

Waste management system

Table of content

- **Introduction**
 - .1 Purpose
 - .2 Scope
 - .3 Objective
 - .4 Assumption and Dependencies
 - .5 Overview
 - .6 General section
 - 1.6.1 Problem statement
 - 1.6.2 Introduction
 - 1.6.3 Drawbacks of existing System
 - 1.6.4 Advantage of existing System
- **Overall Description**
 - .1 Product Perspective
 - .2 Product Functions
 - .3 Operating Environment
 - .4 Assumptions and Dependencies
- **System Features**
 - .1 Functional Requirements
- **External Interface Requirements**
 - .1 User Interface
 - .2 Hardware Interface
 - .3 Software Interface
 - .4 Communication interface
 - .5 Performance Requirements
- **Non-functional Requirements**
 - .1 Performance Requirements

.2 Operation scenario

Waste management system

Introduction

1.1 Purpose

The major purpose of project is to build capacity of practitioners and decision makers to guide and handhold them to plan, design and implement Online WMITS including collection, transportation ,inspection and treatment in a city/ geographical area and country. Online WMITS System for awareness, inspection and manage the waste material by using latest system. Simulate and offer aid for system for the benefit of manufacturers and users. Help in the improvement of standards, terminology equipment's, methods and implementation practices in the field of WMITS The main purpose of WMITS is to help automate the entire process that the Department of Environmental Quality (DEQ) Waste Management Division (WMD) staff members perform throughout an inspection. The goals of WMITS are:

- To minimize the time span of any inspection
- To minimize the amount of paper work required
- To provide a searchable database of all past inspections
- To provide an automated channel for the public to request information (under Freedom of Information Act)

1.1 Scope

Eventually, multiple users will be using the product simultaneously. Therefore, concurrent connection will be an issue for implementation. In addition, this is a pilot product that hopefully, if successful, can be used in other locations as well. This leads to issues about future support for a larger user base.

1.2 Objective

Put it in a simple way, a good product will be work perfectly, doing the right thing at the right time. To do that, the software has to go through a series of tests before its final release. Error free software is

extremely difficult to achieve. After all, nothing is perfect. Especially for software developed in a short time frame. But high quality can be achieved with a detailed test specification. All (or least most) of the test case will be listed, the development team will follow it step by step, item by item, to test all the necessary objects, data flows, limits, boundaries, constraints of the software.

1.3 Assumption and Dependencies

- A way in which DEQ could add new facilities to the database. ·
- A way in which DEQ could generate electronic checklists. ·
- A search on all electronic checklists. ·
- A way in which they could generate letters to be sent out to facilities based on inspection results.
- A way in which all letters and checklists could be stored electronically.
- A way to search for existing facilities. · A way to print blank checklists and staff reports.
- A way in which they could view data which was entered into the database prior to our software.
- DEQ wanted a product that would allow them to easily add new checklists and letters or change existing checklists and letters.

1.4 Overview

- Minimize the time span of any inspection
- Help to reduce and recycle the e-waste
- Reach out to the Organization for waste management
- minimize the amount of paper work required
- provide a searchable database of all past inspections

1.6.1 Problem Statement

E-waste poses a huge risk to humans, animals, and the environment. The presence of heavy metals and highly toxic substances such as mercury, lead, beryllium, and cadmium pose a significant threat to the environment even in minute quantities.

1.6.2 Introduction

Waste Management Inspection Tracking System (WMITS) helps to generate complaint and provide appropriate response from the administration this will help to build inspection request where inspector will visit and generate checklist and assign to the administration and administration will assign organization for the waste management

1.6.3 Drawbacks of existing System

- **Ineffective Legislation**

There is absence of any public information on most SPCBs/PCC websites. 15 of the 35 PCBs/PCC do not have any information related to WMITS, their key public interface point. Even the basic WMITS Rules and guidelines have not been uploaded. In absence of any information on their website, especially on details of recyclers and collectors of waste, citizens and institutional generators of waste are totally at a loss to deal with their waste and do not know how to fulfil their responsibility. So, there is failure in successful implementation of WMITS

- **Reluctance of authorities' involved**

Lack of coordination between various authorities responsible for waste management and disposal including the non-involvement of municipalities.

1.6.4 Advantage of System

- Saves valuable time and resources during an incident
- Allows more efficient and effective waste management decision-making during an incident
- Encourages stakeholders (e.g., state, local, tribal and territorial governments; owners of private storage, treatment and disposal facilities; residents) to work together before an incident occurs
- Boosts the community's resiliency, resulting in a quicker and less costly recovery to its pre-incident state
- Enhances communities' adaptation to the waste-related impacts of climate change
- Minimally detracts from, or otherwise impacts, the broader response and recovery efforts due to the efficient implementation of waste management activities
- Minimize the time span of any inspection
- provide a searchable database of all past inspections
- minimize the amount of paper work required

Overall description

2.1 Product Perspective

- User Module : Used for issue inspection complain and track the response
- Administration Module : response the complaint and assign Inspector and assign final checklist to the organization
- Checklist module : waste classification, criteria, date, time, location
- Inspector Module: generate editable final checklist from the inspection
- Organization Module : Accept the inspection checklist and collect and manage the waste

2.2 Product Functions

- User module

- Generate complaint

- Track response

- Administration Module

- Assign inspector

- Validate checklist

- Assign organization

- Inspector Module

- Generate final editable checklist

- Checklist

- Inspection details

2.3 Operating Environment

The proposed software is to run on client/server model network. A client/server can deliver the better performance than the file server system because a client application and database server work together to split processing load of applications (thus the term distributed processing). The server manages the database among the number of clients, while the client send, request, and analyze the data entry form with small specific data set, such as rows in a table not file as in the file server system. A database server is intelligent enough so that it lock and return only the rows

Client request, which ensure concurrency, minimize the network traffic and increase the system performance.

System Feature

1.1 Functional Requirements

- The system should be secured enough to rely upon.
- Users should not be allowed to delete/modify any records.
- Users should not be allowed to take financial reports.
- Every users report should keep the tracks of user inputting the record.
- System should provide facility of exporting its data in text format.
- System should be able to integrate with its Phase II developments.
- Details of operations conducted in every screen
- Data handling logic should be entered into the system
- It should have descriptions of system reports or other outputs
- Complete information about the workflows performed by the system
- It should clearly define who will be allowed to create/modify/delete the data in the system
- How the system will fulfil applicable regulatory and compliance needs should be captured in the functional document

1. New User Registration

When user want to generate inspection complaint they have to first register themselves by providing their details. . After completion of registration a User ID and Password will be provided to the user. They can use that id and password for login

2. User Log in

A registered user can login to the system by using his User ID and Password provided after registration. After Successful login Home page is shown. In home page user can generate complaint by providing appropriate complaint detail with its date and time and can track response

3. Complaint

- Inspection complaint
- Date
- Time
- Location

4. Administration

Validate the complaint and assign inspector

5. Inspector

Edit checklist and submit final checklist to the organization

6. Organization

Approve or decline the checklist and manage collect and recycle the waste

External Interface Requirements

4.1 User Interface

The application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. It has been required that every form's interface should be user friendly and simple to use. Besides, there should be facility of accessing the system through keyboard along with the mouse.

- Front end tools:

Android

JavaScript

- Back end tools:

MySQL

SQLite

4.2 Hardware Interfaces

User side

Windows	Windows 10
RAM	512 MB

Hard disk	1 GB
processer	1.0 GHz

Server side

Windows	Windows 9 ,Windows 10
RAM	1 GB
Hard disk	3 GB
processer	2 GHz

Administration side

ODEQ - Microsoft Visual Basic [Design] Window

This is the main window that we will use to access the database using Visual Basic. We will have several different drop-down menus in this window. File, Facility, Inspection, approve, Reports, Maintenance and Help are the drop down menu that willbe available in this window we will try to use all the menus and then different optionsavailable in each of the window

4.3 Software Interfaces

Client side

Web browser	Chroma, IE
Operating System	Android, Windows

Server side

Web server	IIS 7.5
Frame work	PHP,MYSQL
Database work	MS SQL SERVER
Web browser	Chroma , IE
Operating System	Windows server 2010,android

4.4 Communication interface:

- This Application supports android OS.
- User will get response about the inspection complaint.
- Administration will get response from organization for the checklist approval.
- Inspector will give validation response from administration for the final checklist

4.5 Performance requirements:

- Response Time: Response time will be less than 1s.
- Security: The details of the students will be kept secured by providing password facility and while updating details after OTP validation.
- Efficiency: response for the inspection can track and update in time

Non-functional Requirements

5.1 Performance Requirements

- It also needs a security system to prevent data.
- System need to update and delete the record.
- System needs store information about new entry of Customer.
- Every unsuccessful attempt by a user to access an item of data shall be recorded on an audit trail.
- Privacy of information, the export of restricted technologies, intellectual property rights, etc. should be audited.
- The system must meet Web Content Accessibility Guidelines
- How often does the system experience critical failures? And how much time is it available to users against downtimes?

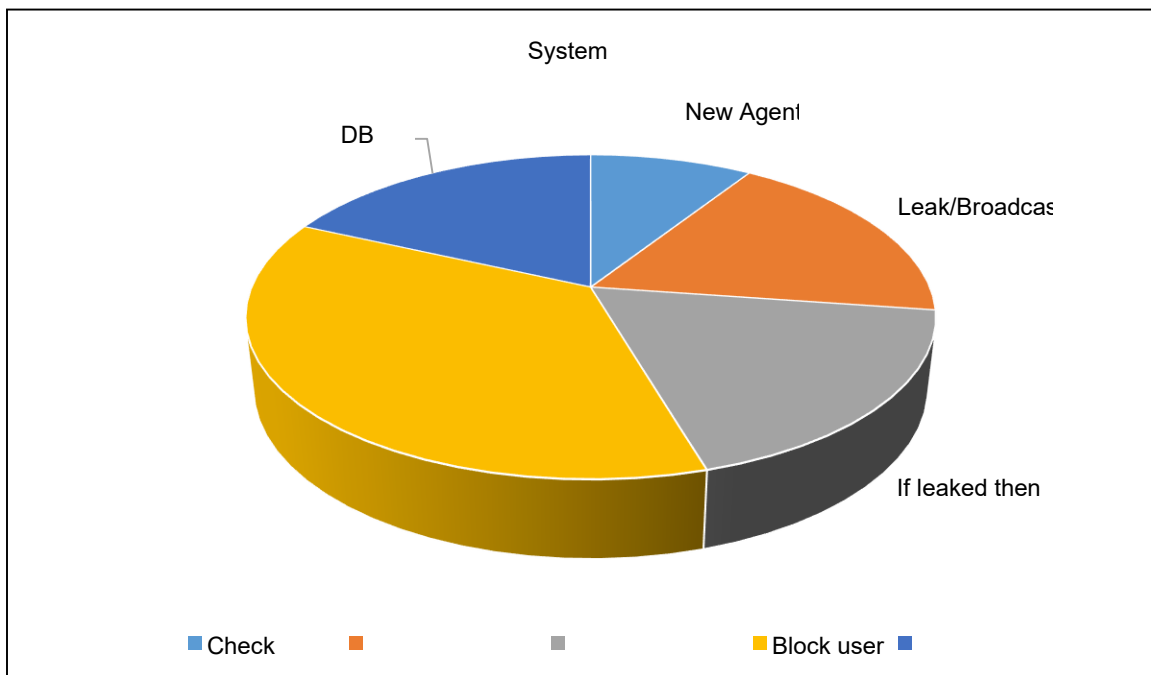
- Which hardware, operating systems, browsers, and their versions does the software run on? Does it conflict with other applications and processes within these environments?
- How fast does the system return results? How much will this performance change with higher workloads?
- How easy is it for a customer to use the system?
- Does the system match local specifics?
- How fast a software system or its particular piece responds to certain users' actions under certain workload. In most cases, this metric explains how much a user must wait before the target operation happens (the page renders, a transaction is processed, etc.) given the overall number of users at the moment. But it's not always like that. Performance requirements may describe background processes invisible to users, e.g. backup. But let's focus on user-centric performance.

5.2 Operational scenario

- User logged in but he/she forgot their password.
- User forgot to check the response
- Administrator may have entered wrong details of the inspector.
- Inspector may not assign

5.3 Preliminary Schedule

- 1.1 We gave 1 month to discuss about software, cost, deadlines, risk analysis etc.
- 1.2 2 month for requirement gathering from user and then by analysing that we made SRS document and verified that from user.
- 1.3 2 months for Data collection like details of different companies which wanted to involve.
- 1.4 3 months to Design software according to user requirements and
- 1.5 2 months for implementation and testing for final product.



5.4 Safety Requirements

- How are the system and its data protected against attacks?

- Particularly important where sensitive information such as personal details or financial data is being handled, **security** includes other NFRs such as **confidentiality** and **authentication** to ensure this information is protected by default.

5.5 Security Requirements

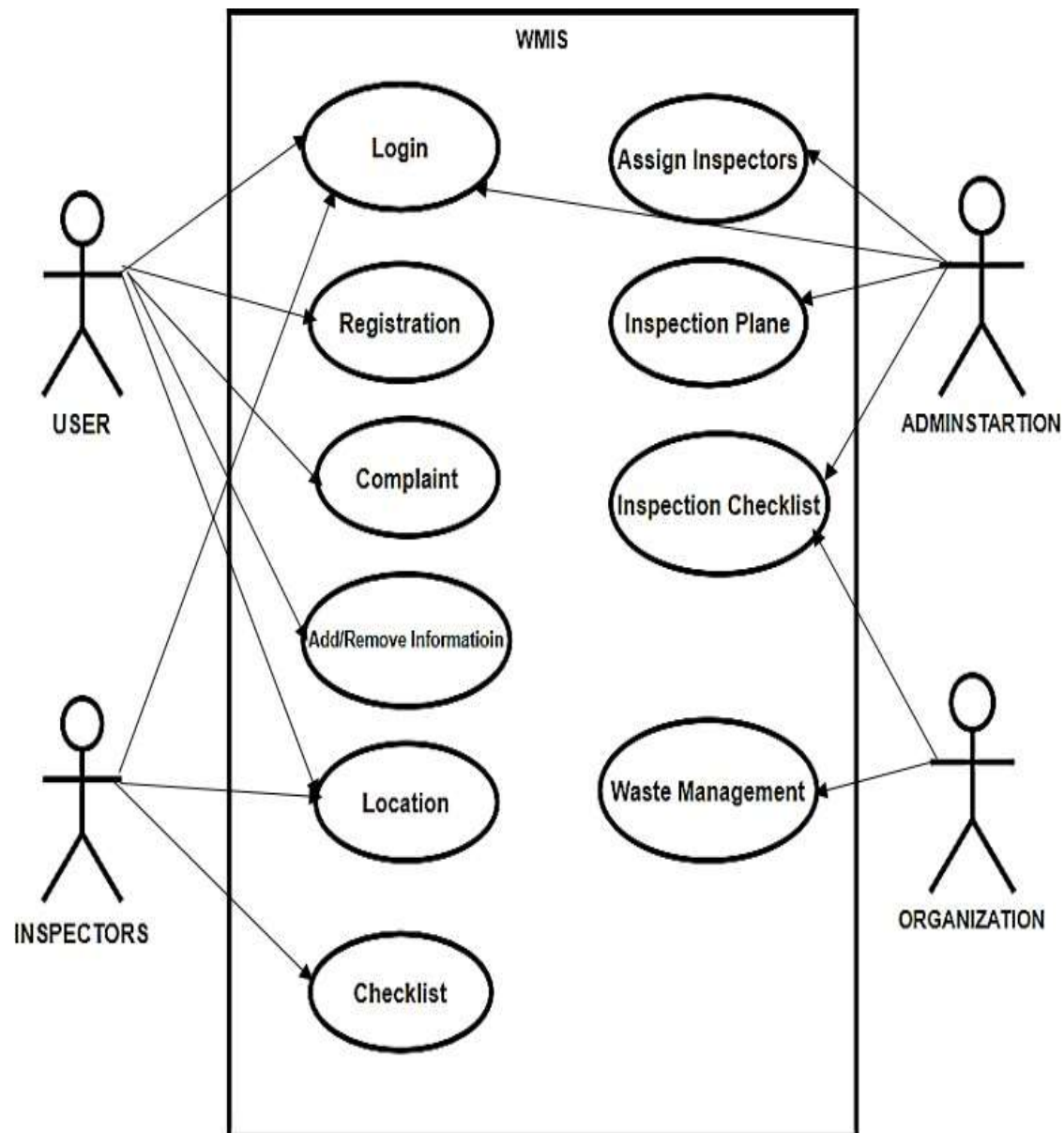
- Limitation on who can access what data
- What data will be encrypted whenever authentication will be needed
- Secured database

