CHAPTER 5

DESIGN, DEVELOPMENT AND IMPLEMENTATION OF SDSS

The purpose of this chapter is to identify and discuss the main component of SDSS, brief details about the software and programming enviornment so used for development of SDSS, together with designing of User Interface and step by step process for development of SDSS. This chapter is organized in five sections. The first section gives the design overview by covering details about tools & techniques overview, product perspective, technology forecast, user classes and operating environment. The second and third section includes the brief details about hardware and software Architecture so used, together with specification about geospatial requirements, adobe flex and its coding standard. The fourth section covers the various levels of flow diagrams i.e. context level, level 1,2 and 3. The final section includes the detailed description about various system features together with their functionality and working.

5.1 DESIGN OVERVIEW

5.1.1 Tools and techniques overview:

The coming of web 2.0 technologies has opened up the never foreseen vista of Internet in its most potent form. The data that was earlier hidden in files obscured by numerous formats has come to the fore in a multitude of representation forms. With the coming of web 2.0, the user participation has become the prime force driving the internet and the presence of internet has left just the realm of personal computers and has pervaded all kinds of devices, wired as well as unwired. The users are now able to generate the content that they wish to view as against the static pages they were forced to see in the past. The client -server interaction is ever on the rise with each coming application.

Another thing that has changed is the way data is presented to the users. Earlier the user had limited choice and had to be content with the static pages that were on offer. The new technologies have brought to fore a torrent of technologies each having their own strong points. Some of these can be enumerated as: JavaScript, Ajax, Flex, Silverlight etc. Of
these technologies JavaScript and Ajax have been relatively older and form the core of many applications developed by Google and Yahoo. But the trouble with these technologies is that they involve a high amount of coding with a rigid syntax that requires a lot of training.

Adobe Flex is a software development kit released by Adobe Systems for the development and deployment of cross-platform rich Internet applications based on the Adobe Flash platform. Flex applications can be written using Adobe Flex Builder or by using the freely available Flex compiler from Adobe. The flex builder of Adobe supports various server side programming environment like ASP.Net, Cold Fusion, J2EE, PHP for cross platform software development. The deployment of rich GIS application is based on industry standard Adobe flex player for best performance and visualization.

Flex has brought a new life to data presentation on internet. The quality of graphics produced on flex is exceptional. Additionally, content can be generated for each by working on easy to use Software Development Kits (SDK): Flex Builder for Flex. In the case of Flex, a large number of sample applications are also available along with a very strong community and support which is very crucial for any software development cycle. These sample applications also include web applications showcasing the power of GIS of the type being aimed at in the current project. Additionally, to display all the power of Flex, the user just requires a small flash plug-in at the client end that is usually present in most cases.

Browser and OS compatibility is another issue where Flex has an upper hand. Flex just requires a flash plug-in to be installed at the client’s browser which is easily available for all browsers and OS viz. Windows, Linux etc. Silverlight - developed by Microsoft, on the other hand is not compatible with all browsers and OS.

Additionally, flex gives the developer the option of data representation in a vast number of ways. There are a large number of charts, graphs and tables that can easily be created out of the existing data. Each of these ways is very rich looking and gives the feel of working on a system that is alive.
5.1.2 **Product features:** Any Decision support system aiming to have geographic content along with other collateral data must be a comprehensive one and should have the following features at the least:

- **Spatial data viewer:** The spatial data viewer can load data from centralized.

- **Map display tools:** To display the map and to enable the user to view the area of his choice using tools such as Levels of Display, Zoom In, Zoom Out, Zoom to scale, Pan, Go to, full extent, Overview Map, Choice of displayed layers, identification, theme overlays etc.

- **Query and Search:** To help the user locate the information across various layers by having Fixed Queries as well as advanced option to build his own queries, Search using keywords on multiple layers etc.

- **Data presentation:** This will primarily use the display of data through appropriate kinds of Charts, Graphs and data in tabular form.

- **Analysis:** To help the user arrive at decisions by combining data from multiple layers using Union, Intersection and perform various tools over them like proximity analysis, buffering, clipping etc.

- **Annotation and output:** These features will enable the user to customize the data and to take the output forward. Such a tool set can include symbologies to mark a point, highlight an area, create pdf maps, print etc.

