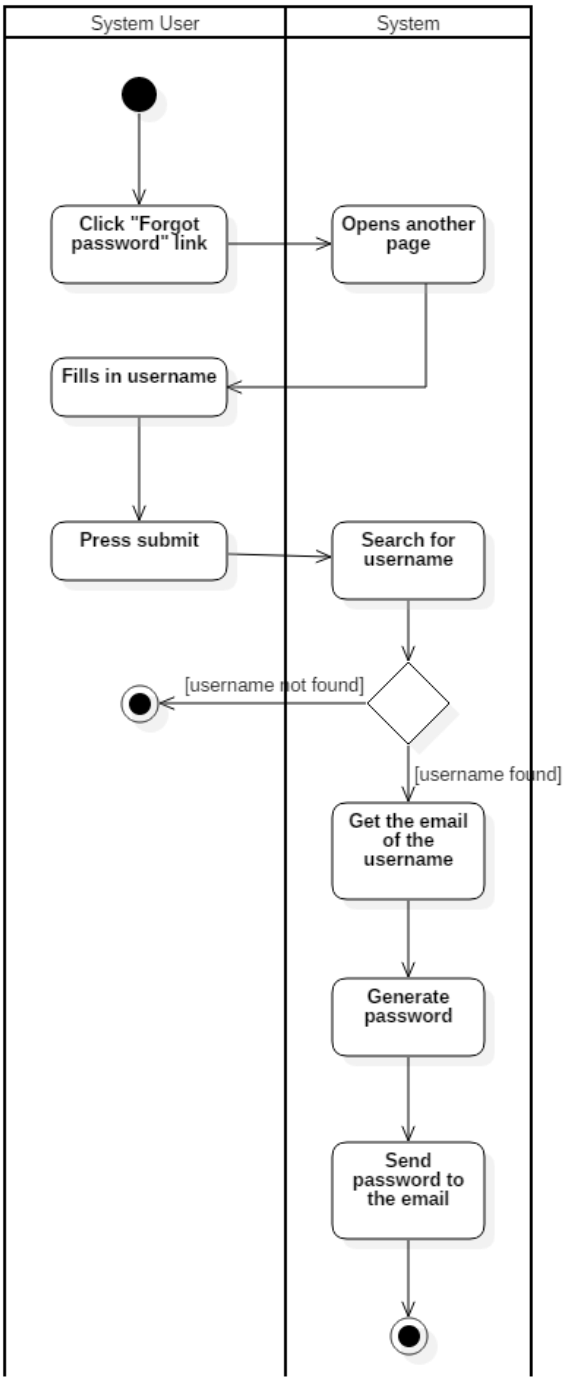


Figure 5. Activity Diagram “Get New Password”

3.2.3.2 Sequence Diagram: Get New Password

Figure 6. Sequence Diagram “Get New Password”

3.2.3.3 Object Collaboration Diagram: Get New Password

Figure 7. Object Collaboration Diagram “Get New Password”

3.2.4 Function 3: Change Password

3.2.4.1 Scenario: Change Password

Use Case Id	UC 3
Use Case Name	Change Password
Description	Change password for an account
Relationship	-
Related Actor	Administrator, user
Pre-condition	Actor has to be logged in
Post-condition	The password of the user or administrator is changed

Normal Flow	
Actor	System
1. Actor opens ‘Account’ tab on navigation bar 3. Actor clicks ‘Change Password’ button 5. Actor fills his current password, new password, and another password field to confirm the new password 6. Actor presses ‘Submit’ button	 2. System open account page 4. System open change password page 7. System validate the input 8. System change the user’s password
Alternate flow	
Actor	System
4.a. Actor filled in wrong current password.	4.a.1. Actor redirected to previous page with error message,

4.b. Actor filled in invalid new password such as too short etc. with error message 4.c. Actor filled in wrong new password confirmation.	4.b.1. Actor redirected to previous page, continue with normal flow (4) 4.c.1. Actor redirected to previous page, continue with normal flow (4)
Exception	-