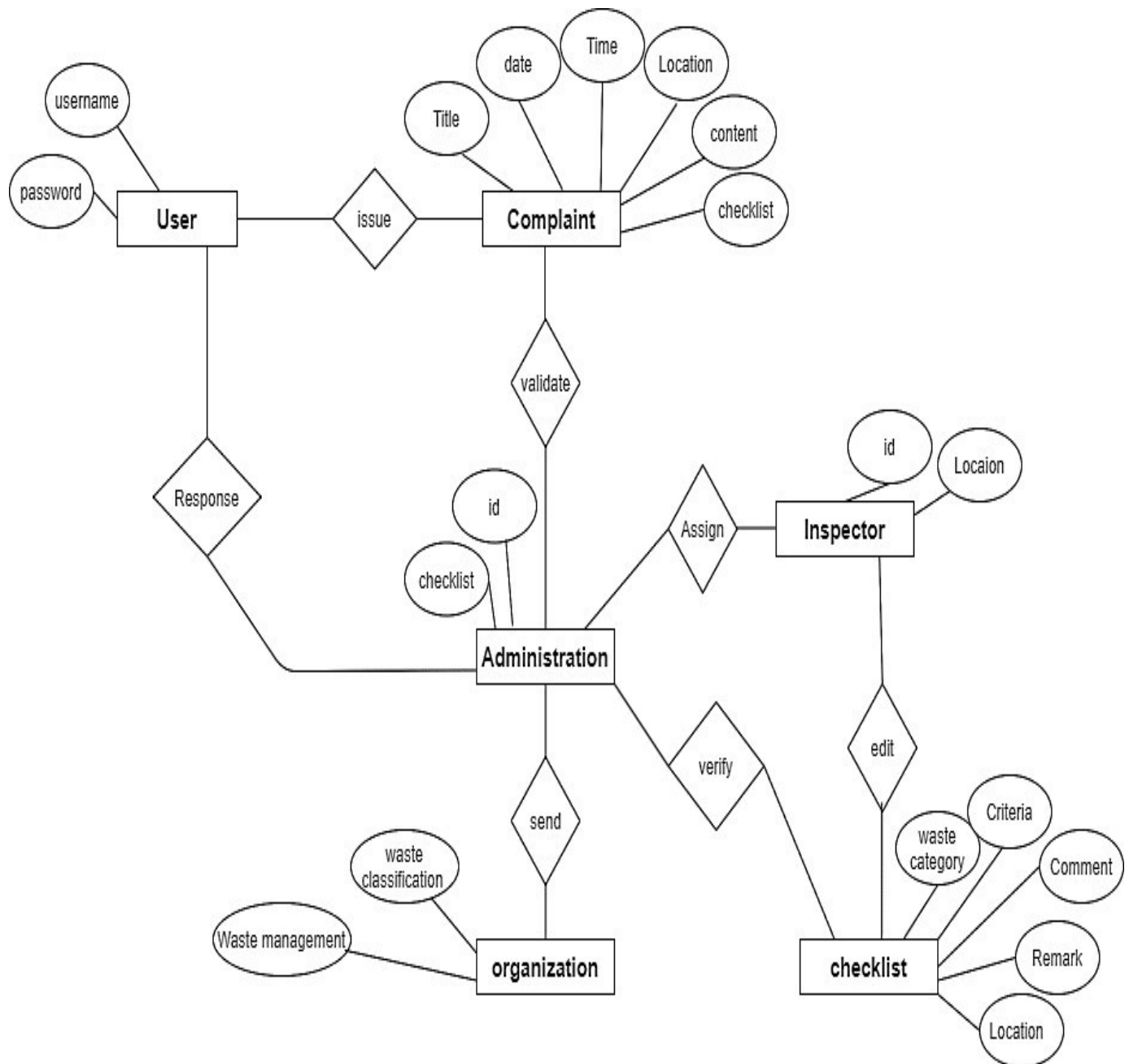
5.6 Reference

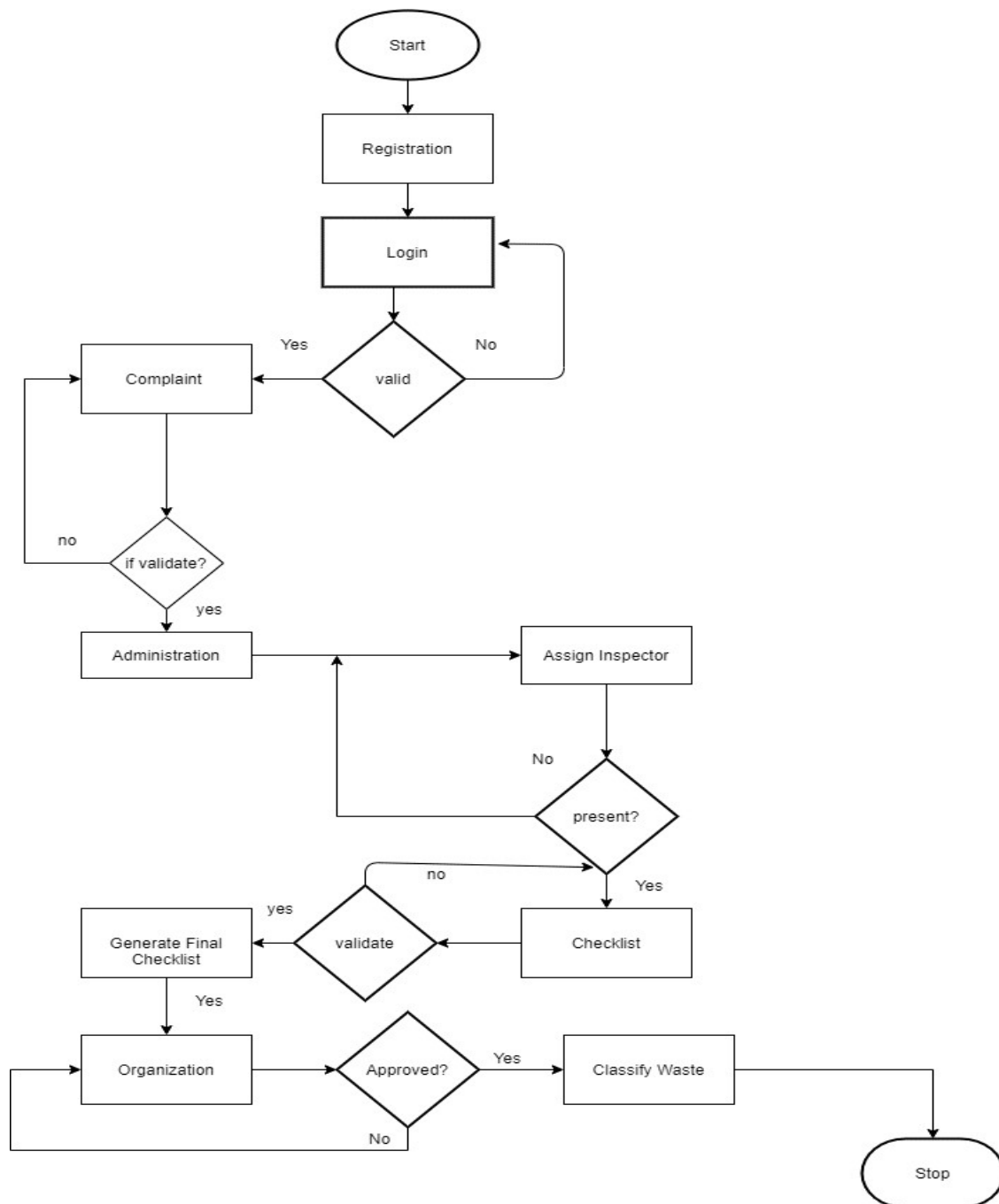
[Waste Management Inspection Tracking System by Zakriya Maturidi \(prezi.com\)](#)

[Waste Management Benefits, Planning and Mitigation Activities for Homeland Security Incidents | US EPA](#)

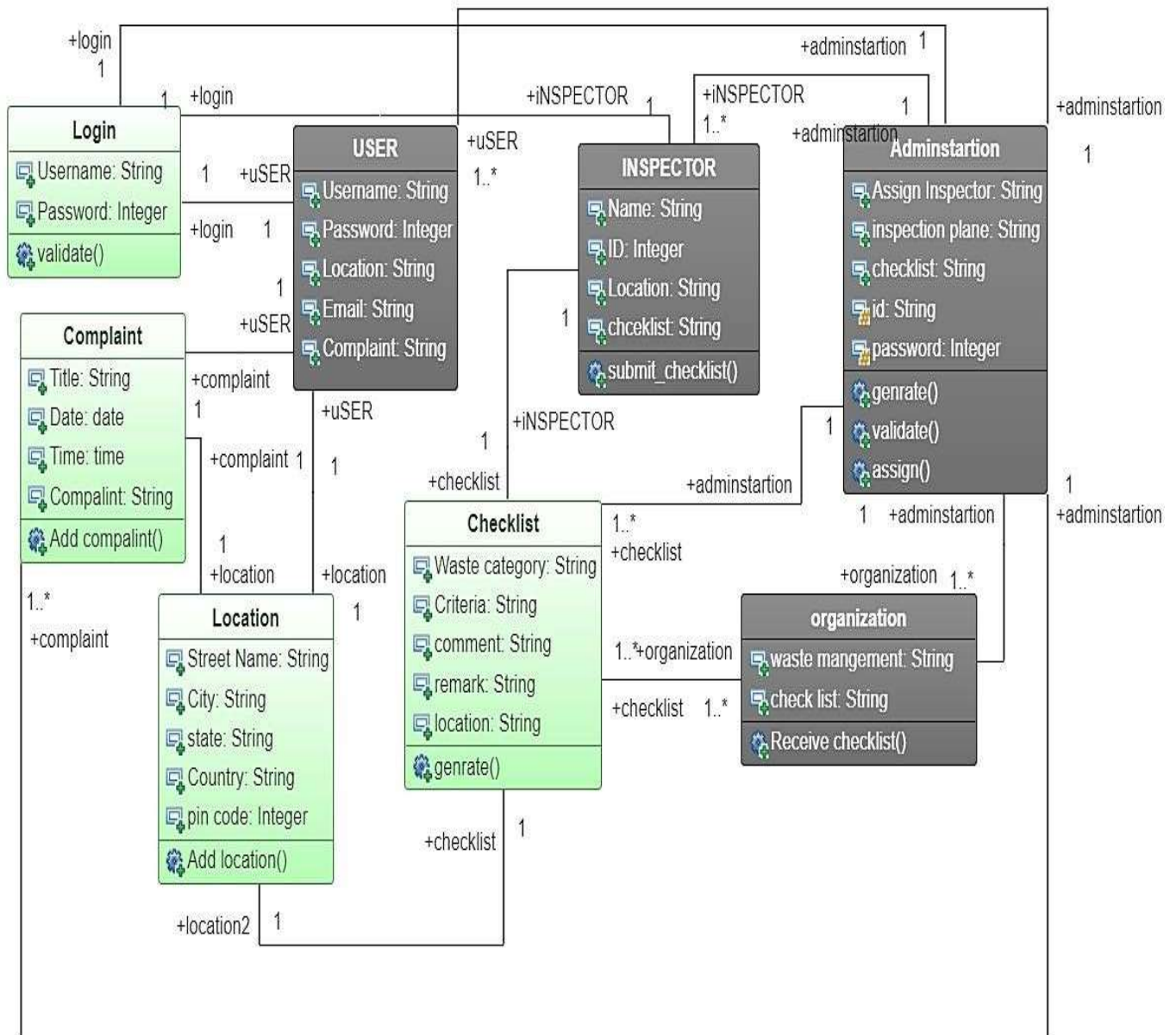
[Waste Management Software | Waste Tracking Software - Gensuite](#)