5.1.3 **Technology forecast**
Attempts to create various kinds of information system in electrical domain have been made throughout the world over the past years. While the old system constituted of looking up data from physical files, the coming of cheap processing power and Internet to distribute it has ushered in a new era for creating a digital information system. While these serve as the basic guiding force behind the proposed information system, the scope of the current work encompasses much beyond them. The best use of the latest technologies must
be made so as to present the data in simplest of ways, yet not losing any of its potency. At the same time, an eye must be kept on the technologies that are appearing on the horizon and attempt be made to learn and implement them at all times so that the existing system may not appear out of date when it is presented in the public domain.

5.1.4 User Classes and Characteristics

The basis for categorizing the users into classes must be based upon the amount of data access that may be granted to them. Based on this, the class of users that has been arrived at is:

- **DISCOM Users**: - These are the premium / main users which will be able to get full access to the web application at all scales. All the facilities envisaged in SDSS will be accessible by these users. These users are basically the DISCOM officials.
- **General users**: - the web application can be designed in such a way that only consumer details module is made available to the general public together with the mapping module. These users are basically the general consumers of electricity.

5.1.5 Operating Environment:

Since spatial decision support system for infrastructure management is proposed to be a tool to view and distribute data over Internet, it is very desirable that it should be totally independent of the type of system it is being viewed upon. This may even be the crux of the utility of any such product in the coming days. As such, development of the system should be in such a way, that the user must be able to access it from machines with its operating systems like Windows. Similarly, it must be accessible across all web -browsers like Internet Explorer, Firefox, Netscape Navigator, Google Chrome etc.

Since the proposed system is being built using Adobe Flex, the only pre-requisite for the systems in this respect is that the browsers must have Flash players installed on them. But this is hardly a constraint as a large amount of information on the Internet is available in this form and almost all users have flash players already installed on their systems.
5.2 HARDWARE ARCHITECTURE

The SDSS web application developed is based on basic client server architecture. The web application is being developed using Adobe Flex and IIS as the default web server. The whole application and supporting file will be residing at root drive of the IIS server. The actual application is based on the service created using Arc GIS Server software. ArcGIS Server is a distributed system consisting of several components. The components of ArcGIS Server can be summarized as:

GIS server—Hosts and runs services. The GIS server consists of a server object manager (SOM) and one or more server object containers (SOCs).

Web server—Hosts Web applications and Web services that use objects running in the GIS server. Clients—Web browsers can be used to connect to Web applications running in the Web server. Desktop applications can connect either through Hyper Text Transfer Protocol (HTTP) to ArcGIS Web services running in the Web server.

5.3 SOFTWARE ARCHITECTURE

5.3.1 Operating System

*System requirements:*-

<table>
<thead>
<tr>
<th>Processor</th>
<th>Pentium 1GHz minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Microsoft Windows 2000 Professional with Service Pack 3 or higher</td>
</tr>
<tr>
<td></td>
<td>Microsoft Windows Server 2003</td>
</tr>
<tr>
<td></td>
<td>Microsoft XP Home Edition with Service Pack 1 or higher</td>
</tr>
<tr>
<td></td>
<td>Microsoft XP Professional Edition with Service Pack 1 or higher</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>Internet Explorer 6.0 or higher</td>
</tr>
<tr>
<td>Windows Installer 3.1</td>
<td>RAM 512 MB minimum</td>
</tr>
<tr>
<td>Installation Drive</td>
<td>1433 MB NTFS disk space</td>
</tr>
<tr>
<td>System Drive</td>
<td>Up to 50 MB</td>
</tr>
</tbody>
</table>

*Table 5.1: System Requirements*
Software Interface Used:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Windows 7 ultimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front End</td>
<td>Flex Builder 3</td>
</tr>
<tr>
<td></td>
<td>Microsoft .NET(C#)</td>
</tr>
<tr>
<td></td>
<td>Visual Studio 2008</td>
</tr>
<tr>
<td>Back end GIS</td>
<td>ArcGIS server 10</td>
</tr>
<tr>
<td>Web server</td>
<td>IIS (Internet Information Services)</td>
</tr>
<tr>
<td>Other Software’s used</td>
<td>ArcGIS Desktop 9.2</td>
</tr>
<tr>
<td></td>
<td>Google Earth pro</td>
</tr>
</tbody>
</table>