3.2.4.2 Activity Diagram: Change Password

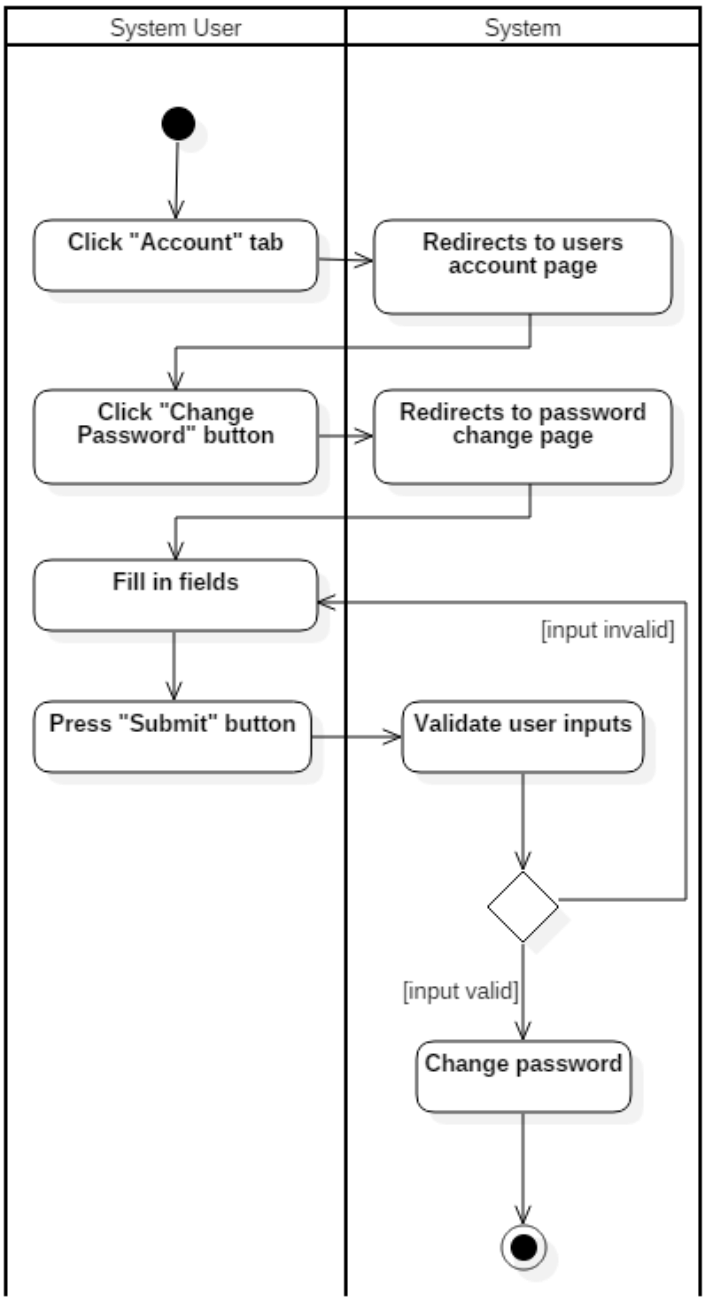


Figure 8. Activity Diagram "Change Password"

3.2.4.3 Sequence Diagram: Change Password

Figure 9. Sequence Diagram “Change Password”

3.2.4.4 Object Collaboration Diagram: Change Password

Figure 10. Object Collaboration Diagram “Change Password”

3.2.5 Function 4: Show Reports Overview

3.2.5.1 Scenario: Show Reports Overview

Use Case Id	UC 4
Use Case Name	Show reports overview
Description	This page shows the user inputted records about broken facility, along each of its state: solved or unsolved
Relationship	-
Related Actor	Administrator
Pre-condition	Administrator has to be logged in
Post-condition	Administrator in reports overview page

Normal Flow	
Actor	System
1. Administrator presses the ‘Overview’ tab on the navigation bar	2. System shows up a page containing necessary information about user reports
Alternate Flow	
Actor	System
1.a. Administrator just logged in into the system	1.a.1. System immediately redirects administrator to ‘Overview’ section continue to Normal Flow (2)
Exception	-

3.2.5.2 Activity Diagram: Show Reports Overview

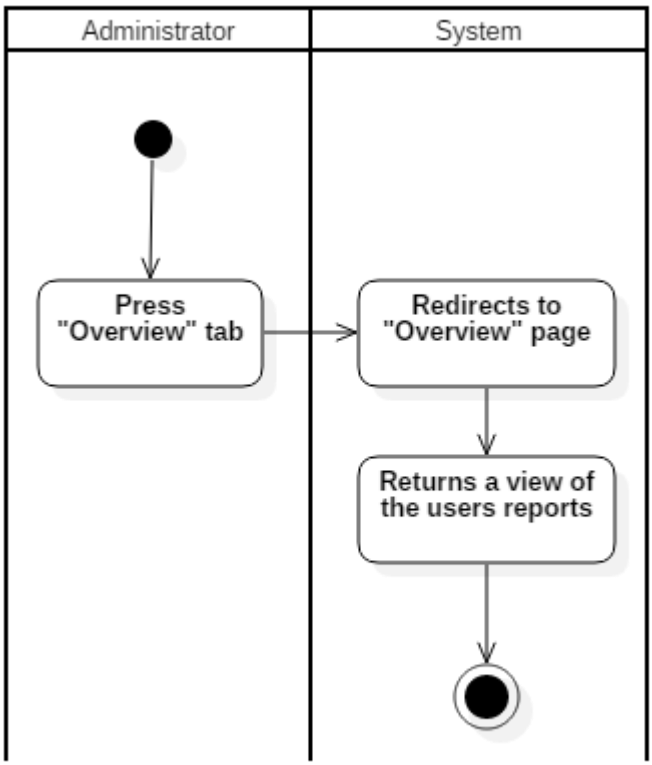


Figure 11. Activity Diagram “Show Reports Overview”

3.2.5.3 Sequence Diagram: Show Reports Overview

Figure 12. Sequence Diagram “Show Reports Overview”

3.2.5.4 Object Collaboration Diagram: Show Reports Overview

Figure 13. Object Collaboration Diagram “Show Reports Overview”

3.2.6 Function 5: Remove Records

3.2.6.1 Scenario: Remove Records

Use Case Id	UC 5
Use Case Name	Remove records
Description	Remove user inputted records about broken facility. The removal may be done for many records at once
Relationship	Extends “Show reports overview”
Related Actor	Administrator
Pre-condition	• Administrator has to be logged in • Page ‘Overview’ is opened
Post-condition	The marked (i.e. ticked) report is no longer exist and will not show in ‘Overview’ page

Normal Flow	
Actor	System
1. Administrator ticks checkbox that is presented inline with the records which will be removed 2. Administrator presses ‘Action’ drop down located below the list of records 4. Administrator chooses ‘Remove’ action 6. Administrator confirms the removal	 3. System shows drop down of possible action 5. The system prompts administrator to confirm removal 7. System remove the records, page refreshed

Alternate Flow	
Actor	System
1.a. Administrator filters the record beforehand 1.a.1. Administrator presses ‘Filter’ button located above the list of records 1.a.3. Administrator supplies the information of which will be used as the filter 1.a.4. Administrator presses ‘Filter’ button	 1.a.2. System redirect to interface with filtering components 1.a.5. System retrieves data that qualifies the specified filter, interface shows list of recoreds, then continue to normal flow (1)
Exception	6.a. Administrator cancels the removal. The flow ceases