USE CASE DIAGRAM

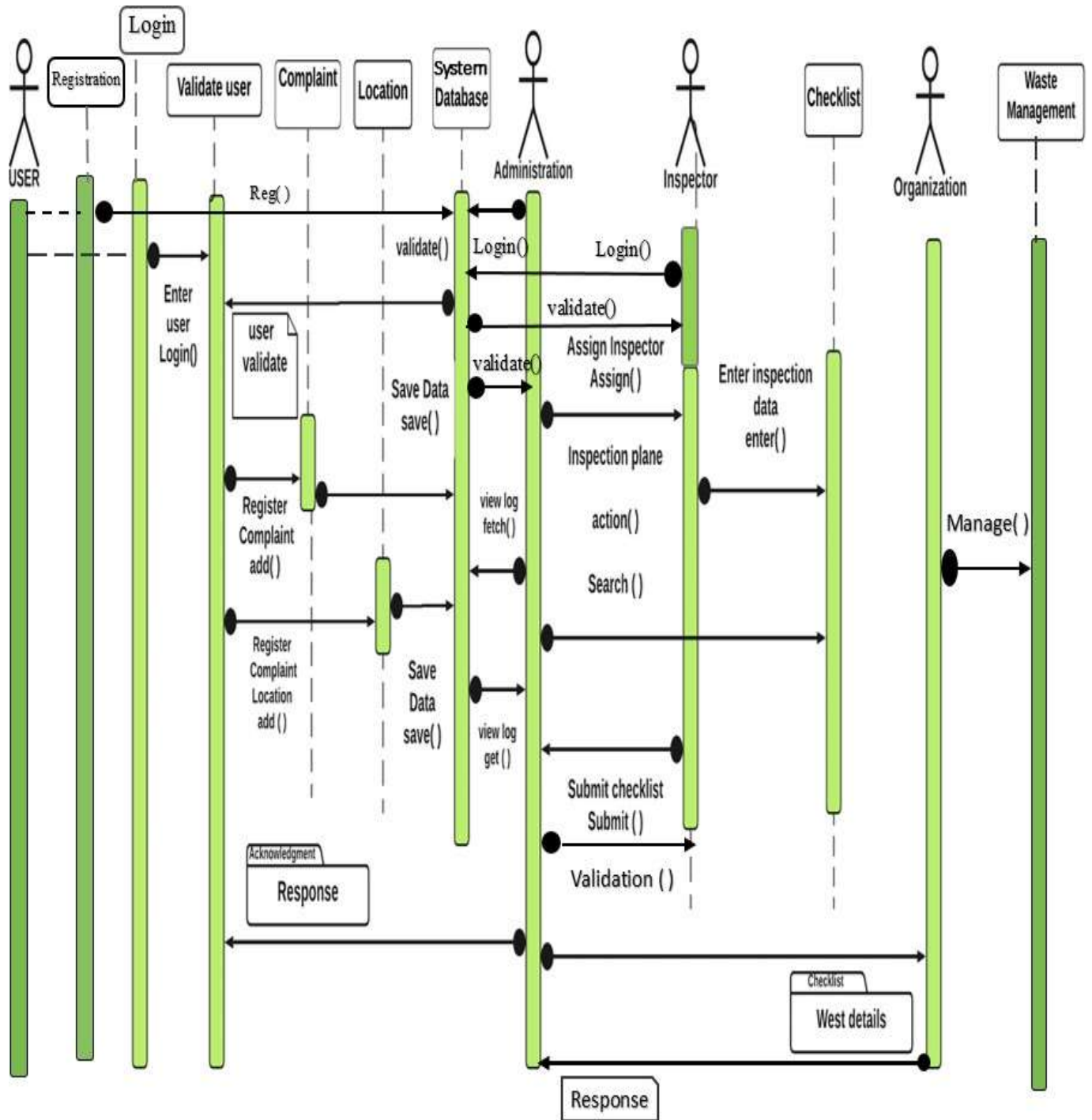
ENTITY RELATIONSHIP DIAGRAM

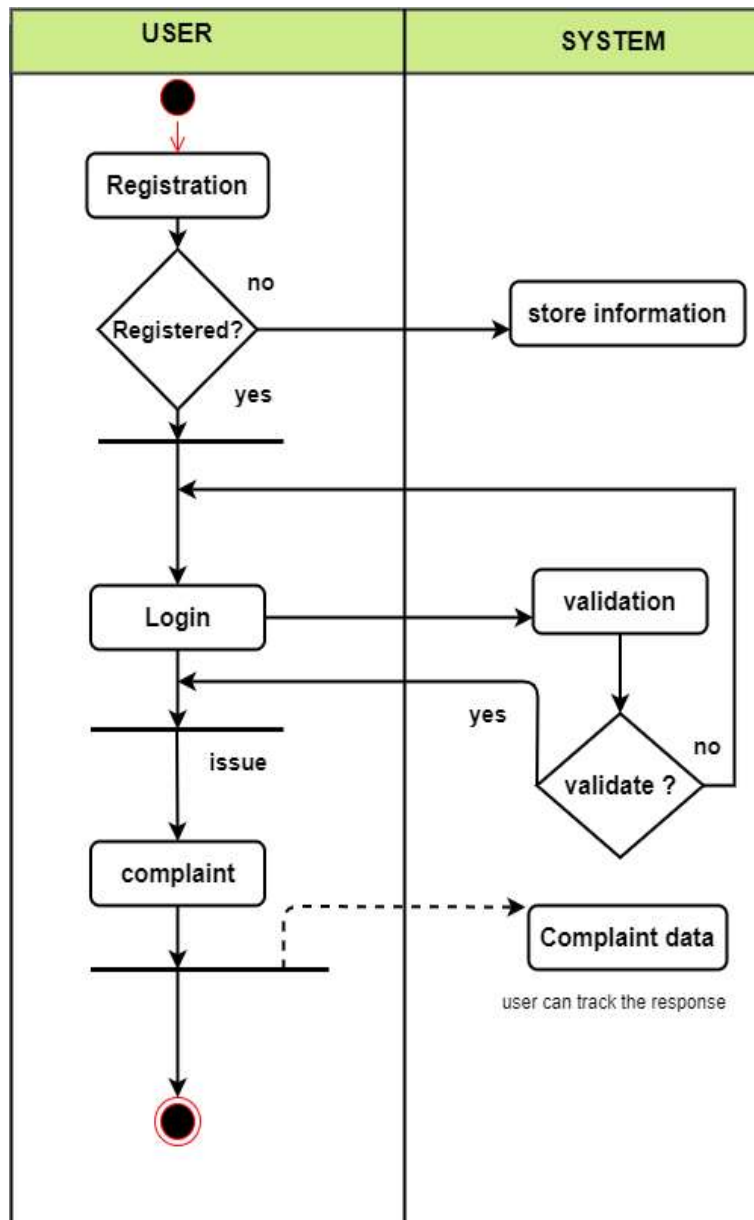
Flowchart

CLASS DIAGRAM

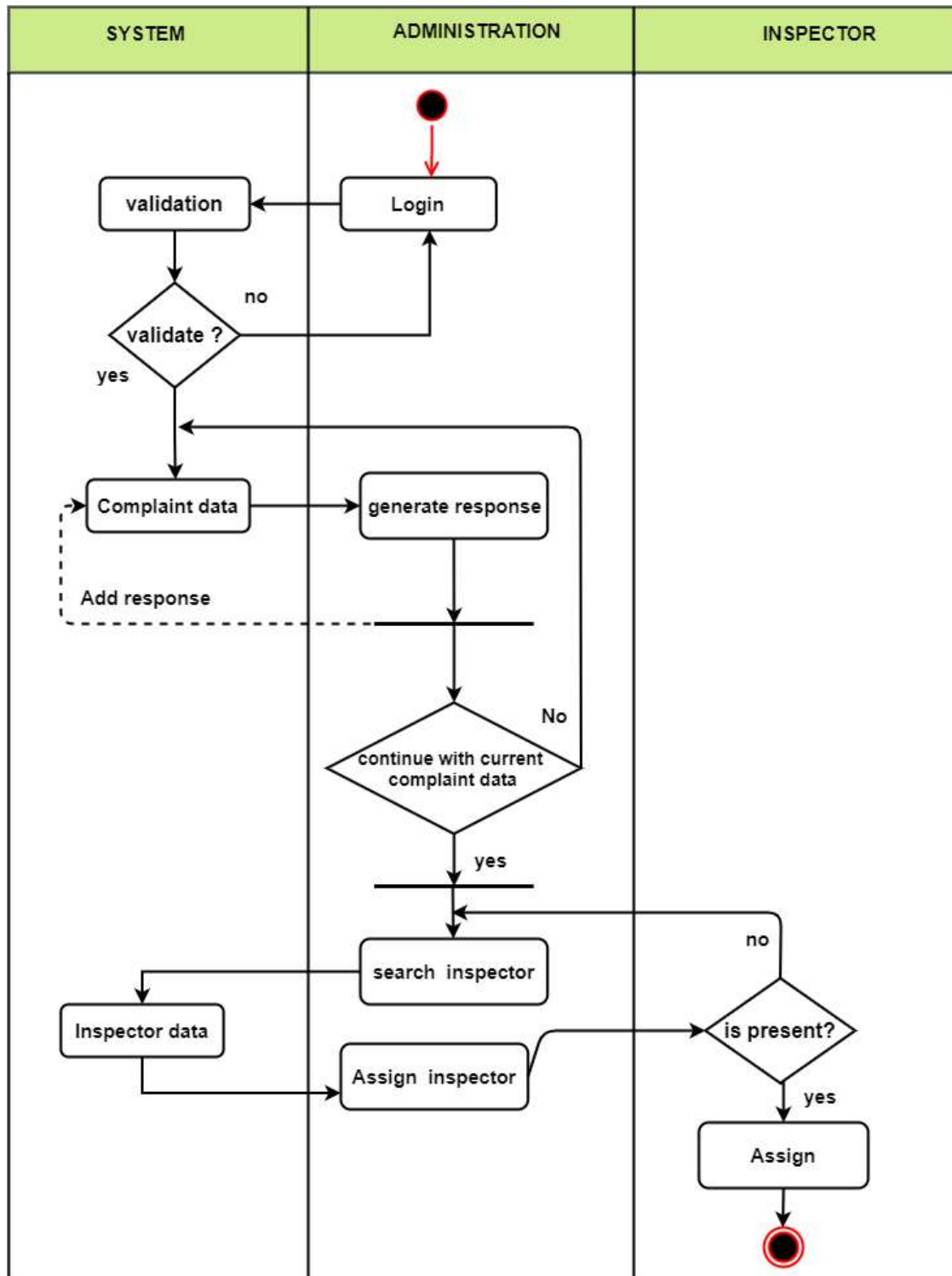


SEQUENCE DIAGRAM

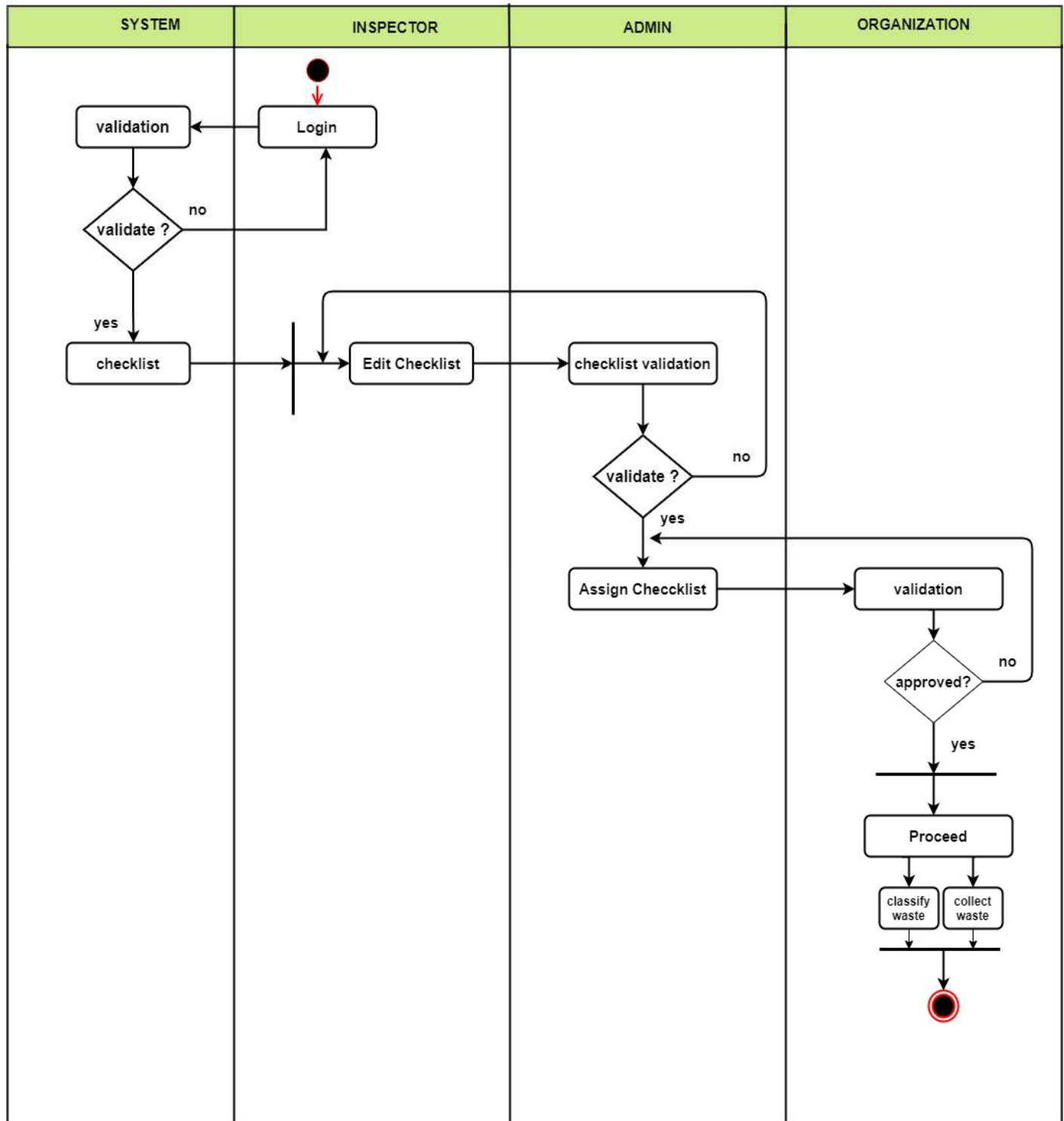


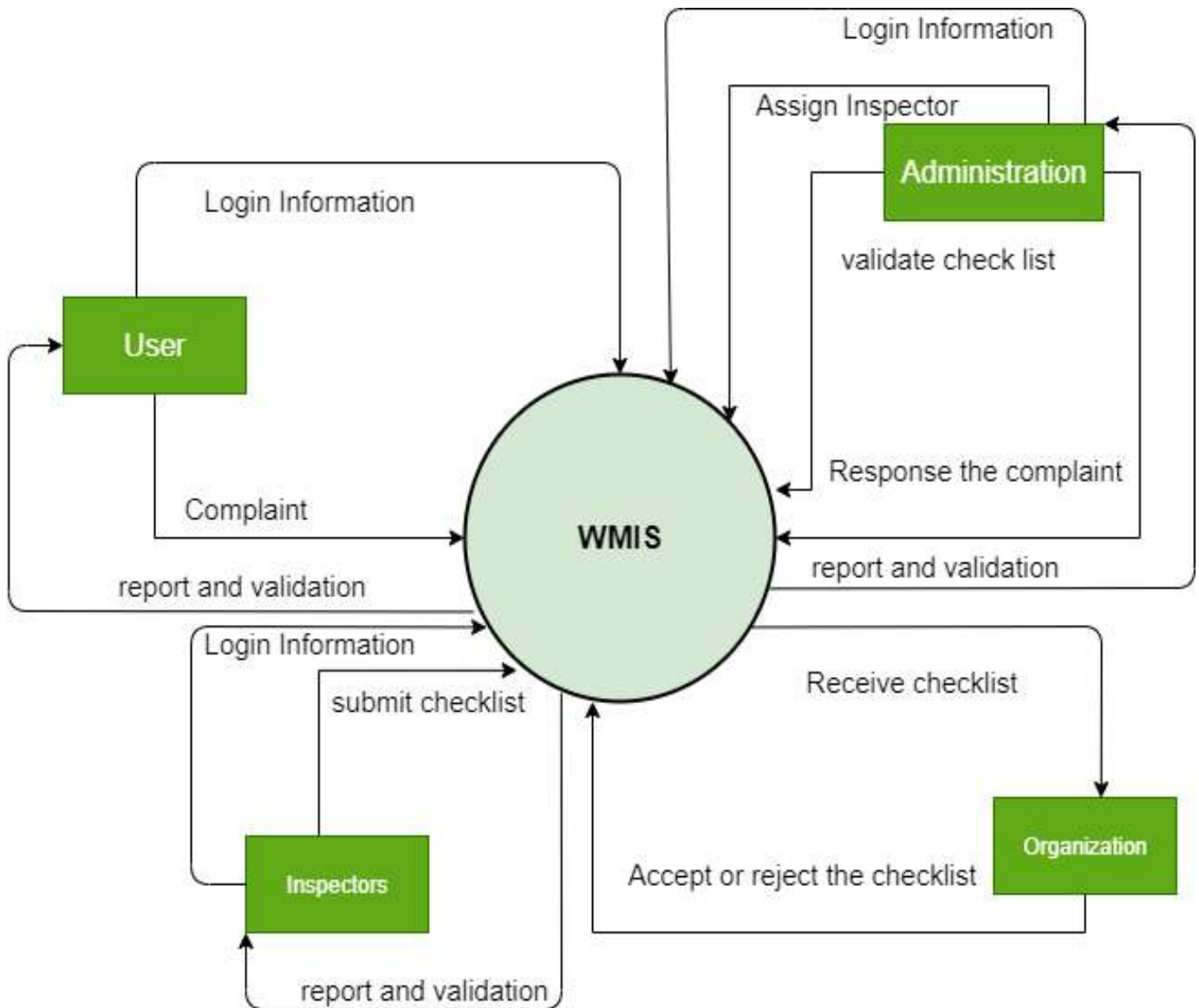
ACTIVITY DIAGRAM**USER ACTIVITY DIAGRAM**

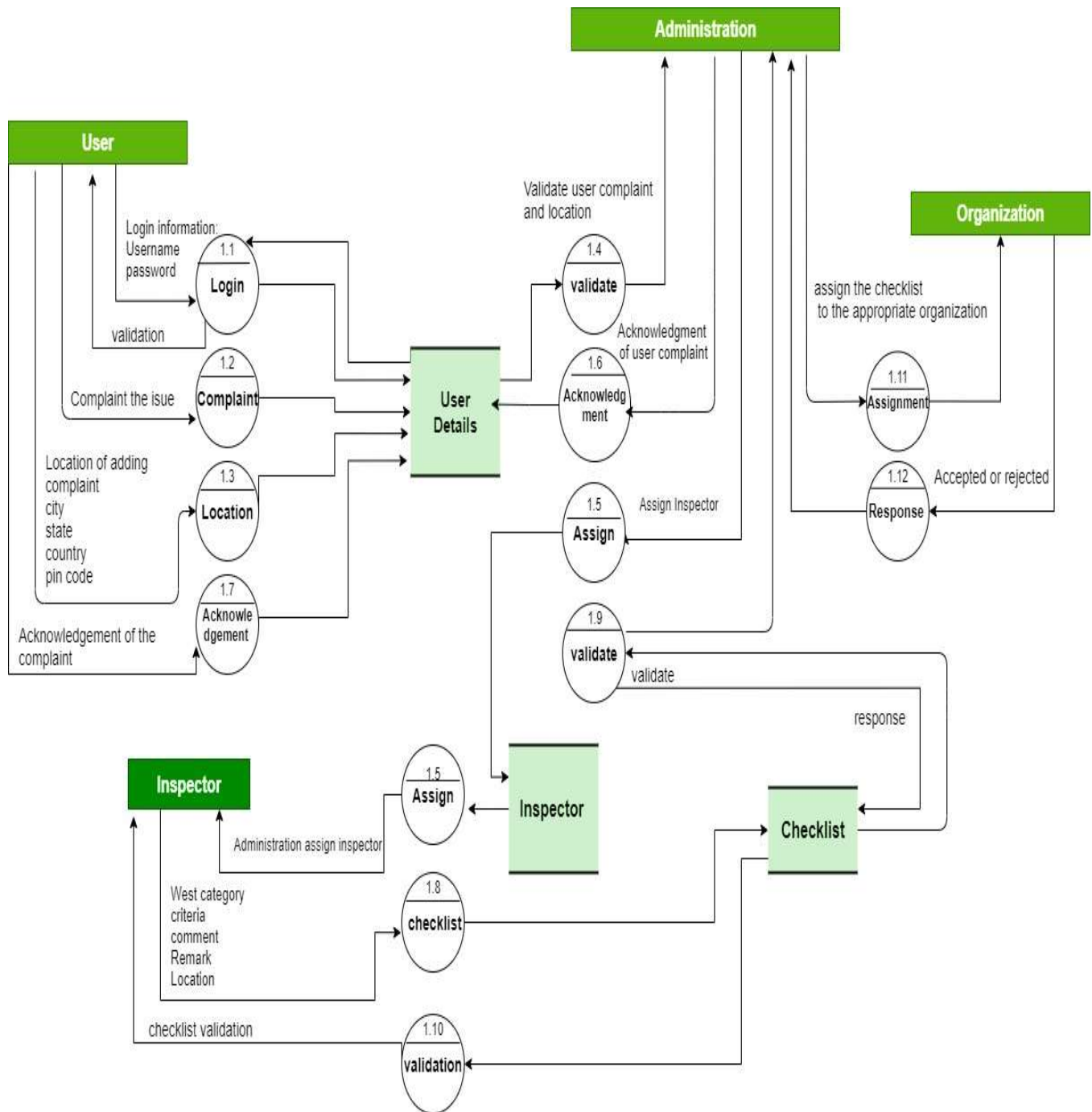
ADMINISTRATION ACTIVITY DIAGRAM



INSPECTOR to ORGANIZATION ACTIVITY DIAGRAM



Data flow diagram**CONTEXT LEVEL DFD****LEVEL 1 DFD**



Test case design

Test Case 1: Verify User Login

# TS1						
Title		Verify User Login				
Description		To test the different scenarios that might arise while an user trying to login				
#	Summary	Dependence	Pre-condition	Post-Condition	Execution Steps	Expected Output
TC1	Verify that user already registered with the WMITS is able to login with correct user ID and password		UserID 149405 is a registered user of WMITS; user's password is this_is_password	User is logged in	<ul style="list-style-type: none"> – Type in UserID as 149405 – Type in password this_is_password – Click on the 'Login' button 	"Home" page for the user is displayed
TC2	Verify that an unregistered user of WMITS is unable to login		UserID 149405xx is not a registered user of WMITS	User is not logged in	<ul style="list-style-type: none"> – Type in UserID as 149405xx – Type in password whatever 	The "Login" dialog is shown with a "Login failed! Check your user ID and password" message