Table 5.2: Software interface used

5.3.2 Geospatial Requirements:

The web application so developed will employ GIS technology as a means to display, query, and analyze electrical data. Providing support for this GIS capability will significantly improve user access to all electrical and collateral information data. To this end, the Architecture will primarily employ GIS software components from the ESRI’s ArcGIS suite of geospatial tools. These are described below:

The ArcGIS family of products- ArcGIS Desktop, ArcGIS Engine, and ArcGIS Server - is built from Arc Objects. Casual users of ArcGIS Desktop are probably not aware of, nor do they care about, Arc Objects. They simply run Arc Map and edit their map. It’s not until they want to start customizing ArcGIS Desktop or building their own applications that they become more familiar with Arc Objects. The main difference between running a desktop application and one based on a GIS server is in where the Arc Objects components reside. For desktop applications, the objects needed for the application are instantiated and run in the application itself on the local machine. For instance, when one starts Arc Map and opens a map document, the Arc Map application instantiates the objects that allows drawing of the map, adding layers, editing the layout, and so on. Essentially, all the objects stored in the map document are rehydrated for use during that Arc Map session. When one saves the map, the current state of the objects used during the session is written to the map document for later use. Unlike desktop applications, clients of a GIS server -based application access the Arc Objects components remotely. One can think of the GIS serve r
as the container that hosts the objects for all client applications that need to use them in much the same way an Arc Map session hosts the objects for an individual user. However, with the GIS server, objects are shared between clients. Because of this, the way clients use the GIS resources running on the server is a little different. When one creates an application that accesses a GIS server, the prescribed coding guidelines for accessing the objects running on the server needs to be followed. These guidelines are presented and described in the Developer Help system.

5.3.2.1 ArcGIS Server:

Familiarity with ArcGIS Desktop also leads to familiarity with the GIS resources one works within the desktop environment. For example, if one wants to display GIS data, one does so through maps and globes. If locations need to be found by their address, an address locator can be used. If some analysis needs to be done, the geoprocessing tools in the toolbox can be utilized. Each of these items encapsulates some level of GIS functionality. Fundamentally, the GIS functionality is exposed through a specific set of Arc Objects components that implement the functionality.

On the GIS server, instead of working with, for instance, map documents, globe documents, and address locators, one works with map services, globe services, and geocode services. GIS resources that have been made available on the server are collectively referred to as services. Actually, one still uses map documents, globe documents, and address locators, as these are the sources for the services hosted on the GIS server. Thus, if one wants to share a map that has been made on the GIS server, the map document to define the map service that runs on the server is used.

The main purpose of a GIS server is to host services and distribute them to client applications that need to use them. Additionally, the GIS server provides a set of tools that allows managing the services; for example, one can use the ArcGIS Server Manager application to add and remove services. It's useful to understand how a GIS server system is put together so that one can build applications that effectively utilize Arc Objects running in a server environment. This section serves as an introduction to the components
that compose a GIS server. The following image shows the ArcGIS Server system architecture.

![The ArcGIS Server System Architecture](image)

**Figure: 5.1 Components of ArcGIS Server System**
An ArcGIS Server system is made up of some of the following components:

- **GIS server**— The GIS server hosts the GIS resources, such as maps, globes, and address locators, and exposes them as services to client applications. The GIS server itself is composed of two distinct parts: the server object manager (SOM) and server object containers (SOCs). As the name implies, the SOM manages the services running on the server. When a client application requests the use of a particular service, it's the SOM that actually provides one for the client to use. The SOM connects to one or more SOCs. The SOC machines host the services that the SOM manages. Depending on the configuration, one can run the SOM and SOC on different machines and also have multiple SOC machines. The Figure 5.1 above shows a SOM machine connected to two SOC machines.

- **Web server**— The Web server hosts Web applications and services that use the resources running on the GIS server.

- **Clients**— Clients are Web, mobile, and desktop applications that connect to ArcGIS Server Internet services or ArcGIS Server local services.

- **Data server**— The data server contains the GIS resources that have been published as services on the GIS server. These resources can be map documents, address locators, globe documents, geodatabases, and toolboxes.

Manager and Arc Catalog administrators - ArcGIS Server administrators can use either Manager or Arc Catalog to publish their GIS resources as services. Manager is a Web application that supports publishing services, administering the GIS server, creating Web applications, and publishing ArcGIS Explorer maps on the server. Arc Catalog includes a GIS Servers node, which can be used to add connections to GIS servers for either general server usage or administration of a server's properties and services.