3.2.6.2 Activity Diagram: Remove Records

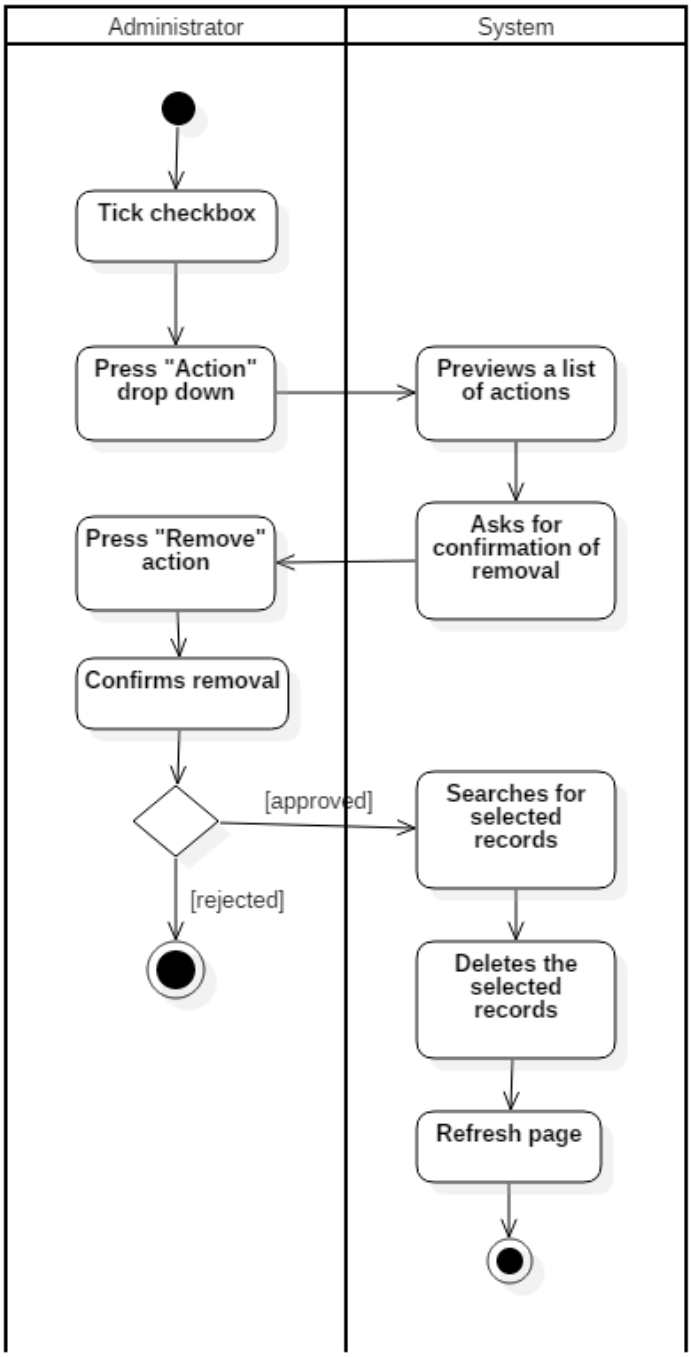


Figure 14. Activity Diagram “Remove Records”

3.2.6.3 Sequence Diagram: Remove Records

Figure 15. Sequence Diagram “Remove Records”

3.2.6.4 Object Collaboration Diagram: Remove Records

Figure 16. Object Collaboration Diagram “Remove Records”

3.2.7 Function 6: Mark Records

3.2.7.1 Scenario: Mark Records

Use Case Id	UC 6
Use Case Name	Mark records
Description	Mark records as solved or unsolved. The marking can be done for many records at once
Relationship	Extends “Show reports overview”
Related Actor	Administrator
Pre-condition	• Administrator has to be logged in • Page ‘Overview’ is opened
Post-condition	The marked (i.e. ticked) report has its state changed

Normal Flow	
Actor	System
1. Administrator ticks checkbox that is presented inline with the records which will be marked 2. Administrator presses ‘Action’ drop down located above the list of records 4. Administrator chooses one of the ‘Mark’ action 6. Administrator confirms the marking	3. System shows drop down of possible action 5. The system prompts administrator to confirm marking 6. System mark the records, page refreshed
Alternate Flow	
Actor	System
1.a. Administrator filters the record beforehand 1.a.1. Administrator presses ‘Filter’	

button located above the list of records 1.a.2. Administrator is redirected to interface with filtering components 1.a.3. Administrator supplies the information of which will be used as the filter 1.a.4. Administrator presses 'Filter' button	1.a.5. System retrieves data that qualifies the specified filter, interface refreshed and show list of records then continue to normal flow (1)
4.a. Administrator chooses 'Mark as solved'	4.a.1. System will change the marked data as 'Solved', then continue to normal flow (5)
4.b. Administrator chooses 'Mark as unsolved'	4.b.1. System will change the marked data as 'Unsolved', then continue to normal flow (5)
Exception	6.a. The administrator cancels the marking. The flow ceases