					<ul style="list-style-type: none"> – Click on the 'Login' button 	
TC3	Verify that user already registered with the WMITS is unable to login with incorrect password		UserID 149405 is a registered user of WMITS; user's password is this_is_password	User is not logged in	<ul style="list-style-type: none"> – Type in UserID as 149405 – Type in password whatever – Click on the 'Login' button 	The "Login" dialog is shown with a "Login failed! Check your user ID and password" message
TC4	Verify that user already registered with the WMITS is unable to login with incorrect password given twice consecutively	TC3	This test case is executed after execution of TC3 before executing any other test case	User is not logged in	<ul style="list-style-type: none"> – Type in UserID as 149405 – Type in password whatever2 – Click on the 'Login' button 	The "Login" dialog is shown with a "Login failed! Check your user ID and password" message
TC5	Verify that user already registered with the WMITS is unable to	TC4	This test case is executed after execution of TC4 before executing	User is not logged in	<ul style="list-style-type: none"> – Type in UserID as 149405 – Type in password whatever3 	The "Login" dialog is shown with a "Login failed! Check your user ID and password" message; the security question

	login with incorrect password given thrice consecutively		any other test case		– Click on the 'Login' button	and input box for the answer are displayed
TC6	Verify that a registered user can login after three consecutive failures by correctly answering the security question	TC5	This test case is executed after execution of TC5 before executing any other test case. Answer to the security question is my answer	Email sent containing new password. The email is expected to be received within 2 minute.	– Type in the answer as my answer – Click on the 'Email Password' button	Login dialog is displayed; an email containing the new password is received
TC7	Verify that a registered user's account is blocked after three consecutive failures and answering the security		Execute the test cases TC3, TC4, and TC5 once again (in order) before executing this test case	User account has been blocked	– Type in the answer as not_my_answer – Click on the 'Email Password' button	The message "Your account has been blocked! Please contact the administrator." appears

	question incorrectly			
--	----------------------	--	--	--

Test Case 2: Administration Login

# TS2						
Title		User Registered Login				
Description		To test that user trying to registered complaint				
#	Summary	Dependancy	Pre-condition	Post-Condition	Execution Steps	Expected Output
TC1	Verify that an administration of WMITS is unable to login		administration ID 123 is not registered	Administration not logged in	<ul style="list-style-type: none"> Type in ID as 123 Type in password whatever2 Click on the 'Login' button 	<p>The "Login" dialog is shown with a "Login failed! Check your ID and password" Message</p>
TC2	Verify that an administration of WMITS is unable to login			Administration not logged in	<ul style="list-style-type: none"> Type in ID as 149409 Type in password whatever2 Click on the 'Login' button - 	<p>The "Login" dialog is shown with a "Login failed! Check your ID and password"</p>

						Message
TC3	Verify that an administration of WMITS is able to login		ID 123 is a registered admin of WMITS; admin's password is pass_100	Administration logged in	<ul style="list-style-type: none"> – Type in ID as 123 – Type in password pass_100 Click on the 'Login' button 	Show home page
TC7	Verify that an administration account is blocked after three consecutive failures and answering the security question incorrectly		Execute the test cases TC1, TC2, again and again (in order) before executing this test case	Admin account has been blocked	<ul style="list-style-type: none"> – Type in ID as 123 – Type in password pass_100 Click on the 'Login' button 	The dialog is shown with a “your account has been blocked to unblock submit problem letter”

# TS3						
Title		Inspector Login				
Description		To test that user trying to registered complaint				
#	Summary	Depende ncy	Pre-condition	Post-Condition	Execution Steps	Expected Output
TC 1	Verify that inspector of WMITS is unable to login		Inspector ID 156 is not registered	Inspector not logged in	<ul style="list-style-type: none"> – Type in ID as 156 – Type in password whatever3 – Click on the 'Login' button 	<p>The "Login" dialog is shown with a "Login failed! Check your ID and password" Message</p>
TC 2	Verify that inspector of WMITS is unable to login			Inspector not logged in	<ul style="list-style-type: none"> – Type in ID as 156 – Type in password whatever3 – Click on the 'Login' button – 	<p>The "Login" dialog is shown with a "Login failed! Check your ID and password" Message</p>
TC 3	Verify that inspector of WMITS		ID 156 is a registered admin of WMITS; admin's	Inspector logged in	<ul style="list-style-type: none"> – Type in ID as 123 – Type in password pass_100 Click on 	Show home page

	is able to login		password is pass_100		the 'Login' button	
TC 7	Verify that an Inspector account is blocked after three consecutive failures and answering the security question incorrectly		Execute the test cases TC1, TC2, again and again (in order) before executing this test case	Inspector account has been blocked	<ul style="list-style-type: none"> – Type in ID as 123 – Type in password pass_100 Click on the 'Login' button 	The dialog is shown with a “your account has been blocked to unblock submit problem letter”

Test Case 4: User Complaint

# TS4						
Title		User Registered Complaint				
Description		To test that user trying to registered complaint				
#	Summary	Dependence	Pre-condition	Post-Condition	Execution Steps	Expected Output
TC 1	User register complaint		User should login with their id and add correct Location	Complaint Issued	<ul style="list-style-type: none"> – User click on the complaint option. – User can fill given Complaint form like title, date, time, content name etc. 	The "Complaint" dialog is shown with a "successfully Registered your complaint!" Track your complaint
TC 2	Verify that complaint Unable to issue		Given complaint Details by user is not valid	Complaint not Issued	<ul style="list-style-type: none"> – User click on the complaint option. – User can fill given Complaint form like title, date, time, content name etc. 	The "Complaint Response" dialog is shown with a "complaint decline try again!"
TC 3	Verify that complaint inspection		Given complaint issue by user is not validate	Complaint not Issued	<ul style="list-style-type: none"> – User click on the complaint option. – User can fill given 	The "Complaint Response" dialog is shown with a "complaint

	issue formally elaborated				Complaint form like title, date, time, content name etc.	decline try again!"
TC 4	Verify that same complaint can't issue until receive response		Given complaint already issued by the user	Complaint Issued	<ul style="list-style-type: none"> – User click on the complaint option. – User can fill given Complaint form like title, date, time, content name etc. 	The "Complaint Response" dialog is shown with a "complaint already issued wait for the response!"

Test Case 5: Administration Assign Inspector

# TS5						
Title		Administration Assign Inspector				
Description		To test that administration trying to assign inspector				
#	Summary	Dependancy	Pre-condition	Post-Condition	Execution Steps	Expected Output
TC1	Verify that Administration able to assign inspector		Inspector must be present for the complaint area location	Inspector assign for the inspection	<ul style="list-style-type: none"> – Administration find inspector by assign area and inspector id assign inspector and – schedule the inspection 	Inspector successfully assign for the inspection!

TC2	Verify that Administration able to assign inspector	TC1	Inspection schedule must be appropriate	Inspector assign	<ul style="list-style-type: none"> – Administration find inspector by assign area and inspector id assign inspector and – schedule the inspection 	Inspector Assign for the inspection failed!
------------	---	-----	---	------------------	---	---

Test Case 6: checklist validation

# TS6						
Title		Checklist validation				
Description		To test that checklist is generated properly and approved				
#	Summary	Dependancy	Pre-condition	Post-Condition	Execution Steps	Expected Output
TC1	Verify that Checklist is Approved or not		Inspector must be Edit right details	Final Check list generate	<ul style="list-style-type: none"> – Type waste category Criteria Comment Remark Location 	Checklist Approved!
TC2	Verify that Checklist unable to approve if not submit in Response time	TC1	Inspector not response the inspection check list within time	Cannot generate Final Check list	<ul style="list-style-type: none"> – Type waste category Criteria Comment Remark Location 	Checklist validation out of time do inspection again!

Test Case 7: organization response

# TS7						
Title		organization Response				
Description		To test that checklist submit to the organization and approved or decline				
#	Summary	Dependence	Pre-condition	Post-Condition	Execution Steps	Expected Output
TC1	Verify that Checklist assign and approved by the organization		Organization should welcome the approaches	Checklist submit to the organization	– Administration send final checklist to the organization	Approach Approved!
TC2	Verify that organization decline the response if checklist not appropriate	Tc1	Checklist submitted to the organization	Checklist examine	– Administration send final checklist to the organization	Approach decline!

2. Villager Telephone System

1. Introduction:

A Villager telephone system is a [telecommunications device](#) that permits two or more users to conduct a [conversation](#) when they are too far apart to be heard directly. A telephone converts [sound](#), typically and most efficiently the [human voice](#), into electronic [signals](#) that are transmitted via [cables](#) and other communication channels to another telephone which reproduces the sound to the receiving use.

2. Purpose:

The use of a telephone easily connects people in such a way that you can meet people you normally would not. Because the phone system is more personal than a written letter, it also provides the opportunity to also share and bond unlike previous generations. While the tradition of putting pen to paper declined, hearing the voice inflection during a phone call provided the groundwork for closer relationships.

3. Scope:

- This system can be used to rural people will be able to communicate other people.
- This system can be used to people can save their numbers with name.
- This System can be used to search people name with their Contact number.
- This system can be used to record information with their voice.
- This System can be used to Text Message to any person.

4. software Requirement Specification

1. User Characteristics:

1. Admin
2. Rural People
3. Central office

❖ Admin:

The administrator controls the all telephone system activities.

The admin also has a username & password through which he enters in the admin view.

It has the following:

Add or Delete a contact number.

Add or Delete subscriber details.

2. Rural People:

It contains following:

- People can add contact number and other details.
- People can search contact with name.
- People can record their voice.
- People can text message to other people.

3. Central office:

It contains following:

- These people can edit the information of people.
- These people can edit or record call details and other type of information.
- These people can record mobile tower information.

2. Functional Requirements:

R.1 Registration:

R.1.1 login:

Input: user id and password

Process: Entry in database (store temp.)

If entered input is valid.

Output: login successfully.

R.1.2 registration:

Input: user id and password

Process: generate register id and store database and check if person will same

Email available or not if not then store in database else an error generated.

R.2.1 search based on Contact Number:

Pre-condition: login & registered

Input: Contact Number

Process: fetch the matching of search option.

Output: display Contact Details.

R.2.2 search based on Name:

Pre-condition: login & registered

Input: Enter name

Process: checking whether the entered name is available or not and fetch the matching of search option.

Output: display Details.

R.3 text Message:

Pre-condition: login & registered

Input: Enter Message or Information

Process: Send The Message to other People

Output: deliver Successfully.

R.4 logout:

Pre-condition: login & registered

Input: option=logout and customer gets email id and password

Process: people logout for those account

Output: successfully logout

3.Non-Functional Requirements:

1) Usability

- This section includes all of those requirements that effect usability.
- We get the response within seconds.
- The software must have a simple, user-friendly interface so customers can save time and confusion.

2) Reliability

- The system is more reliable because of the qualities that are inherited from the chosen platform. The code built by using PHP is more reliable.

3) Supportability

- The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform.

4) Implementation

- The User Interface is a GUI developed using HTML. The main processing is done on the server side using Apache tomcat and for the programming environment PHP is used, for backend database MySQL is used.

5. Interface Requirement

Various interfaces for the product could be-

1)Login Page

2)Registration Page

3)Search Page.

4)Record Screen Page

5)Text Message Page

6. Software Interface

- PHP (front end)
- HTML
- JavaScript
- MS Word 97 or later
- Web Browser: Microsoft Internet Explorer, Mozilla, Google Chrome or later
- MySQL Server (back-end)
- Operating System: Windows XP / Windows7/ Windows Vista

7. Hardware Interface

- Processor: Intel core i3
- RAM :8GB
- Hard Disk :1TB

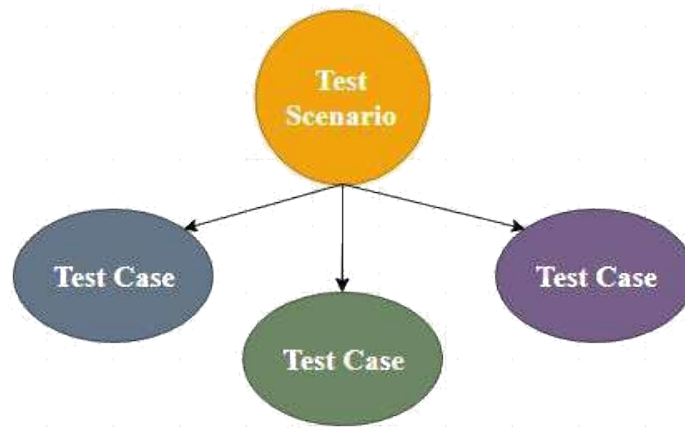
8. Design Constrain

- The System shall be built using a standard web page development tool that conforms to Microsoft's GUI standard like HTML, XML etc.

Test- Case

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.



It is an in-details document that contains all possible inputs (positive as well as negative) and the navigation steps, which are used for the test execution process. Writing of test cases is a one-time attempt that can be used in the future at the time of regression testing.

Test case gives detailed information about testing strategy, testing process, preconditions, and expected output. These are executed during the testing process to check whether the software application is performing the task for that it was developed or not.

Test case helps the tester in defect reporting by linking defect with test case ID. Detailed test case documentation works as a full proof guard for the testing team because if developer missed something, then it can be caught during execution of these full-proof test cases.

Importance:

We will write the test for the following reasons:

- To require consistency in the test case execution
- **To make sure a better test coverage**
- **It depends on the process rather than on a person**
- **To avoid training for every new test engineer on the product**
- **To require consistency in the test case execution:** we will see the test case and start

testing the application.

- **To make sure a better test coverage:** for this, we should cover all possible scenarios and document it, so that we need not remember all the scenarios again and again.
- **It depends on the process rather than on a person:** A test engineer has tested an application during the first release, second release, and left the company at the time of third release. As the test engineer understood a module and tested the application thoroughly by deriving many values.
- **To avoid giving training for every new test engineer on the product:** When the test engineer leaves, he/she leaves with a lot of knowledge and scenarios. Those scenarios should be documented so that the new test engineer can test with the given scenarios and also can write the new scenarios.

#	Summary	Dependenc	Pre - Condition	Post – Condition	Execution Steps	Expected Output
TC1	Admin register new user's number into database		Admin must be logged in from admin side	Users' number has been added to database	1) Fill up User Details(number) 2) Click on submit button	Registration successful message is shown and mobile number generate for User
TC2	Admin tries to add number of already registered User		Admin must be logged in from admin side	User number not added again	1) Fill up User Detail 2) Click on submit button	User already registered message is shown
TC3	Admin deletes/updates the number of User	TC1	User must be register number already	Users' number has been updated	1) Fill up users' detail 2) Click on update button	Data updated successfully message is shown

Airline flight Control system**• Introduction**

- .1 Purpose
- .2 Scope
- .3 Objective
- .4 Assumption and Dependencies
- .5 Overview

• Overall Description

- .1 Product Perspective
- .2 Product Functions

• Functional Requirements**• Design Constrain****• External Interface Requirements**

- .1 Hardware Interface
- .2 Software Interface
- .3 Other Requirements

• Non-functional Requirements

- .1 Performance Requirements
- .2 Operation scenario

1.1 Purpose

Airline Flight Control System aims to automate the flight operations and ticketing / seat booking and confirmation system of an Airline company. The software is providing options for viewing different flights available within a different timings for a specific day. That provide customers within facility to book ticket smoothly. The customers can modify and able to cancel the ticket for any reason. That prepare within a role and policies. The software should provide option for checking availability of the tickets. That is important for the customers to get message if the ticket unavailable. That will be displayed into customers. The customers should be noted when the change has been made or any further changes.