ArcGIS Desktop content authors - To author the GIS resources, such as maps, geoprocessing tools, and globes that will be published to the server, one needs to use ArcGIS Desktop applications such as Arc Map, Arc Catalog, and Arc Globe. Additionally, if a cached map service is being created, Arc Catalog will be needed to create the cache. Working for an organization that manages geographic information, one faces the challenge of sharing one’s collection of geographic information with people inside the
organization and those outside as well. ArcGIS Server provides the platform for sharing the GIS resources, such as maps, with the user community, whether they're sitting in the same office using ArcGIS Desktop or sitting across the country accessing and viewing maps through the Internet.

ArcGIS Server allows the sharing of GIS resources across an enterprise and across the Web. GIS resources are the maps, globes, address locators, geodatabases, and tools that one wants to share with others. These resources are shared by hosting them on the ArcGIS Server system or GIS server, and allowing client applications to use and interact with the resources. The main advantages of sharing the GIS resources on a GIS server are the same as sharing any data through any kind of server technology: the data is centrally managed, supports multiple users, and provides clients with the most up-to-date information. One might ask as to why a GIS server is needed for this, wouldn't any server technology work? In addition to providing access to particular GIS resources, the GIS server also provides access to the GIS functionality that the resource contains. For example, one might be able to share a map with someone through a server, but it would be even better if that person could also interact with the map, like find the closest hospital, restaurant, or bank and get directions to it from their location. Thus, the GIS server not only allows one to share resources, like maps, but also to access the GIS functionality embedded in them.

5.3.2.2 ArcGIS Desktop:
ArcGIS Desktop includes a suite of integrated applications that allow you to perform GIS tasks, from simple to advanced, including mapping, geographic analysis, data editing and compilation, data management, visualization, and geoprocessing. It is available at different product levels, with increasing functionality.

- **ArcReader** - is a basic data viewer for maps and GIS data published in the proprietary ESRI format using ArcGIS Publisher.
- **Arc View** - is the entry level of ArcGIS licensing offered. With Arc View, one is able to view and edit GIS data held in flat file ArcEditor - is a powerful GIS desktop system for editing and managing geographic data. It includes all the functionality of Arc View along with additional advanced editing tools to ensure the quality of your data.
• **Arc Catalog** - Arc Catalog is the data management application, used to browse datasets and files on one's computer, database, or other sources. In addition to showing what data is available, Arc Catalog also allows users to preview the data on a map.

![Arc Map](image)

**Figure 5.2 Arc Map**

• **Arc Map** - Arc Map is the central application used in ArcGIS, where you display and explore the datasets for your study area, assign symbols, and create map layouts for printing or publication. Arc Map is also the application used to create and edit datasets. It represents geographic information as a collection of layers and other elements in a map.

• **Arc Toolbox** – It contains geoprocessing, data conversion, and analysis tools, along with much of the functionality in Arc Info.

• **ArcScene** - ArcScene is a map display similar to Arc Map that is customized for displaying 3-D data.

• **Arc Globe** - Arc Globe is a 3D visualization application that allows you to view large amounts of GIS data on a globe surface.
5.3.3 Adobe Flex:
Flex framework belongs to Adobe line of products. It is a highly productive, free open source framework for building expressive web applications that deploy consistently on all major browsers and on the desktop with Adobe AIR. It is a programming framework with an eclipse based IDE. It creates an interface between the server -side data services and the client. Getting started with Flex begins with working in Flex Builder 3 integrated development environment (IDE). Adobe ® Flex® Builder™ software is a highly productive Eclipse™ based development tool enabling intelligent coding, interactive step through debugging, and visual design of the user interface layout, appearance, and behavior of rich Internet applications (RIAs). For the development of SDSS, flex builder 4 is used. To make it work with ArcGIS we use ArcGIS API for Flex provided by ESRI.
Using the Flex API, developers can combine GIS -based Web services from ArcGIS Server with other Web content, which can be displayed in simple, dynamic mapping applications over the Web or on the desktop. The API exploits the powerful geospatial capabilities of ArcGIS services. Users can transform their local data into a visually rich
interactive map, query and display GIS data features and attributes, locate addresses, identify features, and perform complex spatial analytics. The two languages used to write Flex applications are: MXML and Action Script.