3.2.7.2 Activity Diagram: Mark Records

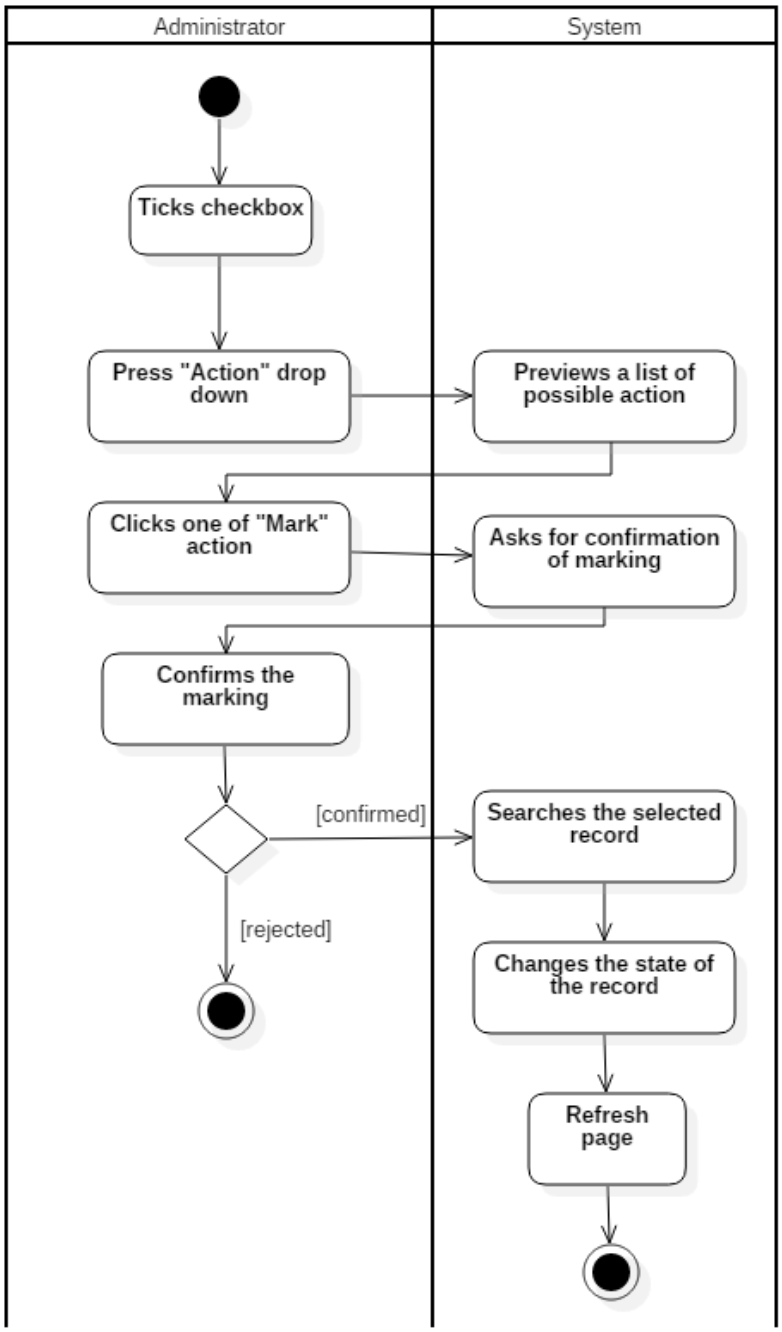


Figure 17. Activity Diagram "Mark Records"

3.2.7.3 Sequence Diagram: Mark Records

Figure 18. Sequence Diagram “Mark Records”

3.2.7.4 Object Collaboration Diagram: Mark Records

Figure 19. Object Collaboration Diagram “Mark Records”

3.2.8 Function 7: Check Inputted Record

3.2.8.1 Scenario: Check Inputted Record

Use Case Id	UC 7
Use Case Name	Check Inputted Record
Description	Shows the overview of the reports that has been filed in by the corresponding user
Relationship	-
Related Actor	• User
Pre-condition	User is logged in
Post-condition	User is in ‘Overview’ page
Normal Flow	
Actor	System
1. User clicks on ‘Overview’ tab on navigation bar	2. System shows a page with lists of inputted report is shown
Alternate Flow	
Actor	System
1.a. User just logged in	1.a.1. System immediately redirect to ‘Overview’ page, then continue to normal flow (2)
Exception	-

3.2.8.2 Activity Diagram: Check Inputted Record

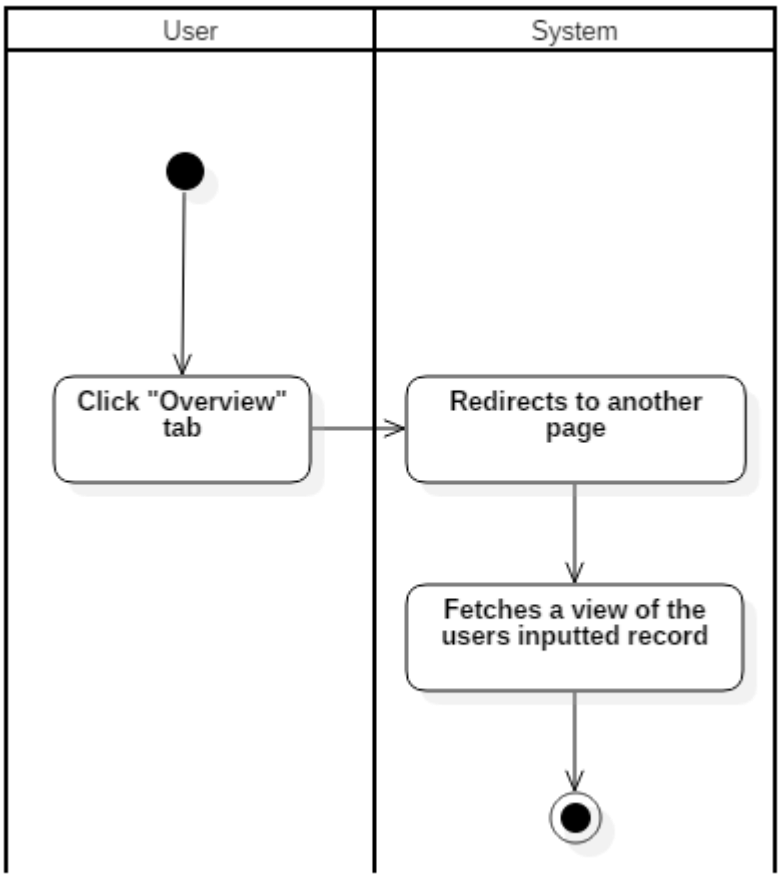


Figure 20. Activity Diagram “Check Inputted Record”

3.2.8.3 Sequence Diagram: Check Inputted Record

Figure 21. Sequence Diagram “Check Inputted Record”

3.2.8.4 Object Collaboration Diagram: Check Inputted Record

Figure 22. Object Collaboration Diagram “Check Inputted Record”