1.2 Scope

The airline booking website is an application stored in the user server. The purpose of the website is to resolve the client to allow website users to perform tasks related to booking an airline flight. The system enables to perform the following functions:

Automation of flight operations
Automation of ticketing / seat booking confirmation system
Improved and optimized service

1.3 Assumption and Dependencies

- System could provide a platform to book tickets .
- System consisting various features including the admin side logins and employee and flight management system

1.4 Overview

The remaining section of this document provide a general description including characteristic of the users of this product, the product's hardware, and functional and non-functional requirements of the product.

2. Overall Description

Developing an AIRLINE FLIGHT CONTROL SYSTEM- AFCS for an air line company that want to automate its flight operations and ticketing / seat booking and confirmation system.

2.1 Product Perspective

- User Module : Used for ticket booking ,check flight status , inquiry reservations
- Administration Module : Scheduling flights, routing flights ,managing airline data

2.4 Product Functions

- User module
 - Inquiry
 - Ticket
 - booking
 - Status of
 - booking
- Administration Module
 - Adding flights
 - Adding employee info
 - Scheduling flights
 - Reserving booked seats

3. Functional Requirements

User Satisfaction: The system is such that it stands up to the user expectations.

Response Time: The response of all operations is good.

Error Handling: Response to user errors and undesired situation has been taken care of to ensure that the system operates without halting.

Safety and Robustness: The system is able to avoid or tackle disastrous action. In other words it should be fool proof.

Portable: The software should not be architecture specific. It should be easily transferable to other platforms if needed.

User Friendliness: The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

4. Design Constrain

There are a number of factors in the client's environment that may restrict the choices of a designer. Such factors include standards that must be followed, resource limits, operating environment, reliability and security requirements and policies that may have an impact on the design of the system.

Standard Compliances: This specifies the requirement for standards the system must follow. The standards may include the report format and accounting properties.

Hardware Limitations: Hardware limitations can include the types of machine to be used, operating

system available on the system, languages support and limits on primary and secondary storage.

Reliability and Fault Tolerance: Fault tolerance requirement can be place a constraint on how the system is to be designed. Recovery requirements are often on integral part here, detailing what the system should do if some failure occurs to ensure certain properties

Security: Security requirements are particularly significant in defense system and database system. They place restrictions on the use of certain commands, control access to data, provide different kinds of access requirements for different people, require the use of passwords and cryptography techniques and maintain a log of activities in the system

5 .Hardware Requirements

For the hardware requirements like memory restrictions, cache size, the processor, RAM size etc... those are required for the software to run.

5.1 MINIMUM Hardware Requirements:

Processor Pentium IV

Hard Disk Drive 100

GBRAM 1 GB

PREFERRED Hardware Requirements:

Processor Core i3 Hard Disk Drive 500

GBRAM 4 GB

5.2 Software Requirements:

Any window based operating system with DOS support are primary requirements for software development. Windows 7 and up are required. The system must be connected via LAN and

connection to internet is mandatory.

5.3 Other Requirement:

- Security
- Portability
- Correctness
- Efficiency

6.Non-Functional Requirements

- **Security:** The system is must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated management.
- **Reliability:** The reliability of the overall project depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Also the system will be functional under a container. Thus the overall stability of the system depends on the stability of the container and its underlying OS.
- **Availability:** The system should be available at all the times, meaning the user can access it using a web browser, only restricted by the down time of the server on which system runs. A customer friendly system which is in access of people around the worlds should work 24 hours. In case of a hardware failure or database corruption, a replacement page will be shown. Also in case of hardware failure or database corruption backups of the database should be retrieved from the server and saved by the Organizer. Then the service will be restarted. It means 24x7 availability.
- **Maintainability:** In case of a failure, a re-initialization of the system will be done. Also the software design is being done with modularity in mind so that maintainability can be done efficiently.
- **Supportability:** The code and supporting modules of the system will be well documented and easy to understand. Online user documentation and Help system requirements will be provided.

6.1 Performance Requirements

- Performance requirements define acceptable response times for system functionality.
- The load time for user interface screens shall take no longer than two seconds.
- The log in information shall be verified within five seconds.

- Queries shall return results within five seconds.

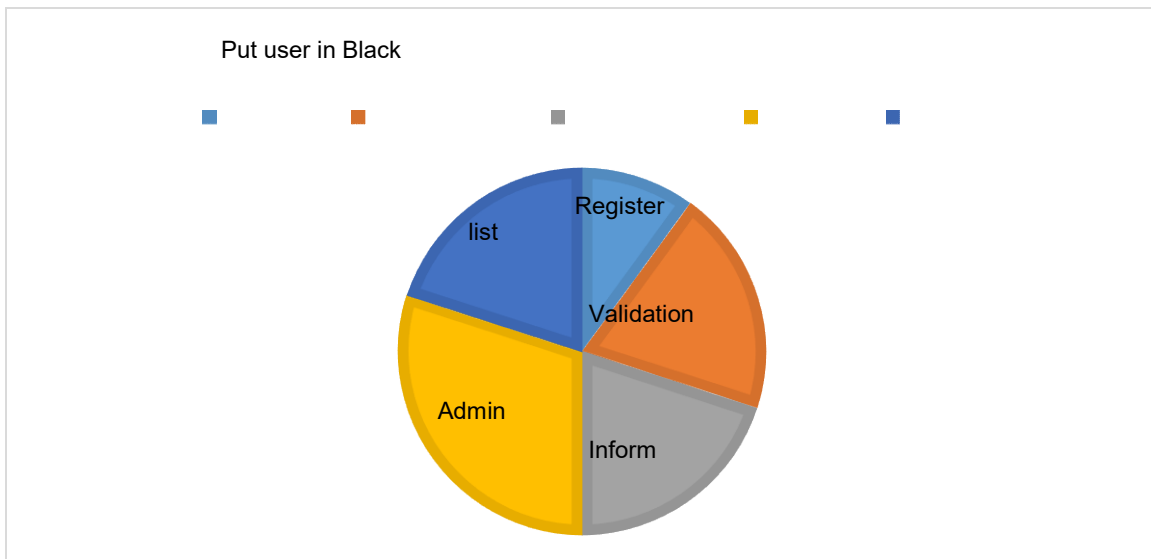
6.2 Logical database requirements

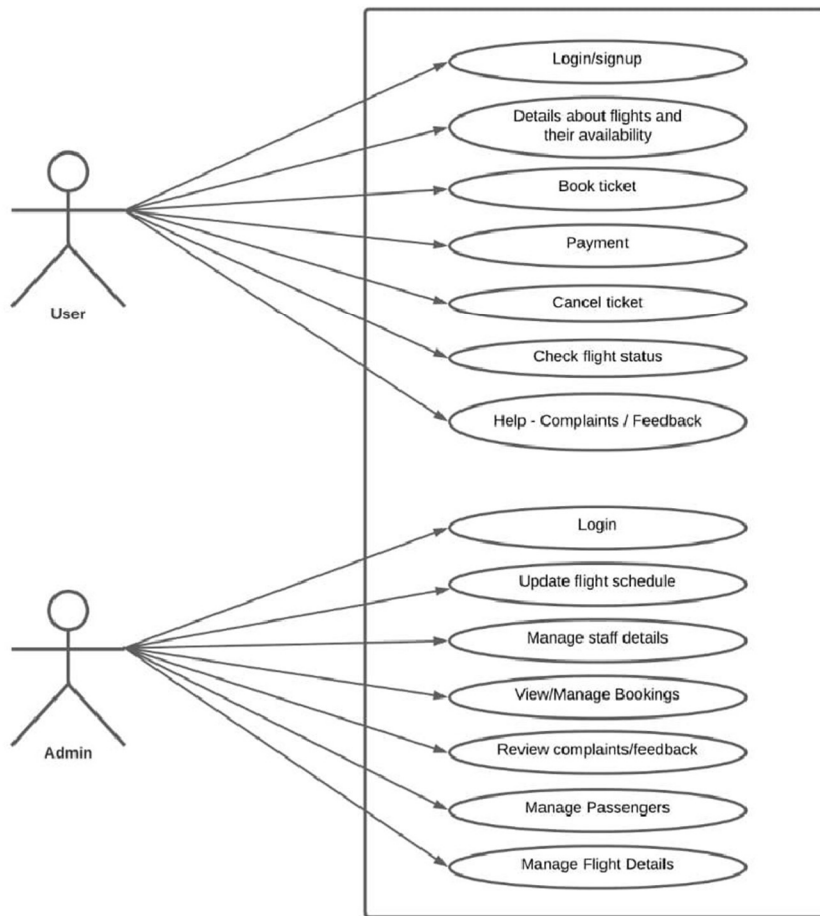
The following information will be placed in the database:

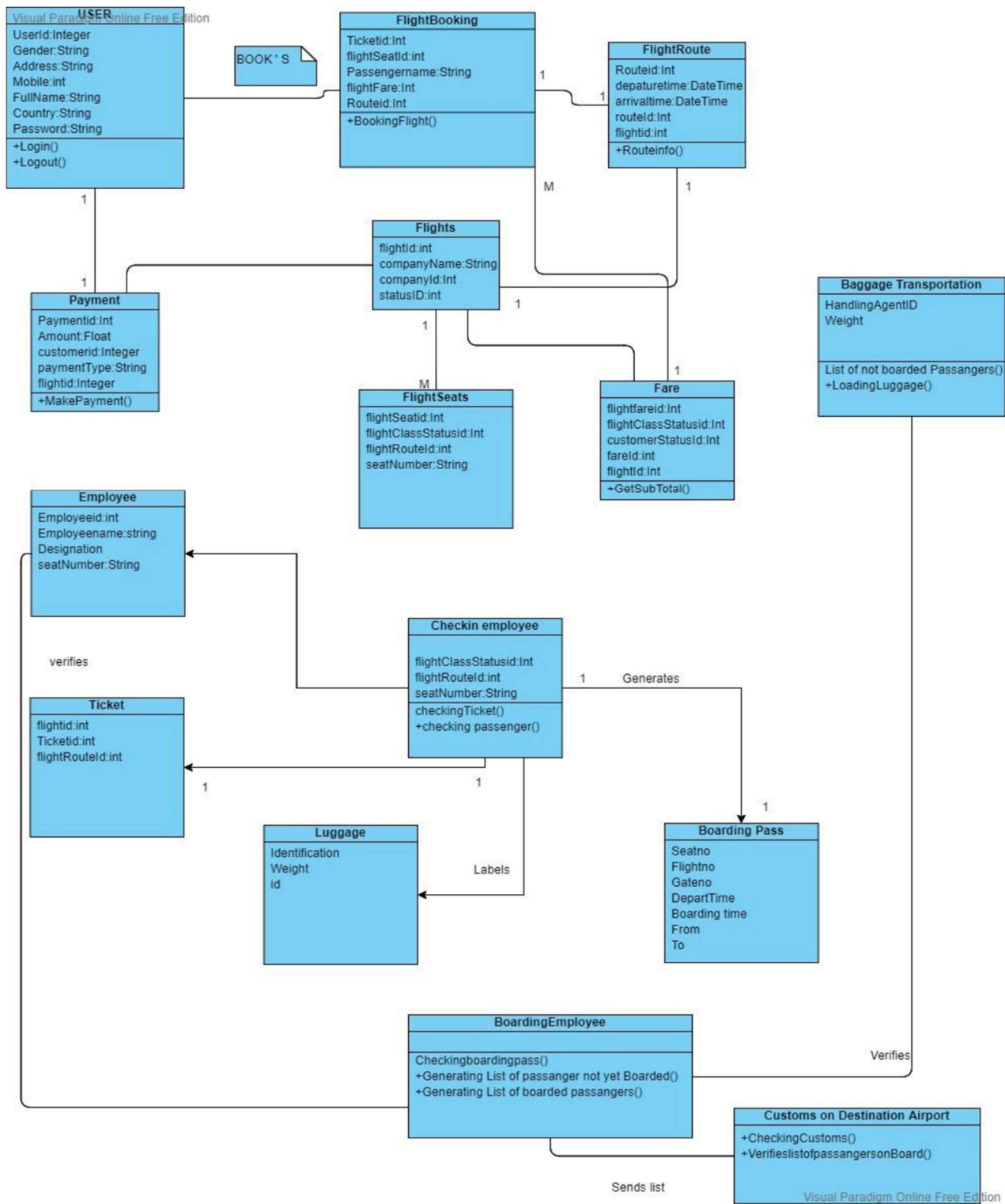
User Details: ID, Login Name, Email, Password and Name.

Preliminary schedule

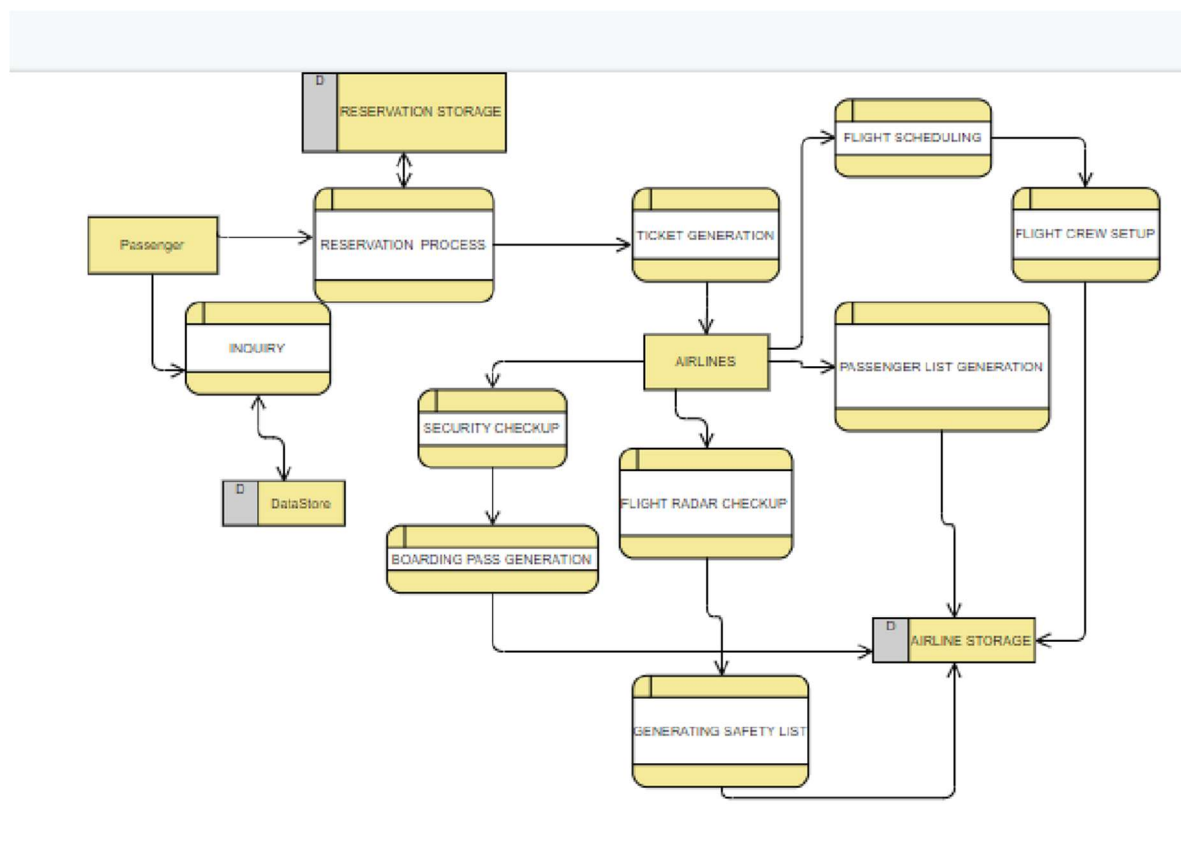
- We gave 1 month to discuss about software, cost, deadlines, risk analysis etc.
- 2 month for requirement gathering from user and then by analysing that we made SRS document and verified that from user.
- 2 months for Data collection like details of different companies which wanted to involve.
- 3 months to Design software according to user requirements and
- 2 months for implementation and testing for final product.