5.3.3.1 About MXML:
MXML is an XML markup language that can be used to lay out user interface components. It can also be used to declaratively define non-visual aspects of an application, such as access to data sources on the server and data bindings between user interface components and data sources on the server. Like HTML, MXML provides tags that define user interfaces. MXML will seem very familiar if one has worked with HTML. However, MXML is more structured than HTML, and it provides a much richer tag set. For example, MXML includes tags for visual components such as data grids, trees, tab navigators, accordions, and menus, as well as non-visual components that provide web service connections, data binding, and animation effects.

One can also extend MXML with custom components that are referenced as MXML tags. One of the biggest differences between MXML and HTML is that MXML -defined applications are compiled into SWF files and rendered by Adobe® Flash® Player or Adobe® AIR™, which provides a richer and more dynamic user interface than page-based HTML applications. One can write an MXML application in a single file or in multiple files. MXML also supports custom components written in MXML and Action Script files.

5.3.3.2 About Action script:
Flex developers can use Action Script to implement custom behavior within their Flex applications. Firstly MXML tags are used to declare things like the containers, controls, effects, formatters, validators, and web services that the application requires, and to lay out its user interface. Each of these components provides the standard behavior one would expect. For example, a button automatically highlights when the mouse pointer is rolled over it, without writing any Action Script. But a declarative language like MXML is not appropriate for coding the action to happen when the user clicks a button. For that, a procedural language like Action Script is used, which offers executable methods, various types of storage variables, and flow control such as conditionals and loops. In a general
sense, MXML implements the static aspects of any application, and Action Script implements its dynamic aspects. Action Script is an object-oriented procedural programming language, based on the ECMA Script (ECMA-262) edition 4-draft language specification. One can use a variety of methods to mix Action Script and MXML.

5.3.4 Coding Standards:
Since the SDSS system is being developing in the Flex Builder 4, so the coding standards for writing open-source Flex framework components in Action Script is covered in detail in Annexure 1. Adhering to these standards makes the source code look consistent, well organized, and professional. This is accomplished by making the code more readable by use of consistency.

5.4 DATA FLOW DIAGRAM

The general working of the Application can be explained using the data flow diagrams. The features and the functionalities of the application are explained in detail in the subsequent sections. Data Flow Diagrams present the logical flow of information through a system in graphical or pictorial form. The Data Flow Diagram may be used to represent system or software at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional details. Therefore the DFD provides the mechanism for functional modeling as well as information flow modeling.

Figure 5.4: Context level DFD
Figure 5.4 shows the context level data flow diagram that renders the working of the system at extremely superficial level. It depicts foremost entities involved in the system. It explains that the system is accessible to two types of users: general users and the DISCOM users and their interactions with the system.

Figure 5.5 level 1 DFD which shows the interaction of the users with the system in more detail. Both the types of users interact with the system by visiting the website. The general users can view all the GIS functionalities together with customer details tool. DISCOM users, in addition to having access to this information can have access to detailed information provided from the geo database, by using all the modules and tools.

![Figure 5.5: Level 1 DFD](image)

Figure 5.6 shows the level 2 data flow diagram where the DISCOM user is directed to his respective modules. It portrays all the facilities provided to the DISCOM users through the web application. Here the user can put forward requests for mapping actions, query, network analysis, analysis tool, complain management, viewing Chart and printing
Reports. The mapping actions include zooming, panning, previous & next extend and full extend. After the requests are put forward the database is queried by the respective processes and the requested dataset is retrieved which is then displayed on the map in form of results.

**Figure 5.6**: Level 2 DFD
Figure 5.7: Level 3 DFD
Figure 5.7 is the level 3 data flow diagram which gives the insight of how the processes actually interact with the ArcGIS Server to retrieve the required dataset from the geodatabase. The Map area, type of action to be performed and they are sent to the web server which is further handed over to the ArcObject Proxies. These proxies formulate the request and submit it to the SOM (Server Object Manager), which further submits it to the SOC (Server Object Container). Finally, the request is received by the Arc Objects which work in association with the geodatabase to retrieve the desired dataset and send back to the process through the proper channel so that the user gets the processed map with desired results.