3.2.9 Function 8: File In Report

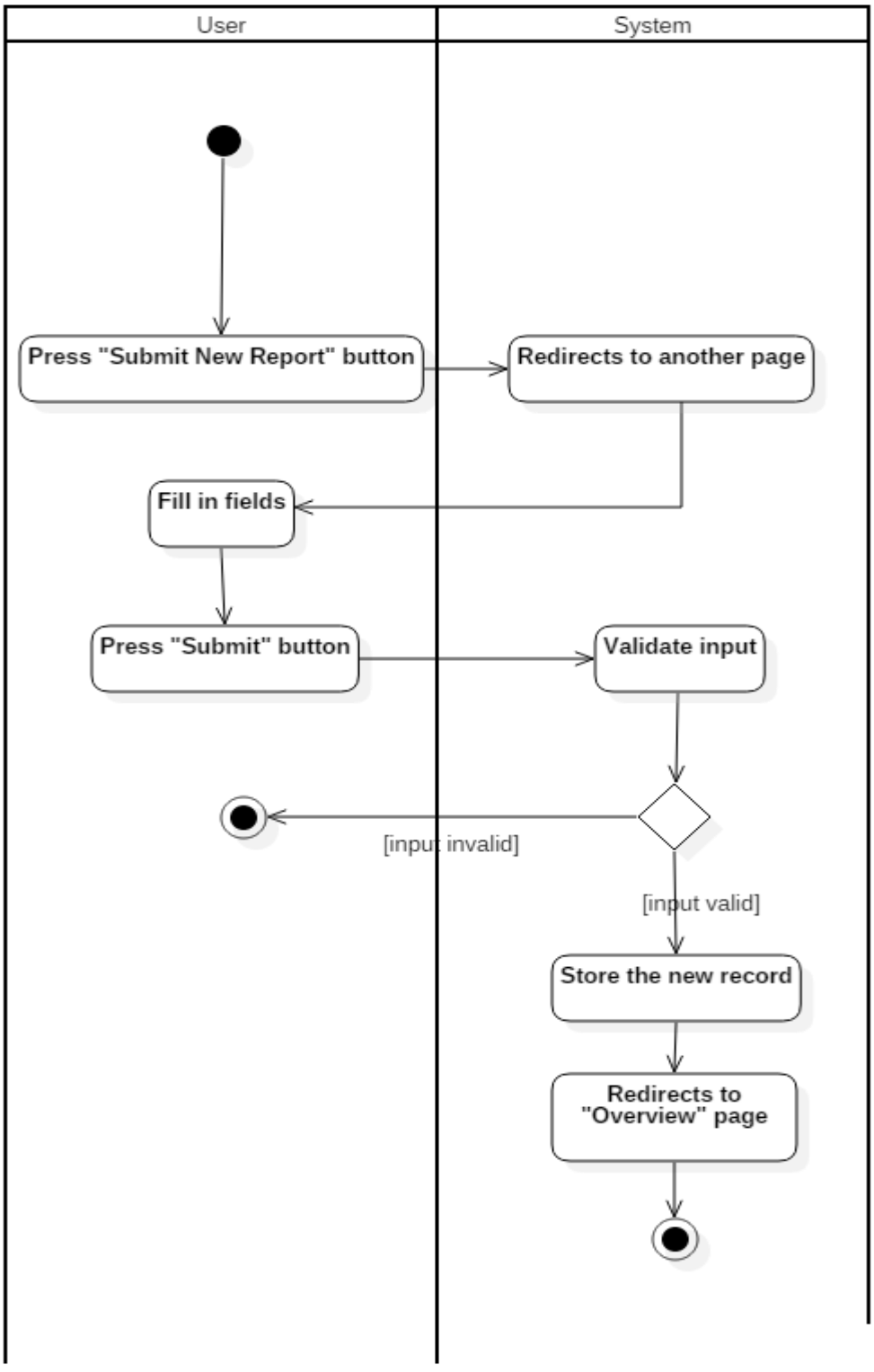
3.2.9.1 Scenario: File In Report

Use Case Id	UC 8
Use Case Name	File in Report
Description	Checks all the file that are reported broken
Relationship	Extends “Check Inputted Records”
Related Actor	User
Pre-condition	User is logged in
Post-condition	A new report is submitted

Normal Flow	
Actor	System
1. User presses ‘Submit new report’ button 3. User supplies necessary information to the given form 4. User presses ‘Submit’	2. System redirected to another page 5. System redirect to ‘Overview’ page
Alternate Flow	
Alternate Flow	4.a. Information supplied is incorrect. 4.a.1. User is redirected to previous page with error message 4.a.2. The flow then goes to Basic Flow (3) 4.b. Information supplied is not complete i.e. some necessary field is not filled. 4.b.1. User is redirected to previous page with error message 4.b.2. The flow then goes to Basic Flow (3)
Exception	-

3.2.9.2 Activity Diagram: File In Report

Figure 23. Activity Diagram “File In Report”



3.2.9.3 Sequence Diagram: File In Report

Figure 24. Sequence Diagram “File In Report”

3.2.9.4 Object Collaboration Diagram: File In Report

Figure 25. Object Collaboration Diagram “File In Report”

3.2.10 Function 9: Remove Inputted Record

3.2.10.1 Scenario: Remove Inputted Record

Use Case Id	UC 9
Use Case Name	Remove Inputted Record
Description	Remove the user inputted report. The removal may be done for many records at once
Relationship	Extends “Check Inputted Record”
Related Actor	User
Pre-condition	User is in ‘Overview’ page
Post-condition	The chosen data is removed

Normal Flow	
Actor	System
1. Ticks checkbox that is presented inline with the records which will be removed 2. Presses ‘Remove’ button located below the list of records 4. Confirms the removal	3. Prompts user to confirm removal 5. Refreshes page ‘Overview’
Alternate flow	
1.a. User filters the record beforehand	
Actor	System
1.a.1. User presses ‘Filter’ button located above the list of records 1.a.3. User supplies the information of which will be used as the filter	1.a.2. Redirect to interface with filtering components

1.a.4. User presses ‘Filter’ button	1.a.5. System retrieves data that qualifies the specified filter 1.a.6. System refreshes the interface where user is in 1.a.7. System shows the interface consisting a list of records 1.a.8. The flow then goes to Basic Flow (1)
2.a. No record is marked	
Actor	System
	2.a.1. System redirects to previous page with error message 2.a.2. The flow then goes to Basic Flow (1)
Exception	-