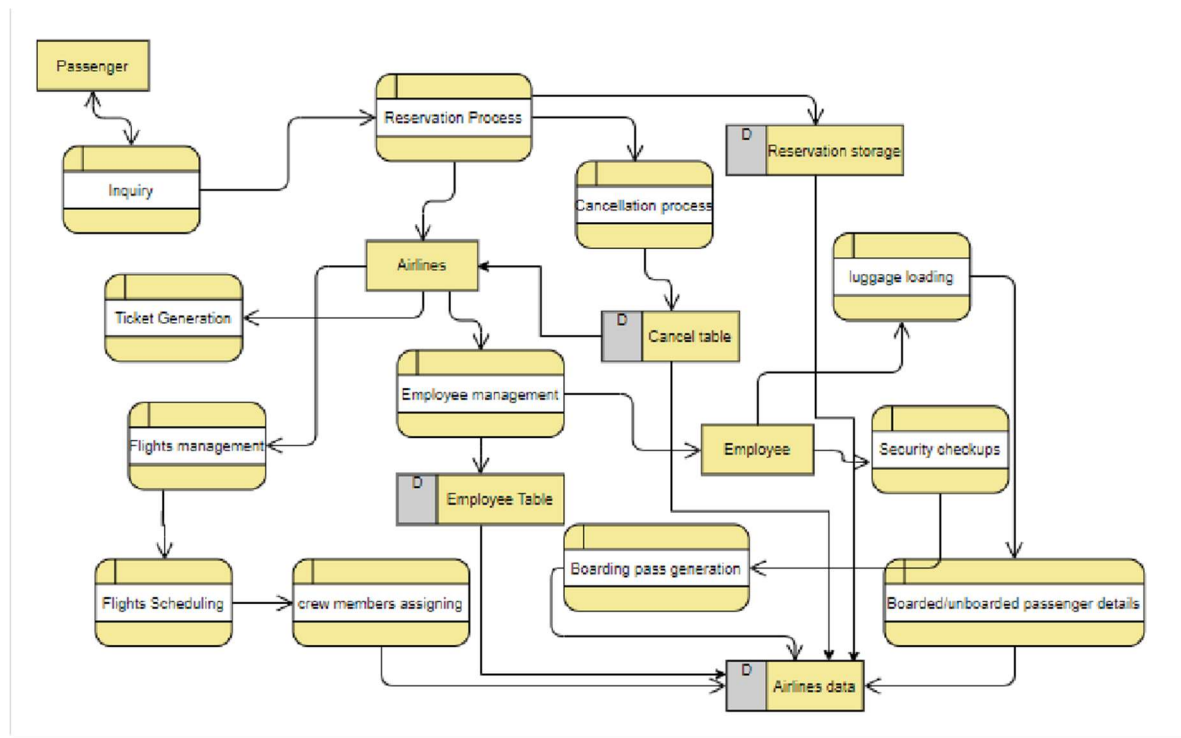
USE CASE DAIGRAM:

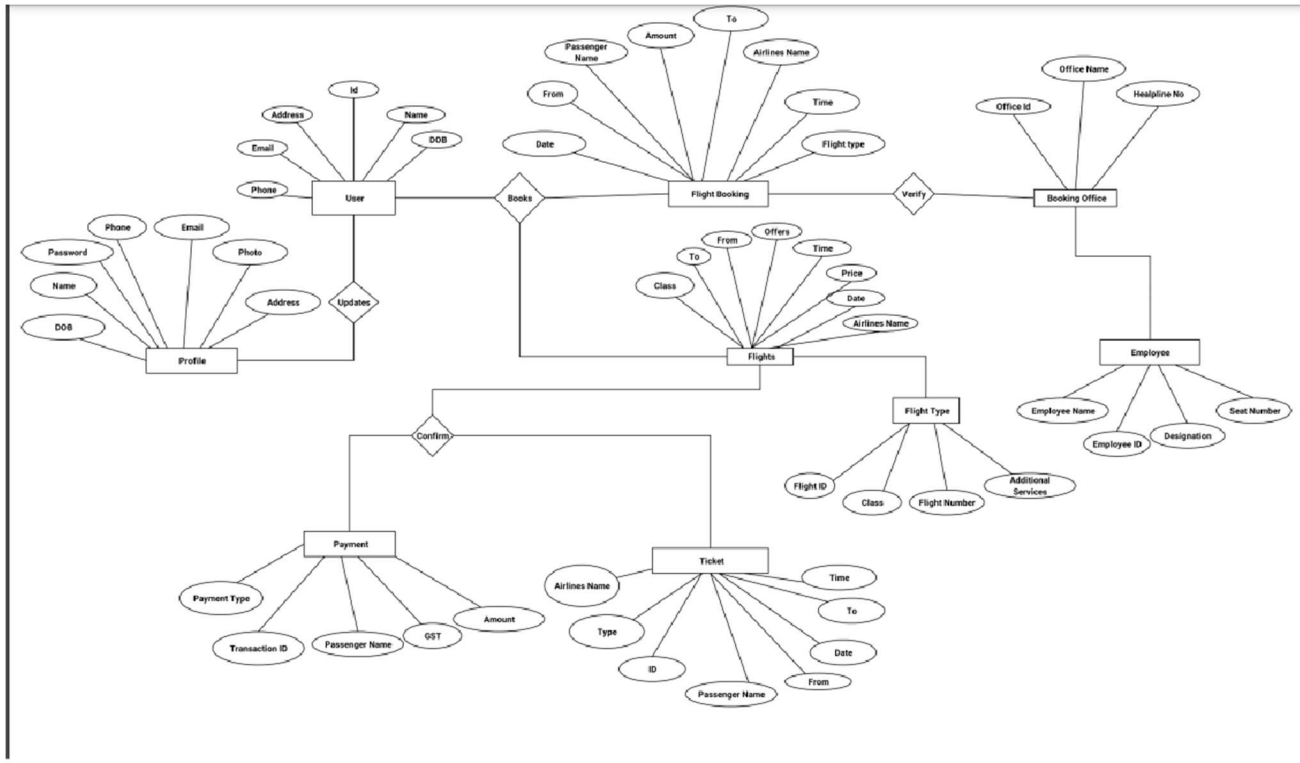
CLASS DAIGRAM:

Context level DFD:

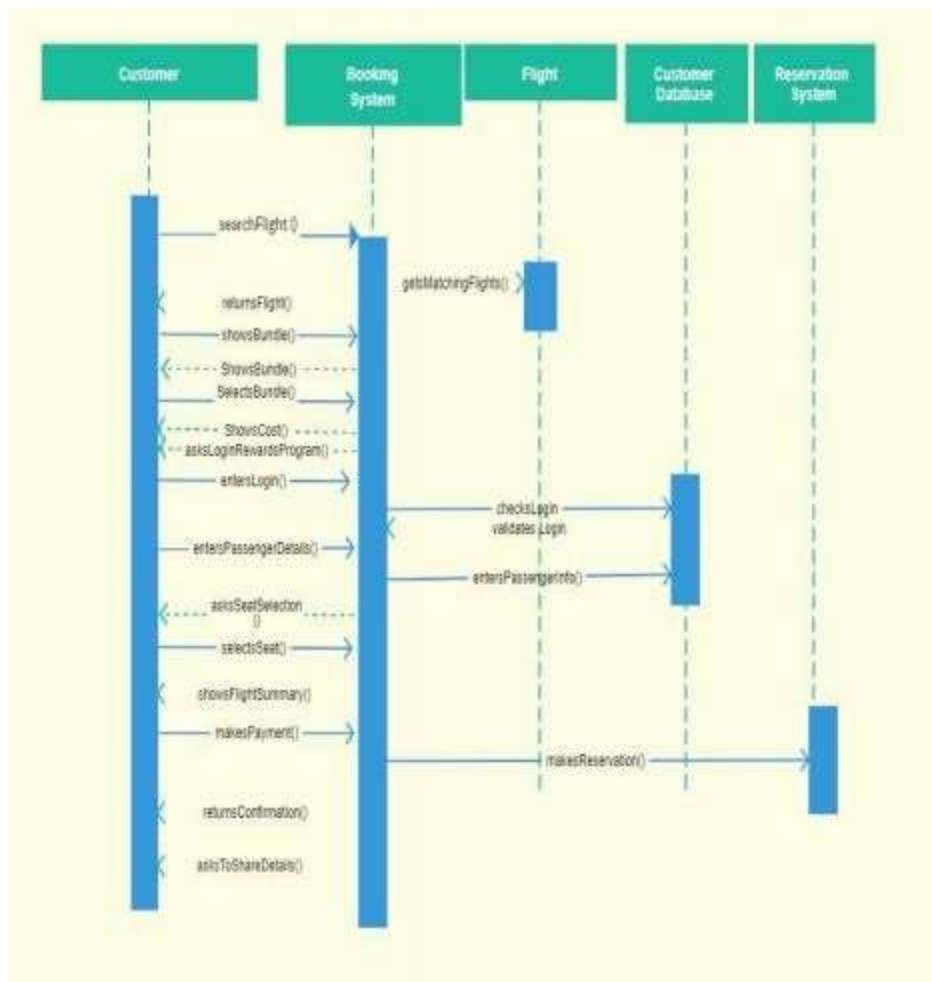


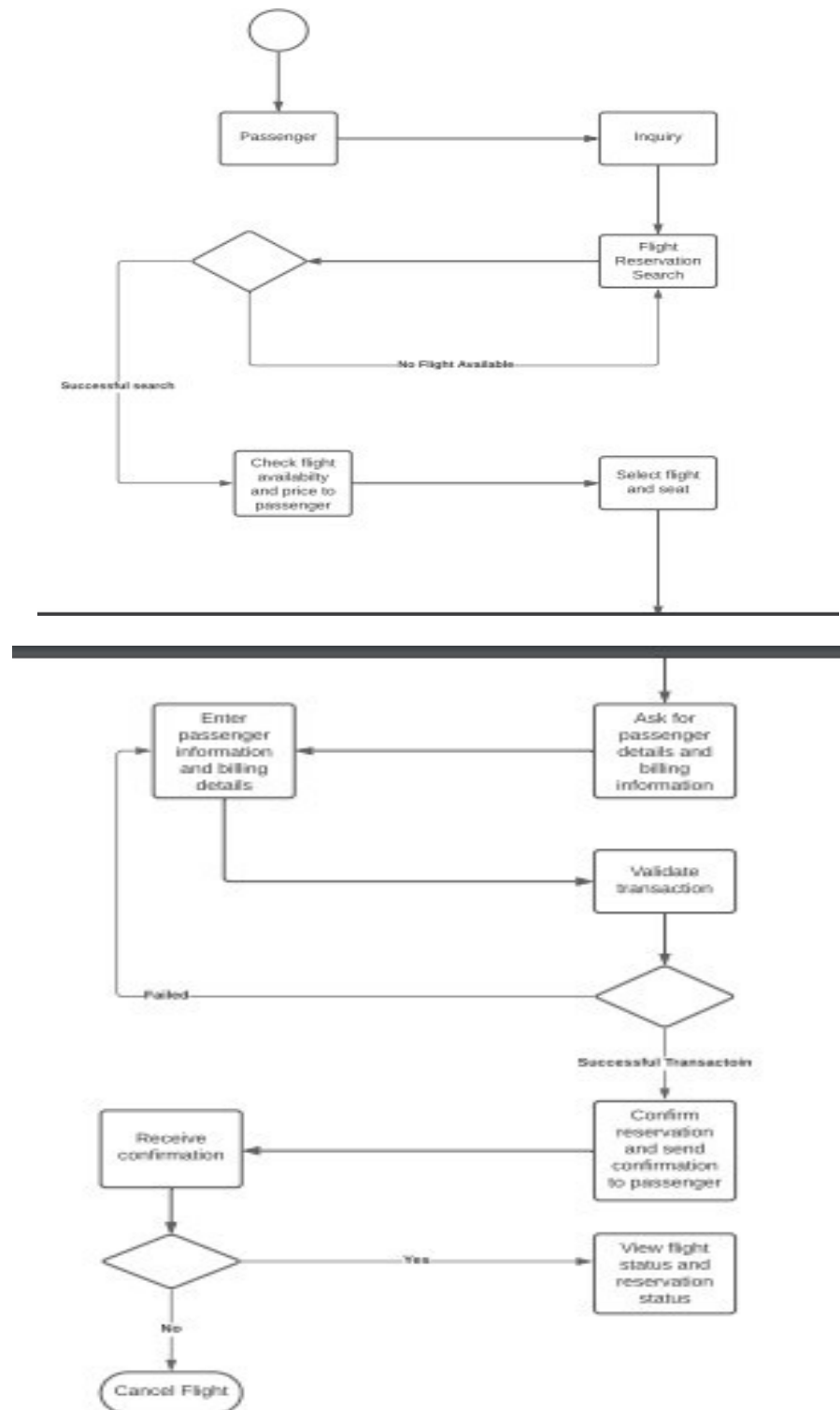
LEVEL-2

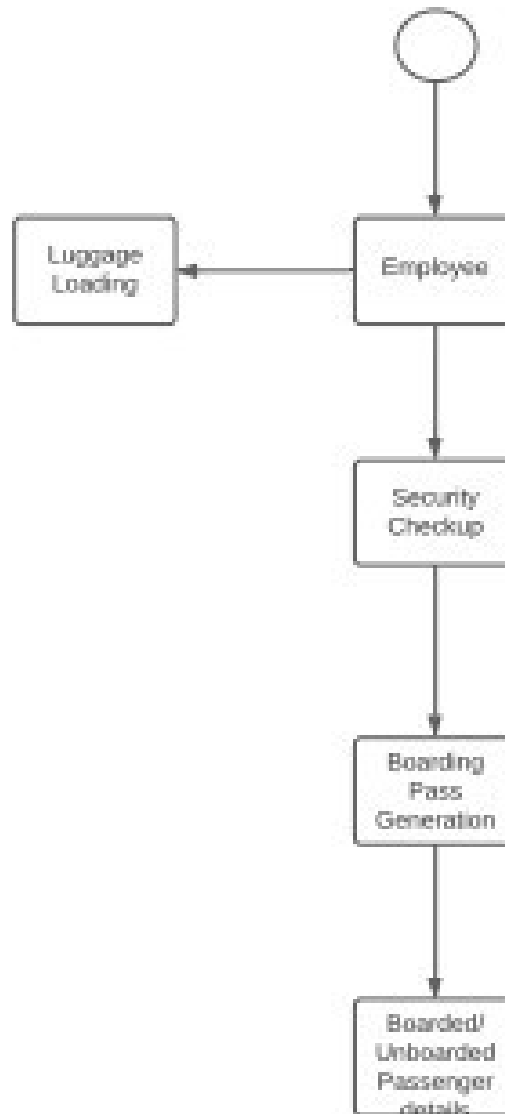


ER DAIGRAM:

SEQUENCE DAIGRAM



ACTIVITY DAIGRAM:

ADMIN:**TESTCASE:**

210130175 Test case id	21 Test case name	Test case description	SE Test Steps Bhavya R			Patelt status (P/F)
			Step	Expected result	Actual result	
AccTest 01	Seat Availability	Verify whether the ticket is available or not.	Enter: <ul style="list-style-type: none">Flight number: 0340Date of journey: 11 Feb 2007Source code: MUMDestination code: DELClass: Economy	Should display accommodation available enquiry		
			Enter: <ul style="list-style-type: none">Flight number: a@1%Date of journey: 11 Feb 2007Source code: MUMDestination code: DELClass: Economy	Display the message: Invalid Flight number		
	Get fare details	Verify the working of get fare	Click on get fare option	System displays a table of fare stating base fare, reservation change in class		
AccTest 02	Reservation	Verify reservation procedure	If seats are available click on get availability	System display reservation form		
			Enter passenger details and flight details	System accept the valid details and display the confirmation form.		
GUI01	Verify aesthetic condition	Is the general screen background the correct colour?	Open the browser and type URL of the site	Background colour is proper making foreground information visible.		
		Is the screen resize and minimisable ?	Open the site, resize and minimize the same.	System gets resized and minimized.		

GUI02	Verify validation condition	Does a failure of validation of every field cause a sensible user error message	Open fare enquiry for a flight and feed following details: Flight no. asd	System displays error message: Invalid flight number! Try again		
Usability accessibility 01	Verify Usability	Does the home page load quickly?	Open the browser and type URL of the site	System displays a home page quickly		
	Interactivity	To verify whether interaction mechanisms (pull-down menus) are easy to use	Open fare enquiry for a flight and check different categories of train-class	Class pull-down menu displays various categories of class		
Usability accessibility 02	Navigation	Does each hyperlink work on each page	Open the home page and try the navigations for: <ul style="list-style-type: none"> Flight schedule Flights Fare Seat availability 	System redirects to respective pages.		

Ambulance dispatch system

Content

- 11. Introduction
- 12. Purpose
- 13. Scope
- 14. Software requirement specification
 - 14.1 User characteristic
 - 14.2 Functional requirement
 - 14.3 Non-functional requirement
- 15. Interface requirements
- 16. Software interface
- 17. Hardware interface
- 18. Design Constraint
- 19. Test-Case
- 20. Diagrams

1. Introduction

The Ambulance Dispatch System (ADS) is a web-based tool to allow the administration of emergency response system. It maintains locations of ambulances that can be dynamically configured at administration time. The system maintains a history of response results for analysis.

2. Purpose

The purpose of the ADS is to enhance the capabilities of 911 ambulance dispatchers in the timely displacement of ambulance(s) to injury scenes in an efficient manner.