5.5 SYSTEM FEATURES

![Diagram of System Features](image-url)
5.5.1 Map Control:

Map control is the control provided by the ESRI, which is basically used to display the map in the application. The map can be zoomed-in and zoomed-out by the mouse roll over and roll out respectively. It also includes built-in feature of panning to the area of interest. Associated with Map control is a navigation bar and scale bar.

![Figure 5.9: Map Control](image)

Navigation Bar can be used to zoom in and zoom out the map extent on the mouse click. Scale Bar can be used to show the scale of the map in current context.

5.5.2 Navigation Tools:

![Figure 5.10: Navigation tools](image)
Navigation tools are the set of tools that will allow user to fully interact with the map. Navigation tools consist of Zoom-In, Zoom-Out, Panning, Previous Extent, and Next Extent to view the different extent of the map.

5.5.2.1 Zoom In
Zoom-In tool would allow the user to perform a rubber zoom -in on the map to reach to the specific area of interest to view the details.

![Zoom In tool](image)

**Figure 5.11:** Zoom In tool

Description:
Zoom-In is a button component, a click on which will allow the user to draw an extent on the map and it will zoom that particular selected area to a defined scale. Zoom-In button uses ESRI’s Navigation Class which provides the navigation. ZOOM_IN property to perform the Zoom In functionality.

5.5.2.2 Zoom Out
Zoom-Out tool would allow user to perform rubber zoom -out to come out of the detailing of the map.

![Zoom Out tool](image)

**Figure 5.12:** Zoom Out tool

Description:
Zoom-Out is a button component, a click on which will allow the user to draw an extent on the map and it will zoom out that particular selected area to a defined scale. Zoom-Out button uses ESRI’s Navigation Class which provides the Navigation. ZOOM_OUT property to perform the Zoom Out functionality.
5.5.2.3 Pan
Panning is one of other basic tools provided in the tool set, which would allow users to shift map holding it with a hand cursor to reach to his area of interest.

![Figure 5.13: Pan Tool](image)

Description:
Pan is a button component, a click on which will help the user to pan the map and move the map where he wishes. It will allow the user to pan around the map without changing the scale of view. Pan button uses ESRI’s Navigation Class which provides the navigation.PAN property to perform the panning functionality.

5.5.2.4 Previous Extent:
Previous Extent allows user to go to the previous extent when the user changes the map extent.

![Figure 5.14: Previous extend Tool](image)

Description:
Previous Extent is a button component, a click on which will help the user to see the previous extent of the map and it zoom the map to that extent. Previous Extent button uses ESRI’s Navigation Class which provide the nav.zoomToPrevExtent() function to perform the functionality. This tool keeps track of the extent history for the browsing of the map and any previous view visualized by the user can be viewed again by using the Previous Extent tool.

5.5.2.5 Next Extent:
Next Extent allows user to return to the next extent when you have gone to previous extent.
Figure 5.15: Next extend Tool

Description:
Next Extent is a button component, a click on which will help the user to see the next extent of the map and it zoom the map to that extent. Next Extent button uses ESRI’s Navigation Class which provide the nav.zoomToNextExtent() function to perform the functionality. This tool too makes use of the browsing history of the map extent and helps in navigating the views along with the Previous Extent tool.

5.5.2.6 Full Extent:
Full Extent allows user to view the map at the full extent.

Figure 5.16: Full extend Tool

Description:
Full Extent is a button component, a click on which will help the user to see the full extent of the map. Full Extent button uses ESRI’s Navigation Class which provide the nav.zoomToFullExtent() function to perform the functionality.

General Guideline for the navigation tools:
Since the navigation tools are the most basic and frequently used tools in any mapping application, these should be arranged in a toolbar on the map for easy access. Suitable icons rather than text should represent these tools, with tool tips to assist the user at all steps.
Other basic Tools too should be placed with these for user-friendly interface.

Overview Map
The Overview Map shows the location of current view in context with the larger geographical area. The shaded red box in the Overview Map panel represents the zoom
window in the map display panel. Users can zoom in and out, and the red box will resize accordingly. To go quickly to any location on map user can pan the red box of overview map and he will be panned to that extent without changing the scale of view. This panel should be draggable and restorable so as to maximize the map display area.