3.2.10.2 Activity Diagram: Remove Inputted Record

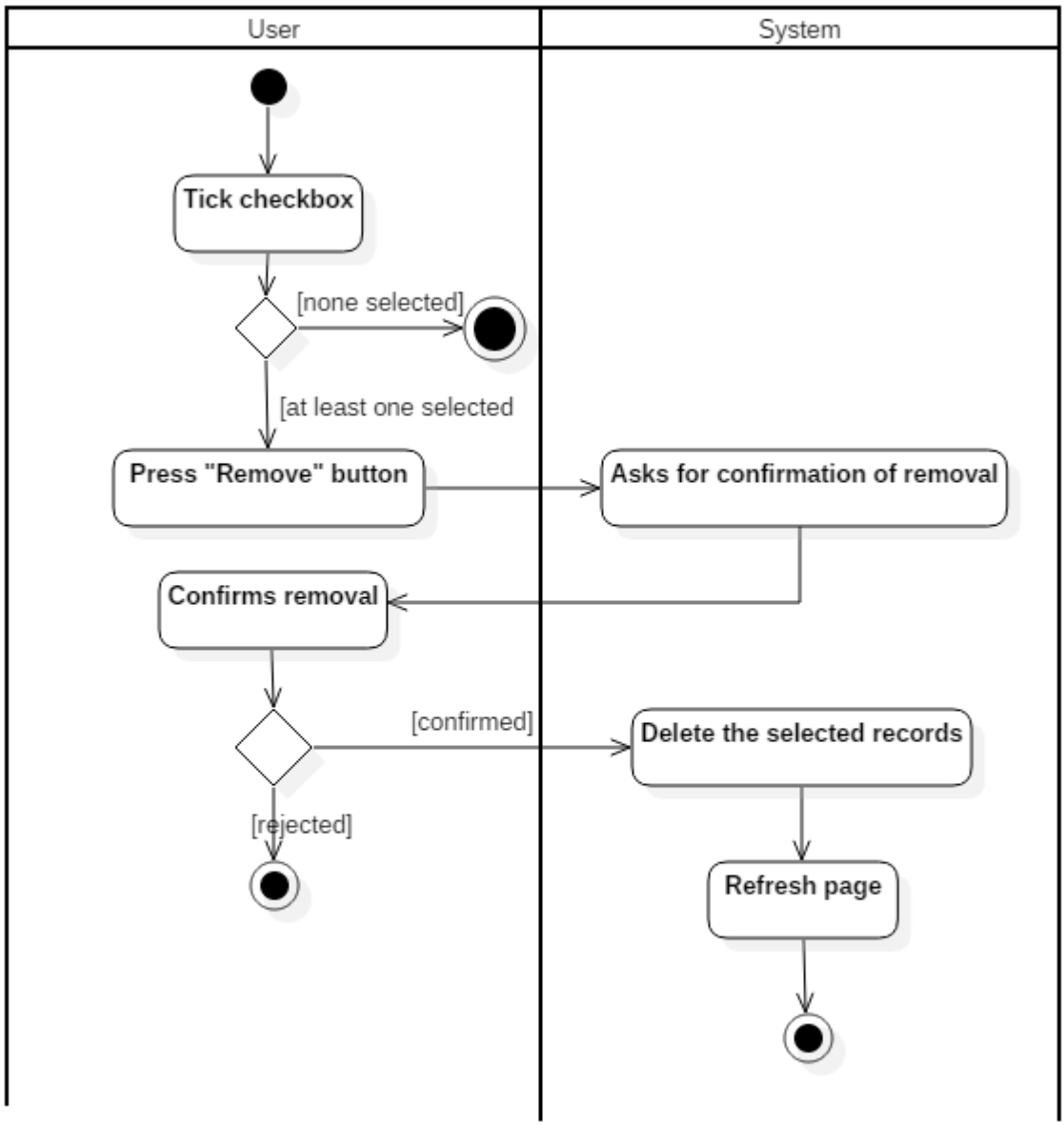


Figure 26. Activity Diagram “Remove Inputted Record”

3.2.10.3 Sequence Diagram: Remove Inputted Record

Figure 27. Sequence Diagram “Remove Inputted Record”

3.2.10.4 Object Collaboration Diagram: Remove Inputted Record

Figure 28. Object Collaboration Diagram "Remove Inputted Record"

3.2.11 Function 10: Edit Inputted Record

3.2.11.1 Scenario: Edit Inputted Record

Use Case Id	UC 10
Use Case Name	Edit Inputted Record
Description	Edits the information supplied of the previously inputted report by the corresponding user
Relationship	Extends "Check Inputted Record"
Related Actor	User
Pre-condition	User is in 'Overview' page
Post-condition	The corresponding record has its data changed

Normal Flow	
Actor	System
1. User clicks 'Edit' button that is placed inline with the record that will be edited	
	2. System redirects to another page that contains the form with the data of the edited record
3. User supplies necessary revisionon to the form 4. User presses 'Submit' button	
	5. System redirects to 'Overview' page

Alternate flow	
1.a. User filters the record beforehand	
Actor	System
1.a.1. User presses 'Filter' button located above the list of records	
	1.a.2. Redirect to interface with filtering components
1.a.3. User supplies the information of which will be used as the filter 1.a.4. User presses 'Filter' button	
	1.a.5. System retrieves data that qualifies the specified filter

	1.a.6. System refreshes the interface where user is in 1.a.7. System shows the interface consisting a list of records 1.a.8. The flow then goes to Basic Flow (1)
4.a. The revision is not complete i.e. some necessary field is not filled	
Actor	System
	4.a.1. System redirects to previous page with error message 4.a.2. The flow then goes back to Basic Flow (3)
4.b.. The revision is incorrect	
Actor	System
	4.b.1. System redirects to previous page with error message 4.b.2. The flow then goes back to Basic Flow (3)
Exception	-