3. Scope of the System

The scope of the ADS starts at the moment the dispatcher gathers information from the caller and ends at the patient(s) arrival to a hospital.

4. Software requirement specification

4.1 Overview

Calling 911 and asking for the ambulance service would connect the caller to a dispatcher who feeds the information s/he receives from the caller into the system. The system would allocate & mobilize a suitable ambulance within 3 minutes, transmit details to the selected vehicle, and track and monitor actual performance and position. An exception message shall be generated if no free ambulance is available for at least 11 minutes. The system would show the location of each patient, the nearest three ambulances, and the nearest available hospitals.

4.2 Functional Requirements

The Ambulance Dispatch System supports the following users: the dispatcher and the ambulance driver. The dispatcher tasks consist of logging into the system, emergency data entry, monitoring the progress of the system, and logging out of the system. The ambulance driver is the key person to respond to the status queries. This setup leads to the following functional requirements:

4.2.1 Logging In

Req. 1 The dispatcher shall log into the system by entering his/her dispatcher identification number and password.

Req. 2 The dispatcher identification number shall be a 5 digit decimal number.

Req. 3 The dispatcher password, an 8 character long alphanumeric string, shall be defined by the dispatcher.

Req. 4 After logging in, the dispatcher shall be taken to the dispatch home screen.

4.2.2 Dispatcher Settings

Req. 5 The dispatcher shall be able to change his/her password and other settings related to his/her account.

4.2.3 Data Entry

Req. 6 After answering a call, the dispatcher shall gather and enter the information into the system.

4.2.4 Data Entry Corrections/Suggestions

Req. 7 The system shall correct formatting errors and offer suggestions (i.e. allow address and emergency abbreviations) to the dispatcher during the data entry process.

4.2.5 Caller Address Location

Req. 8 As soon as the call is received, the system shall automatically start the address location procedures based on the caller's phone number.

Req. 9 If successful, the dispatcher shall verify the address with the caller.

Req. 10 If not, the dispatcher shall ask the caller about his/her location.

4.2.6 Duplicate Calls Detection

Req. 11 The system shall detect potential duplicate calls (calls from 2 or more people describing the same emergency) by performing a quick comparison of the locations and emergency descriptions of all incoming calls and notifying responsible dispatchers if a similar emergency is already in the system.

4.2.7 Ambulance Location

Req. 12 The system shall locate the 3 available ambulances that are closest to the emergency location and present them to the dispatcher in a graphical format, i.e. by displaying a map and marking locations of the emergency and the ambulances.

Req. 13 After the dispatcher chooses one of them the system should transmit the emergency information to the ambulance's mobile receiving unit and start the status monitoring process.

4.2.8 No Available Ambulances

Req. 14 In case the system cannot find any available ambulances in the area, the system shall query the status of the ambulances currently allocated to other emergencies, select 3 that are soon-to-be-available, and present them to the dispatcher. He/she should make the final decision.

4.2.9 Exception Message

Req. 15 An exception message shall be generated for the dispatcher if no ambulance is allocated within 11 minutes of the dispatcher's data entry.

Req. 16 This exception message shall read: "ERROR: NO AMBULANCES HAVE BEEN ALLOCATED".

Req. 17 Upon receiving this message, the dispatcher shall be required to manually select an ambulance by entering the ambulance's identification number.

Req. 18 Ambulance identification numbers shall be 3 hexadecimal digits.

4.2.10 Communication with the Ambulance

Req. 19 The system shall have an interface to communicate with the ambulance driver.

Req. 20 The system shall allow sending the emergency information to the ambulance as well as querying the ambulance about the status of the emergency.

4.2.11 Hospital Availability

Req. 21 As soon as the dispatcher allocates an ambulance to the emergency, the system should present him/her with 3 hospitals closest to the emergency location.

Req. 22 The dispatcher shall make his/her choice, after which the system will transmit this information to the ambulance in charge.

4.2.12 Monitoring Performance and Position

Req. 23 The system shall track the ambulance's performance and position.

Req. 24 The ambulance's performance shall be based on the time it takes to arrive at the scene once allocated and the time to get the patient to the hospital.

Req. 25 The ambulance's position shall be displayed on the map for the dispatcher to monitor.

4.2.13 Monitor Display

Req. 26 The dispatcher's monitor shall display the following data after s/he has completed the data entry:

- The location of the emergency
- The location of the ambulance(s) in route to the emergency.
- The location of the nearest three ambulances to the emergency location.

4.2.14 Monitoring Complete

Req. 27 The dispatcher shall close out the monitoring of an emergency once the allocated ambulance(s) has/have arrived at a hospital.

Req. 28 The dispatcher shall click "Emergency Resolved" to close out the monitoring phase of the dispatch system.

Req. 29 The dispatcher shall be returned to the dispatch home screen when clicking "EmergencyResolved".

Req. 30 The dispatcher can also go to an open emergency request to see its status.

4.2.15 Information Logging

Req. 31 The system shall log all calls and the related emergency information for future review and statistical purposes.

4.2.16 Emergency Transfer/Sharing

Req. 32 The dispatcher shall be able to transfer the emergency to another dispatcher in case s/he has to log out of the system.

Req. 33 The dispatcher shall also be able to share the emergency information with other dispatchers in case he/she needs help.

4.2.17 Logging Out

Req. 34 The dispatcher shall log out of the system by clicking "Log Out".

Req. 35 The dispatcher shall not be allowed to log out while currently monitoring an ongoing emergency unless s/he transferred or shared the emergency with at least one other dispatcher

4.3 *Non Functional Requirements*

4.3.1 *Usability*

- Simple to Operate: The software should be easy to learn and operate; the user should not require special skills or training to operate the system.
- Simple design: The user interface should be kept as simple as possible so as not to make the application too confusing for the user to understand i.e., user friendly interface.
- User awareness: User manual and in-build help file will be provided for the user. Tool tip text will also be provided for quick help.

4.3.2 *Reliability*

- The system should be up and running 24 X 7 X 365 and should be crash safe during 95% of its runtime.
- Mean time between failures (MTBF): The MTBF (if any) should not be less than 6 months.
- Mean time to repair (MTTR): In case of a failure that leads to a system outage, the MTTR should not be more than 2 hours.

4.3.3 Performance

- Short response time: Any page of the application should not take more than 4 seconds to load. The load time of the application should not be more than 4 seconds.
- Population Support: The application should be able to support 250 concurrent users without any performance degradation.

4.3.4 Supportability

- Advanced technologies: As technology is changing so fast, the system should be able to support new technologies for tracking which will be faster and reliable than the ones present now.
- GPS: The system should be able to support GPS tracking in the future.
- Address location using phone coordinates: The system should support locating the address, using phone coordinates of the person making the call.

4.3.5 Implementation

- Programming language: Java and allied technologies should be used for development of the application.
- Apache's Tomcat Web-Server should be used to deploy the application.
- MySQL should be used as the database. Business Objects will be used for reports.

4.3.6 Packaging

- The software will also be available online, and anybody

authorized by the system administrator can access the system.

4.3.7 Legal

- Data from the user should adhere to the rights of data privacy of the user. All the content must be procured through legal channels and there should be no copyright violations.

4.3.8 Security

- As the system will be dealing with delicate data, the system should be secure. The data should be stored in a highly secure manner and should be immune from any hacking attempts.
- The system should be able to scale up to 500 concurrent users (if there is a need in the future) by installing additional hardware components with no degradation in the performance of the system.

4.3.9 Schedule Constraints

- The entire system should be up and running in the user's production environment by 24th July.

4.3.10 Standards Constraints

- All the documents delivered should adhere to the IEEE standards for software engineering.

5. Interface requirements

Various Interface for the product could be:

- Login Screen
- Active Emergencies Screen
- New Emergencies Screen
- Emergency Details Screen

6. Software Interface:

- The system requires a properly configured version of Windows 2000 or windowsXP to run the application.
- These computers must have Microsoft .NET Framework 1.1 or greater installed. The system's server can use either Windows, Linux or UNIX, but it must have MySQL properly installed and configured.

7. Hardware Interface:

- The application can run on any hardware which supports Windows 2000 or windows XP.
- The system must have at least 100 megabytes of free disk space to install the program and 512MB-1GB of memory is required to load the application.
- The Program does not write information directly to the user's computer but instead uses a database which is located on network server. The user's computer transfers and receives data from the server using basic networking

protocols. Allsystem's information is stored in the server's database which stores the data on the server's disk.

8. Design Constraints:

- The customer requested a system where an operator can input information about a 911-incident call. The system must prioritize calls and allocate and mobilize a suitable ambulance to comply with federal mandates.
- These mandates state that an ambulance must be dispatched within an average of 3 minutes of the 911-incident call.
- The ambulance must arrive at the scene of the incident within an average of 11 minutes from its dispatch. At the time of 911 call, once the input information has been gathered the customer requests that the system be able to locate the three nearest ambulance to the incident.
- Dispatchers must be able to update the status of each ambulance. Each status change, along with each of 911 call's information, must be stored to allow reports to be generated.
- These reports can verify the company's compliance with current federal mandates.
- The customer also wishes to replace operators as much as possible in an effort to reduce costs.

9. Test-case:

Test Case # - 001

Test Case Name - User Login

Test Items – Functionality of the User Login screen

Specifications –

Valid UserName -

asb@acb6354.com Valid

Password – abc

Invalid UserName –

sam Invalid Password –

bob

Output Specifications –

Input	Expected Output
Valid UserName + Valid Password	Display Dispatch Screen
Invalid UserName + Valid Password	Display Error Message

Valid UserName + Invalid Password	Display Error Message
Invalid UserName + Invalid Password	Display Error Message

Environmental Needs –

Working CAD System.

Special Procedural Requirements – None

Dependencies – None

Test Case # - 002

Test Case Name – Enter Incident Info

Test Items –

Functionality of the Dispatcher screen

Input Specifications –

Name - Bob

Valid Phone # - 9729007833

Invalid Phone # - 9725885698

Patient condition – In car accident.

Output Specifications –

Input	Expected Output
Valid Phone #	Address field is populated
Invalid Phone #	Throws exception

Environmental Needs –

Working CAD System.

Special Procedural Requirements –

None

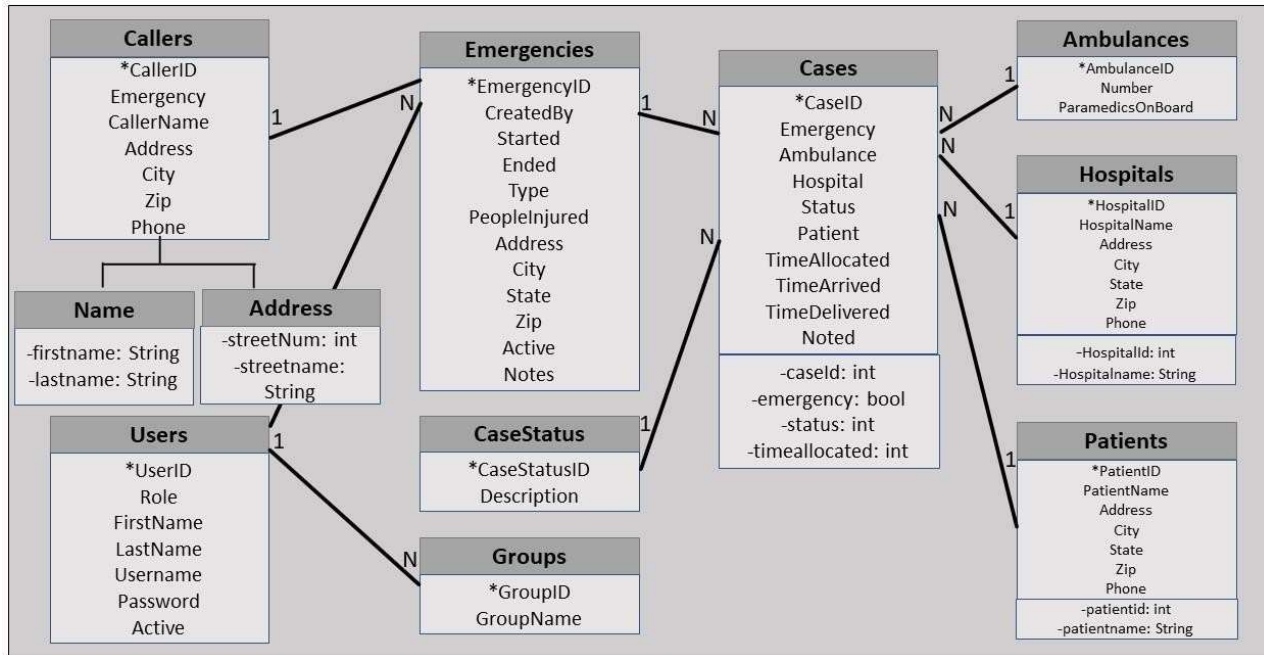
Intercase Dependencies –

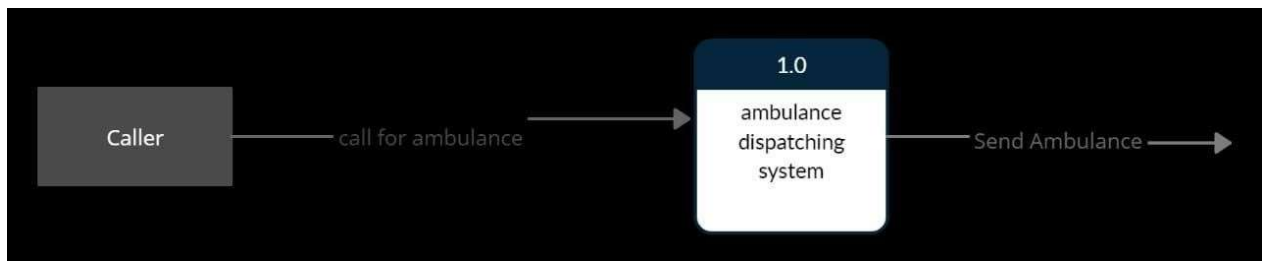
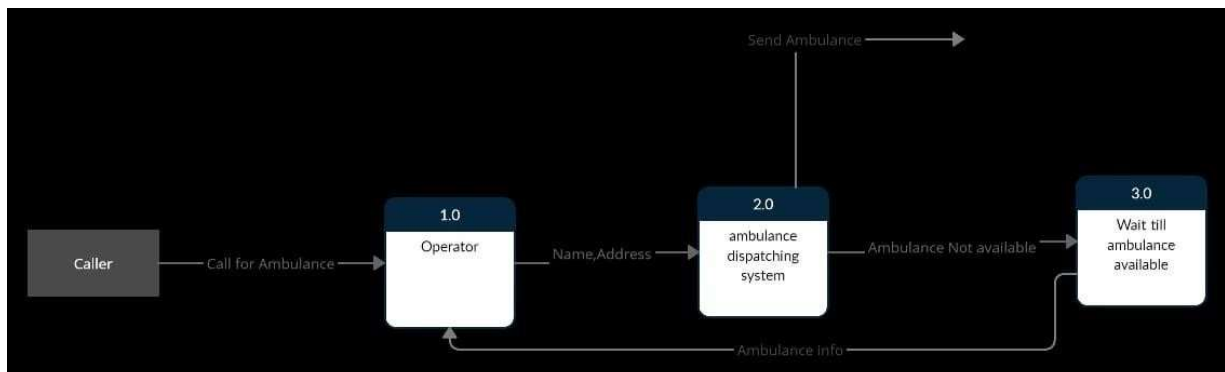
None