**Figure 5.17:** Overview button

**Figure 5.18:** Overview component

Description:
An overview map is also a button component, on click of which a small overview map window is popped up which shows the current location of the map in context with the larger geographical area. Overview Map button click event calls the function showOverview() which further calls the MapOverview.mxml . This mxml uses the action script, overview, as, to perform the functions.

### 5.5.3 Identify and Search tool

#### 5.5.3.1 Identify Tool
When users select the Identify tool and clicks on the map, it retrieves information (attributes) of all active layers at that point and displays them in the Results window. Users can view all the basic information pertaining to each of the attribute of all active layers at
the point selected on the map, in the result window. The result widow is a part of the Identify Panel. The identified layers are populated in the combo -datagrid.

**Figure 5.19:** Identify button

![Identify Results Window](image)

**Figure 5.20:** Identify Tool

![User Interaction Diagram](image)

**Figure 5.21:** User interaction with identify tool
As the user clicks some other point, information corresponding to the new point will be shown in the result window clearing the last result. As the result window is closed the feature gets deactivated allowing the user to work with other tools. The result window is kept floating over the application so that no part of the map is masked by its visibility.

Description:
Identify feature facilitates use to view the details of the feature of interest. This tool will first display the entire layer in the map and let the user choose layer/s. After completing the choices user gets the details of selected layers. If user clicked the boundary of two or more features, then also he will be given details of all features. User interface consist a DataGrid, detailed information of chosen layer/s is displayed in a separate Panel, along with a header and then details of layer in sequence (if more than one layers are selected)

5.5.3.2 Find Tool

Figure 5.22: Find button

Figure 5.23: Find Tool
The Find tool allows the user to find in particular layer. The find tool asks for some user inputs like search text and layer on which the search has to be made. Find panel also has a tree-view as its part where the search results will be displayed. All the results will be stacked in this result panel. The user has the facility to highlight and view the result on the map also to assist the user visually.

**Figure 5.24:** User interaction with search panel

Description:
Find tool is a set of components, it consists of combo-box to select the various layer and field, another component is a text-box which is editable. The user can either enter the text to be searched or select the default values in case of the localized search. There is a find button, the click of which invokes onfind() function which give the user desired output and the result will be shown in the result panel which is tree-view and simultaneously zooms to the searched features on the map. Result will be shown in the tree-view pattern, which is done by creating xml string that is data provider of treeview. A click on a particular result in the result panel calls the glow() function to highlight that area on the map.

### 5.5.4 Tracing Tools

#### 5.5.4.1 Trace Full Feeder

This tool is the core component of SDSS so develop, it performs electrical tracing (Network Analysis) of complete feeder. Network analysis involves network trace, or tracing. The term tracing is used here to describe building a set of network elements according to some procedure. You can think of tracing as placing a transparency on top of...
a map of your network and tracing all the network elements that you want to include in your result onto the transparency.

When working with networks, tracing involves connectivity. A network element can only be included in a trace result if it is in some way connected to other elements in the trace result. The trace result is the set of network features that is found by the trace operation.

![Trace Feeder Report](image)

**Figure 5.25:** Trace full feeder tool

This network analysis tool gives the result in form of list showing the flow of electricity between from pole/DT to To pole/DT, the tool gives an option to save the tracing result in PDF format on your system. It also highlight the result on map by selecting the connected network features like Poles, Distribution Transformers(DT), cables, lines and buildings.

5.5.4.2 Trace DT Network

This tool is one of the core component of SDSS so develop, it performs electrical tracing (Network Analysis) of complete feeder. Network analysis involves network trace, or tracing. The term tracing is used here to describe building a set of network elements according to some procedure. You can think of tracing as placing a transparency on top of a map of your network and tracing all the network elements that you want to include in your result onto the transparency.

When working with networks, tracing involves connectivity. A network element can only be included in a trace result if it is in some way connected to other elements in the trace result. The trace result is the set of network features that is found by the trace operation.
This network analysis tool first ask for selecting the distribution transformer (DT) whose network needs to be traced, after selection it gives the result in form of list showing the flow of electricity between from source DT to the connected poles, the tool gives an option to save the tracing result in PDF format on your system. It also highlight the result on map by selecting the connected network features like Poles, Distribution Transformers(DT), cables, lines and buildings.

5.5.5 Analysis Tools
Various analysis tools as per the requirement of SDSS analysis are added as tool in the toolbox. These tools are meant for analysis of area selected/ drawn by the user. The various Analysis tools should be arranged in component. The component pops up by flying in when the user clicks on the icon of the tool on the Toolbox.