3.2.11.2 Activity Diagram: Edit Inputted Record

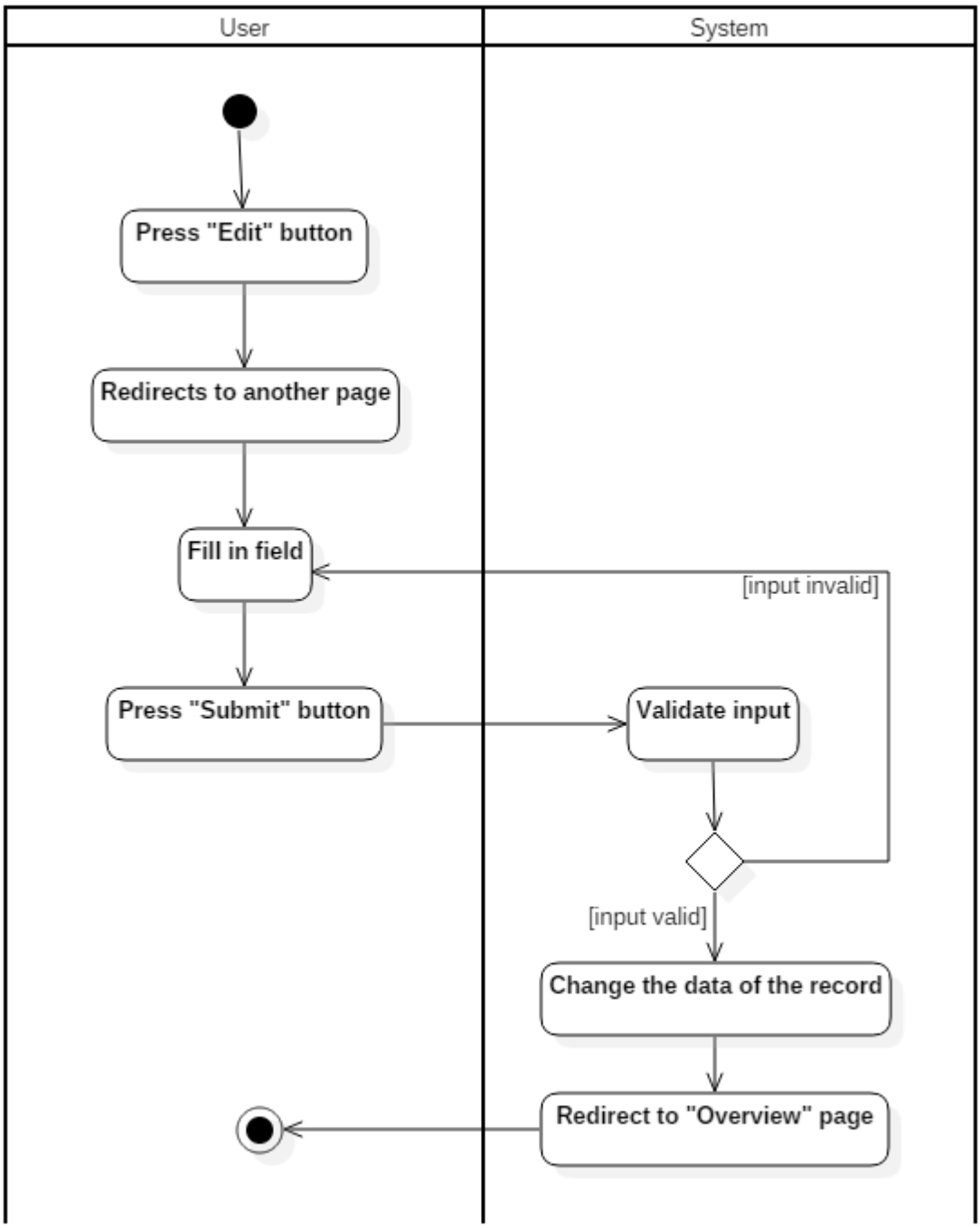


Figure 29. Activity Diagram "Edit Inputted Record"

3.2.11.3 Sequence Diagram: Edit Inputted Record

Figure 30. Sequence Diagram "Edit Inputted Record"

3.2.11.4 Object Collaboration Diagram: Edit Inputted Record

Figure 31. Object Collaboration Diagram "Edit Inputted Record"