10. Diagrams:

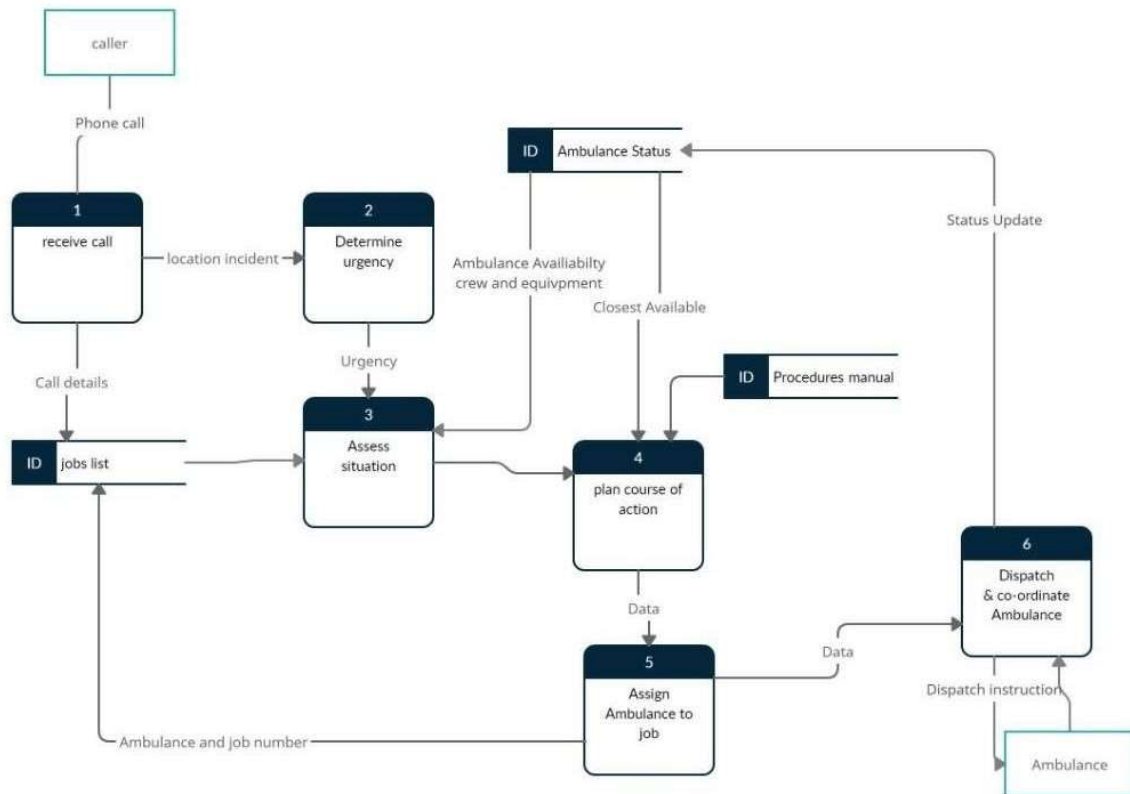
10.1 Class Diagram:

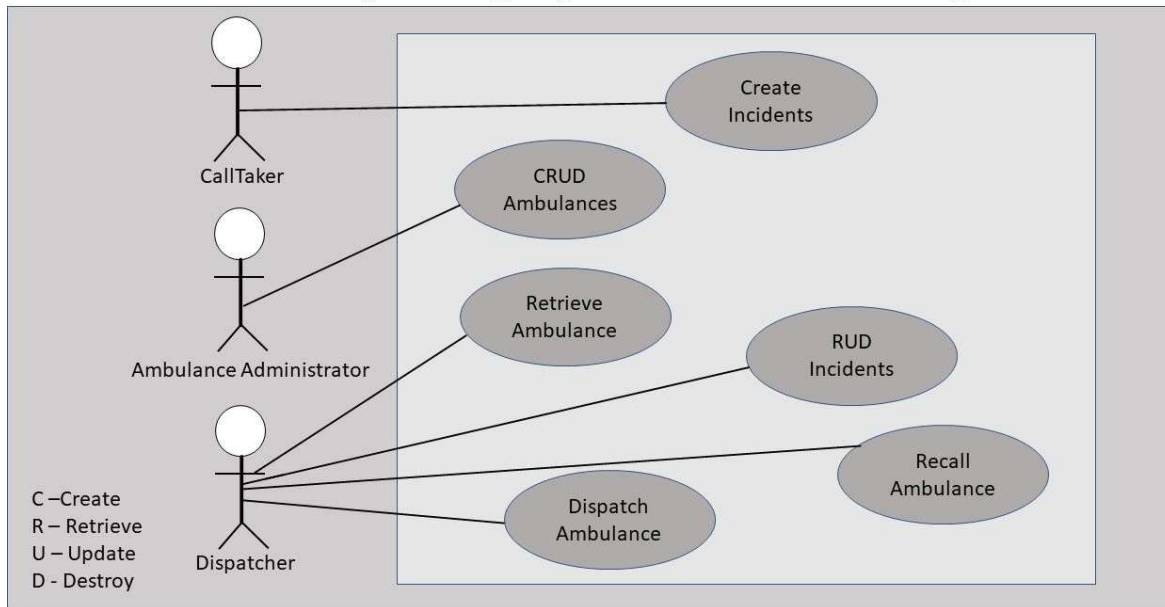
Ambulance Dispatching System : Class Diagram



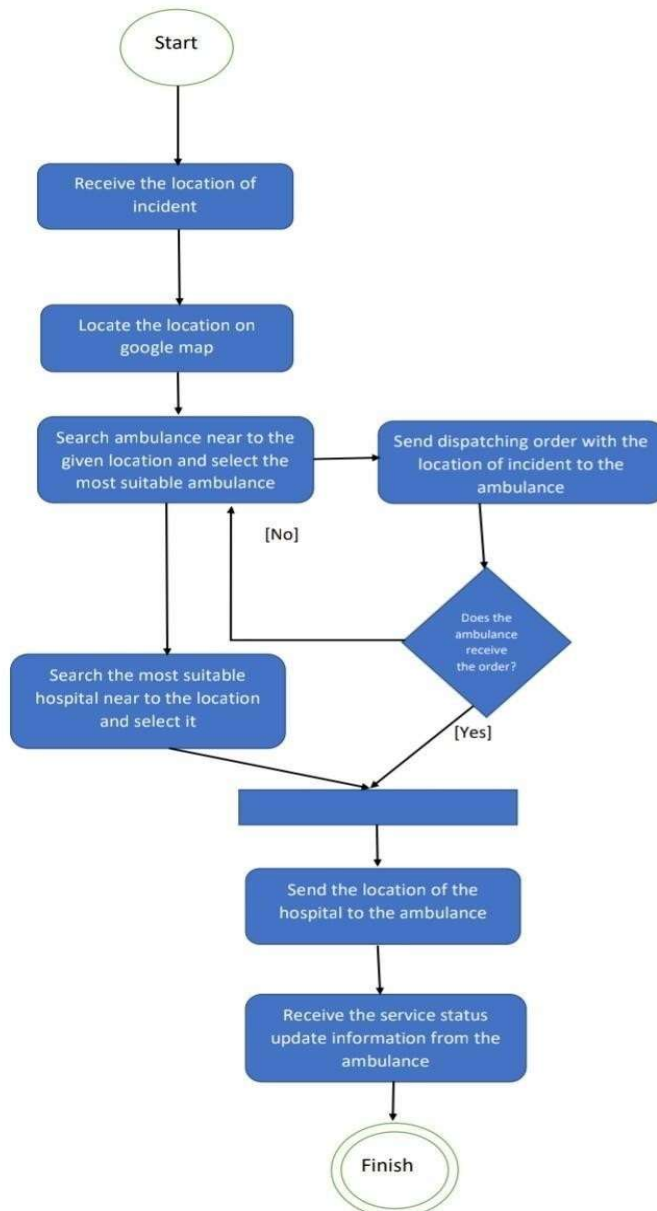
10.2 Data Flow Diagram:*Level 0 Diagram:***Level 1 Diagram:**

Level 2 Diagram:

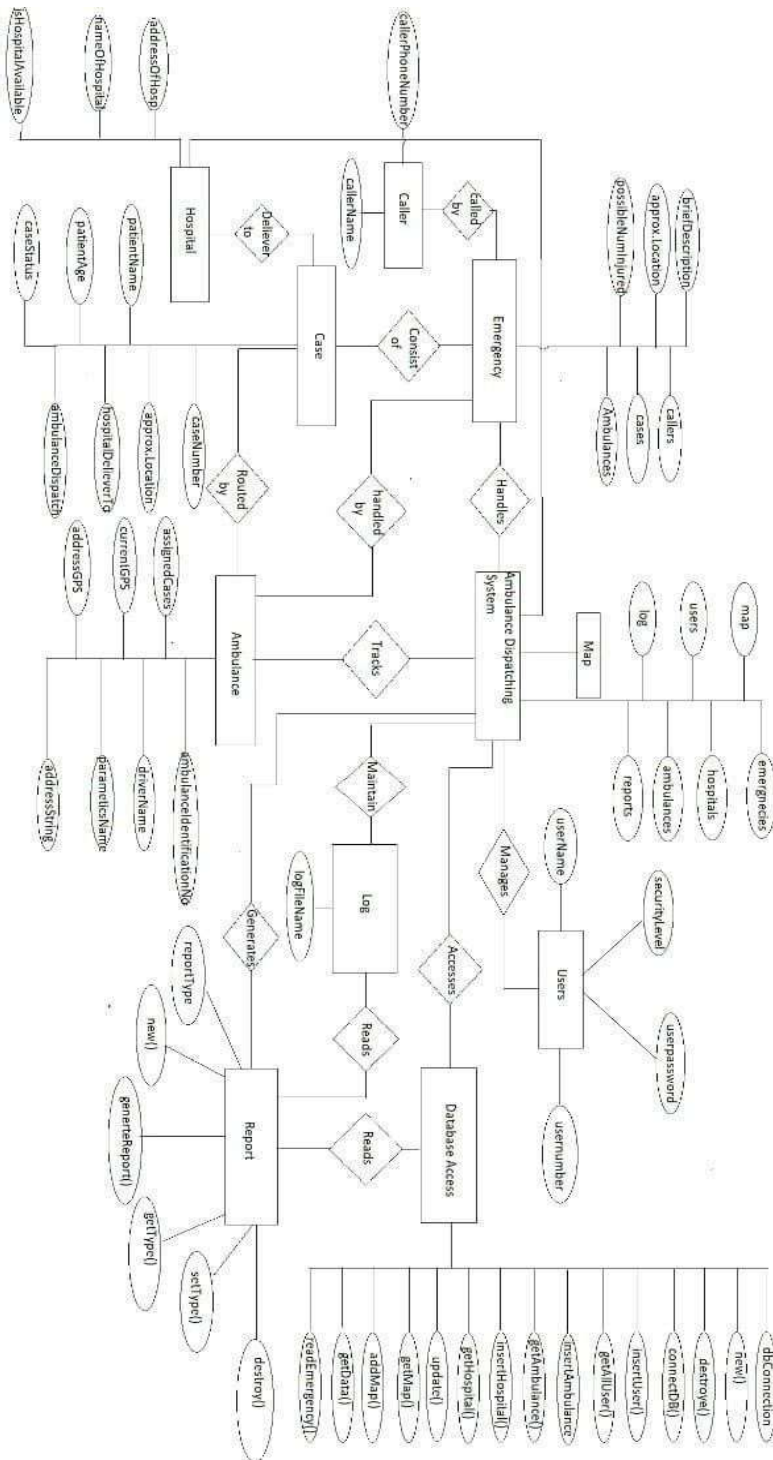


10.3 Use-Case Diagram:**Ambulance Dispatching System : Use-Case Diagram**

10.4 Activity Diagram:

Ambulance Dispatching system : Activity Diagram

ER Diagram:

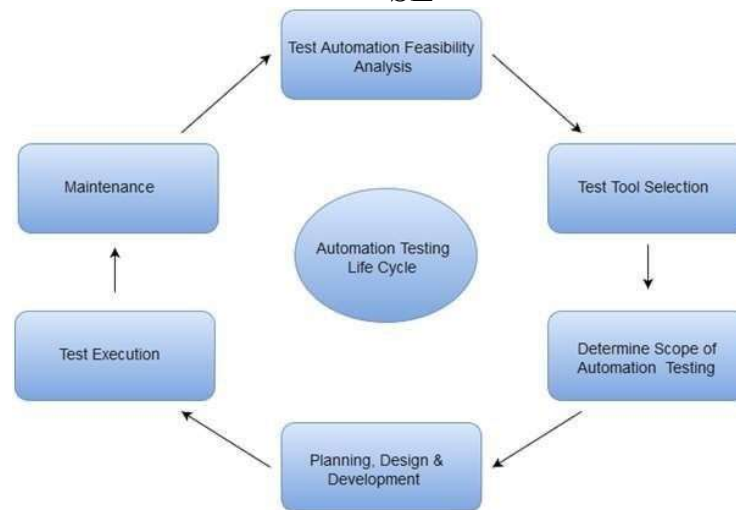


TUTORIAL : 07

Study of any any two Open source tools in DevOps for Infrastructure Automation, Configuration Management ,Deployment Automation, Performance Management, Log Management. Monitoring. (Behat , Watir, Chef, Supergiant, SaltStack, Docker, Hudson etc).

Selenium:

- Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.
- Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.
- Selenium supports a variety of programming languages through the use of drivers specific to each language. Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby. Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.



AUTOMATION TESTING

Automation testing uses the specialized tools to automate the execution of manually designed test cases without any human intervention. Automation testing tools can access the test data, controls the execution of tests and compares the actual result against the expected result. Consequently, generating detailed test reports of the system under test.

Automation testing covers both functional and performance test on an application.

- Functional automation is used for automation of functional test cases. For example, regression tests, which are repetitive in nature, are automated.
- Performance automation is used for automation of non-functional performance test cases. For example, measuring the response time of the application under considerable (say 100 users) load.

SELENIUM FEATURES:

- Selenium is an open source and portable Web testing Framework.
- Selenium IDE provides a playback and record feature for authoring tests without the need to learn a test scripting language.
- It can be considered as the leading cloud-based testing platform which helps testers to record their actions and export them as a reusable script with a simple-to-understand and easy-to-use interface.
- Selenium supports various operating systems, browsers and programming languages. Following is the list:
- Programming Languages: C#, Java, Python, PHP, Ruby, Perl, and JavaScript
- Operating Systems: Android, iOS, Windows, Linux, Mac, Solaris.
- Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
- It also supports parallel test execution which reduces time and increases the efficiency of tests.
- Selenium can be integrated with frameworks like Ant and Maven for source code compilation.
- Selenium can also be integrated with testing frameworks like TestNG for application testing and generating reports.
- Selenium requires fewer resources as compared to other automation test tools.
- WebDriver API has been indulged in selenium which is one of the most important modifications done to selenium.
- Selenium web driver does not require server installation, test scripts interact directly with the browser.

- Selenium commands are categorized in terms of different classes which make it easier to understand and implement.
- Selenium Remote Control (RC) in conjunction with WebDriver API is known as Selenium

GIT:

✓ What is Git?

- Git is an open-source distributed version control system. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers. The version control allows us to track and work together with our team
- Git is foundation of many services like GitHub and GitLab, but we can use Git without using any other Git services. Git can be used privately and publicly.
- Git was created by Linus Torvalds in 2005 to develop Linux Kernel. It is also used as an important distributed version-control tool for the DevOps.
- Git is easy to learn, and has fast performance. It is superior to other SCM tools like Subversion, CVS, Perforce, and ClearCase.

✓ FEATURES OF GIT

Some remarkable features of Git are as follows:

➤ OPEN SOURCE

Git is an open-source tool. It is released under the GPL (General Public License) license.

➤ SCALABLE

Git is scalable, which means when the number of users increases, the Git can easily handle such situations.

➤ **DISTRIBUTED**

One of Git's great features is that it is distributed. Distributed means that instead of switching the project to another machine, we can create a "clone" of the entire repository. Also, instead of just having one central repository that you send changes to, every user has their own repository that contains the entire commit history of the project. We do not need to connect to the remote repository; the change is just stored on our local repository. If necessary, we can push these changes to a remote repository.

➤ **SECURITY**

Git is secure. It uses the SHA1 (Secure Hash Function) to name and identify objects within its repository. Files and commits are checked and retrieved by its checksum at the time of checkout. It stores its history in such a way that the ID of particular commits depends upon the complete

development history leading up to that commit. Once it is published, one cannot make changes to its old version.

➤ **SPEED**

Git is very fast, so it can complete all the tasks in a while. Most of the git operations are done on the local repository, so it provides a huge speed. Also, a centralized version control system continually communicates with a server somewhere.

Performance tests conducted by Mozilla showed that it was extremely fast compared to other VCSs. Fetching version history from a locally stored repository is much faster than fetching it from the remote server. The core part of Git is written in C, which ignores runtime overheads associated with other high-level languages.

Git was developed to work on the Linux kernel; therefore, it is capable enough to handle large repositories effectively. From the beginning, speed and performance have been Git's primary goals.

✓ **BENEFITS OF GIT**

A version control application allows us to keep track of all the changes that we make in the files of our project. Every time we make changes in files of an existing project, we can push those changes to a repository. Other developers are allowed to pull your changes from the repository and continue to work with the updates that you added to the project files.

SAVES TIME

Git is lightning fast technology. Each command takes only a few seconds to execute so we can save a lot of time as compared to login to a GitHub account and find out its features.

➤ **OFFLINE WORKING**

One of the most important benefits of Git is that it supports offline working. If we are facing internet connectivity issues, it will not affect our work. In Git, we can do almost everything locally. Comparatively, other CVS like SVN is limited and prefer the connection with the central repository.

➤ UNDO MISTAKES

One additional benefit of Git is we can Undo mistakes. Sometimes the undo can be a savior option for us. Git provides the undo option for almost everything.

➤ TRACK THE CHANGES

Git facilitates with some exciting features such as Diff, Log, and Status, which allows us to track changes so we can check the status, compare our files or branch