Since different tools would need different user action as input , the instructions are to be properly included in the interface to assist the user to make full use of the tool. The reports/charts created by the tools should also have compliance with the map so that the user can associate the data on them with that on the map

5.5.5.1 Ward wise substation tool
This analysis tool provides the list of ward wise substation i.e. number of substation in each ward and its details, by analysis various attributes from different layers, to give specific format result. This analysis includes various operations like selection, intersection, querying and analyzing.
The result is also highlighted on map on click of any ward, an info window pops up showing the technical details of its substation i.e. substation name, capacity etc.

### 5.5.5.2 DT summary report

This analysis tool provides the list of all the distribution transformers (DT) so connected to the feeder, and their details in tabular as well as chart form, by analysis various attributes from Distribution transformer layer, to give specific format result. This analysis includes various operations like selection, querying and analyzing.
The result is also highlighted on map on click of any DT, an info window pops in showing the technical details the selected transformer i.e. number of poles connected, capacity etc.

5.5.5.3 **Affected customers tool**
This tool basically helps the DISCOM officials to identify the affected customers if there is any fault in DT / pole, or if any DT/Pole is shutdown.

This tool lets the user to select DT or pole whose connected customer’s needs to be identified. When DT or pole is selected, then the tool lists the entire customer whose electrical source is the selected pole / DT, by analysis various attributes from different layers, to give specific format result. This analysis includes various operations like selection, intersection, querying and analyzing.

![Affected Customers Tool](image)

**Figure 5.29:** Affected customers tool

It also highlights all the customers on the map. User can zoom to any particular customer by clicking on it from the result list.

5.5.6 **Customer details**
This tool gives the detailed information about the consumer. With the help of customer details tool, one can search for a particular consumer whose details need to be given. The details include:
- Name of consumer
- Meter ID
- Connection Date
- Last Six Months Consumption
- Sanction Load
- Supply Voltage
- Billed unit of December Month
- Billing amount of December Month
- Billed unit of February Month
- Billing amount of February Month

The searching can be done in two major ways i.e. search by typing and search by given values.

Search by typing: it gives user an option to type the text that need to be searched, then by clicking on search button, list of consumer containing the types texted is displayed, select the required consumer from the list. Then its complete details will be given together with a highlight on the map.

![Customer Details](image)

**Figure 5.30:** (a) Customer details tool – search by typing

Search by given values: it gives user an option to select consumer by its name or Meter ID. In this case user can select the consumer by available list of name or meter ID. Once the consumer gets selected then its complete details will be given together with a highlight on the map.
5.5.7 Report Generation

Report generation tool/module is being associated with all the tools which require displaying the data in tabular form in PDF format. This module is included in tools like DT Summary report, Trace full feeder, trace DT network, customer details tool etc.

In this module the data to be visualized in report is saved in .PDF format. A window pops up for user to select the name & place where to save the report on the system. And on click event of save/generate report button, to report is saved on defined location with given name.

5.5.8 Chart and Printing Tools

5.5.9.1 Print Tool

The Print Tool prints the map in the current view. When this tool is activated the print dialog box opens.
Print tool is invoked by the user by using the print icon on the toolbar. A click on the print tool button pops up a window, which allows the user to select the size of the map/report to be printed. A click on the print tool button invokes the Print() function which helps to take pout the print of the map. It internally uses the FlexPrintJob library file. Printing option may also be used to print the output in the form of images or pdf documents.

5.5.9.2 Charting Tool

Charting tool allows the user to visualize chart as per his requirements. The charts can be of various categories. The interface to generate the desired chart should be simple and user-friendly. Most of the charts would require an input from the user. Based on these inputs the data will be retrieved and displayed in form of a chart. Once generated, these charts can be depicted in various types such as Bar Charts, Area-Charts, Pie Chart, and Doughnut etc. Best efforts should be made to make the charts lively and interactive.

Chart tool is accessible by using icon on the toolbox. A click on the chart tool button pops up a new window, where user can give the inputs. It invokes the showChart() function which uses the displaychart.mxml to display the charts according to the user’s selection option.

In this web application, charting functionality is being used in DT summary report tool. All the inputs are internally given & its respective function is also invoked internally on DT summary report tool click.