3.3 Classes Description

3.3.1 Class Diagram

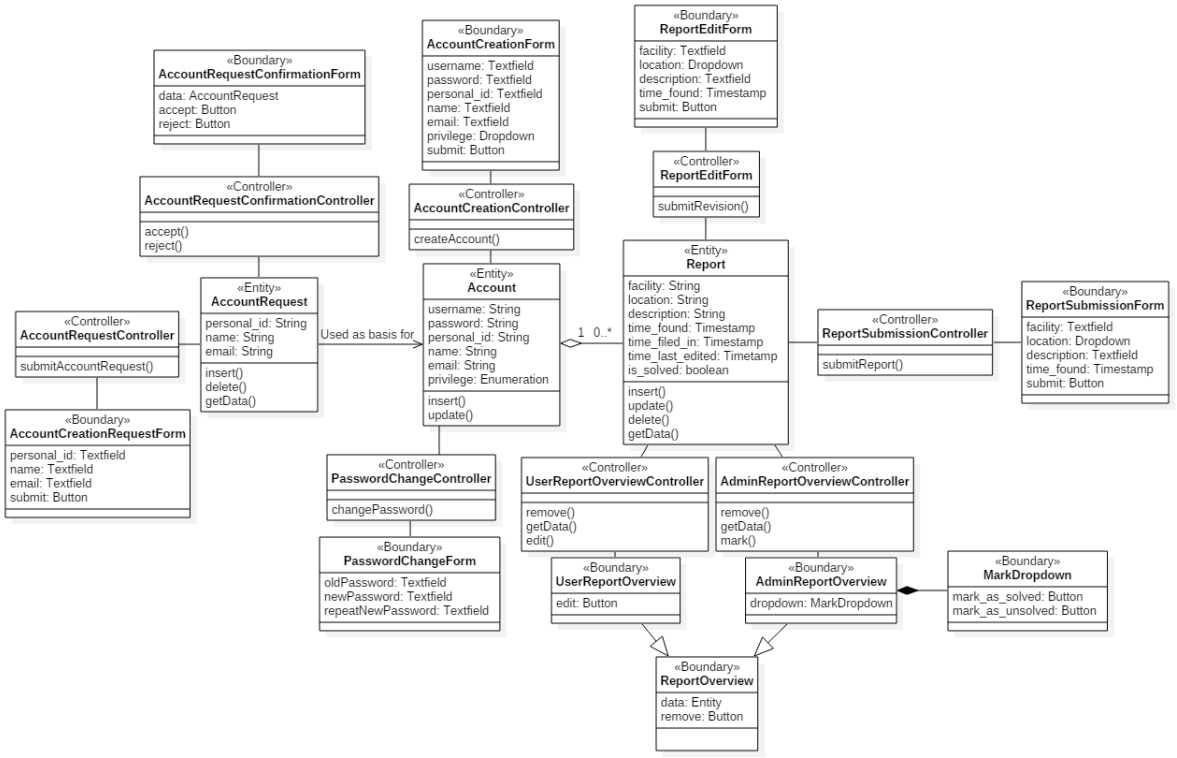


Figure 29. Class Diagram

3.3.2 Problem Domain Description

Table 3 Problem Domain Class Description

No.	Nama	Metode	Atribut	Tugas

3.3.3 Class Controller Description

Table 4 Class Controller Description

No.	Name	Method	Attribute	Responsibility
1.	AccountRequestController	submitAccountRequest()		Inserts a request regarding accoung creation request
2.	AccountRequestConfirmationController	accept() reject()		Accept or reject any account request
3.	AccountCreationController	createAccount()		Creates a new record of account with given data
4.	ReportEditForm	submitRevisionon()		Updates the state of one record
5.	ReportSubmissionController	submitReport()		Inserts the record of report
6.	Control Report Submission	submitReport()		Inserts the record of report
7.	UserReportOverviewController	remove() getData() edit()		Does general bookkeeping of existing reports with limited privilege, also as navigator
8.	AdminReportOverviewController	remove() getData() mark()		Does general bookkeeping of existing reports with full privilege

9.	PasswordChangeController	changePassword()		Manages password changing

3.3.4 Entity Class Description

Figure 30. Physical Data Model

Table 5 Entity Class Description

No.	Name	Attribute	Method	Responsibility
1.	Report	facility: String location: String description: String time_found: Timestamp time_filed_in: Timestamp time_last_edited: Timestamp is_solved: boolean	insert() update() delete() getData()	Manages reports i.e. insertion, removal, etc.
2.	Account	username: String password: String personal_id: String name: String email: String privilege: Enumeration	insert() update()	Serves as authentification and authorization
3.	AccountRequest	personal_id: String name: String email: String	insert() delete() getData()	A request such that an account is made using information given

3.3.5 Boundary Class Description

Table 6 Boundary Class Description

No.	Name	Attribute	Method	Responsibility
1.	AccountRequestConfirmationForm	data: AccountRequest accept: Button reject: Button		Interface to confirm account request
2.	AccountCreationForm	username: Textfield password: Textfield		Interface to create account

		personal_id: Textfield name: Textfield email: Textfield privilege: Dropdown submit: Button		
3.	ReportEditForm	facility: Textfield location: Dropdown description: Textfield time_found: Timestamp submit: Button		Interface to edit existing report
4.	ReportSubmissionForm	facility: Textfield location: Dropdown description: Textfield time_found: Timestamp submit: Button		Interface to make a new report
5.	AccountCreationRequest Form	personal_id: Textfield name: Textfield email: Textfield submit: Button		Interface to make a request for account creation
6.	PasswordChangeForm	oldPassword: Textfield newPassword: Textfield repeatNewPassword: Textfield		Interface to change password
7.	ReportOverview	data: Entity remove: Button		General class of ReportOverview
8.	UserReportOverview	edit: Button		Interface to see the overview of user's existing report
9.	AdminReportOverview	dropdown: MarkDropdown		Interface to see the overview of existing report
10.	MarkDropdown	mark_as_solved: Button mark_as_unsolved: Button		Component for marking the report

3.4 Data Flow Diagram

Figure 31. Data Flow Diagram

3.5 Non Functional Requirement

Table 7 Non Functional Requirement Description

SKPL-Id	Parameter	Requirement
SKPL-N01	Ergonomy	The presentation is responsive and easy to understand. System also provides visual aspect to differentiate between solved

		reports and unsolved reports
SKPL-N02	Response time	Time needed to retrieve the data is noticeably small
SKPL-N03	Security	Authorization is done based on the privilege the user's account has. Passwords are managed very securely
SKPL-N04	Language	Uses Indonesian language
SKPL-N05	Efficiency	Data filtration is to be done efficiently

3.6 Construction Limitation

None.

3.7 Requirements Summary

3.7.1 Functional Requirement Summary

Table 8 Functional Requierement Summary

SKPL-Id	Description
SKPL-F000	User can input the information about the broken facility
SKPL-F001	User can remove the inputted record regarding the broken facility
SKPL-F002	User can edit the record regarding the broken facility
SKPL-F003	User can see whether or not the facility has been taken care of
SKPL-F004	System gives a thorough overview about the report coming in such as what thing is broken, where the object can be found, how severe the damage etc.
SKPL-F005	Staff can filter the records using the time when the report filed in
SKPL-F006	Staff can remove the record regarding the broken facility
SKPL-F007	Staff can mark a record as solved or unsolved
SKPL-F008	System supports creating account for both user and staff
SKPL-F009	System supports changing password for both user and staff
SKPL-	System able to generate new password for user and staff

F010	

3.7.2 Non Functional Requirement Summary

Table 9 Non Functional Requirement Summary

SKPL-Id	Description
SKPL-NF001	Uses username and password
SKPL-NF002	Guarantees that every account has distinct username