# DYNAMIC MAP COMPOSITION IN GIS

**Prepared at**

BISAG

ISO 9001:2008
ISO 27001:2013
CMMI LEVEL-5

**Bhaskaracharya Institute for Space Applications & Geo-informatics**

**Science & Technology Department, Govt. of Gujarat.**

**Gandhinagar**

**By**


| ASHUTOSH MEHTA | RUTU PATEL | RUSHI SHAH |
|---|---|---|
| 15CEUBS114 | 15CEUON117 | 15CEUON120 |


**A project submitted**

**In**
**Partial fulfilment of the requirements**
**For the degree of**

**BACHELOR OF TECHNOLOGY**
**in**
**Computer Engineering**

**Internal Guide:**

Prof. Ami Shah
Assistant Professor
Dept. Of Comp. Engg.

**External Guide:**

Dr. Manoj Pandya
Project Scientist
BISAG, Gandhinagar


**Faculty of Technology**
**Department of Computer Engineering**
**Dharmsinh Desai University**

**April 2019**

# CERTIFICATE

This is to certify that the project work titled

Dynamic Map Composition in GIS

is the bonafide work of

Mehta Ashutoh Piyush (ID No. 15CEUBS114)
Rutu Patel (ID No. 15CEUON117)
Shah Rushi Samir ID No. 15CEUON120)

carried out in the partial fulfillment of the degree of Bachelor of Technology in Computer Engineering at Dharmsinh Desai University in the academic session December 2018 to April 2019.

11/04/19

Prof. Ami Shah
Assistant Professor
Dept. of Computer Engg.

Dr. C. K. Bhensdadia
Head,
Dept. of Computer Engg.

**Faculty of Technology**
**Department of Computer Engineering**
**Dharmsinh Desai University**
**April 2019**

Bhaskaracharya Institute for Space Applications and Geo-informatics
Department of Science & Technology
Government of Gujarat
Phone: 079 - 23213081 Fax: 079 - 23213091
E-mail: info@bisag.gujarat.gov.in, website: www.bisag.gujarat.gov.in

BISAG
ISO 9001:2008
ISO 27001:2013
CMMI LEVEL-5

# CERTIFICATE

*This is to certify that the project report compiled by* ***Mr. Mehta Ashutosh Piyush, Ms. Rutu Patel. and Mr. Shah Rushi Samir*** *students of 8th Semester* ***B.TECH-CE from Dharmsinh Desai Institute Of Technology, Nadiad,*** *have completed their final semester project satisfactorily. To the best of our knowledge this is an original and bona fide work done by them. They have worked on web-based application for* ***"Dynamic Map Composition in GIS"****, starting from December 10th, 2019 to March 29th, 2019.*

*During their tenure at this Institute, they were found to be sincere and meticulous in their work. We appreciate their enthusiasm & dedication towards the work assigned to them.*

*We wish them every success.*

**Dr. Manoj Pandya**
**Project Scientist,**
**BISAG, Gandhinagar**

**T. P. Singh**
**Director,**
**BISAG, Gandhinagar**

## Acknowledgement

We would like to thank Dr. C.K. Bhensdadia, for providing us with the opportunity to pursue training at Bhaskaracharya Institute of Space and Geo-informatics, Gandhinagar (BISAG). We would also like to thank Dr. Manoj Pandya, our external guide at BISAG. He provided us with valuable inputs and understanding of our project during our tenure as student trainees. Our thanks also goes out to Prof. Ami Shah, our internal guide at DDU, for providing useful suggestions and overseeing our project documentation, implementation so as to make it parallel with industry standards. We would also like to thank the BISAG administration for providing us with useful resources and a comfortable workspace during our training period.

## Table of Contents


1.Introduction....1
1.1 Project Details....2
1.2 Technology Used....2
1.2.1 - Geoserver 2.14.1....2
1.2.2 - PostgreSQL 9.4....2
1.2.3 - PostGIS 2.0....2
1.2.4 - Eclipse Jee Oxygen with Tomcat 8.0....3
1.2.5 - MapFish Print 3.0....3
1.2.6- TIBCO Jaspersoft Studio 6.6.0....3
1.3 System Modules....3
1.4 System Hierarchy....3
2.About the System....5
2.1 Introduction....6
2.1.1 Purpose....6
2.1.2 Document Conventions....6
2.1.3 Intended Audience and Reading Suggestions....6
2.1.4 Product Scope....6
2.2. Overall Description....7
2.2.1 Product Perspective....7
2.2.2 Product Functions....7
2.3 Operating Environment....8
2.4 Design and Implementation Constraints....8
2.5 User Documentation....8
2.6 Assumptions and Dependencies....9
2.7 External Interface Requirements....9
2.7.1 User Interfaces....9
2.7.2 Hardware Interfaces....9
2.7.3 Software Interfaces....9
2.7.4 Communications Interfaces....9
2.8 System Features....10
2.8.1 User Registration....10
2.8.2 Map Printing....11

2.8.3 Zoomed Map Printing....13
2.9. Other Non-functional Requirements....13
2.9.1 Performance Requirements....13
2.9.2 Safety Requirements....14
2.9.3 Security Requirements....14
3. System Analysis....15
4. System Design....23
5. Implementation....26
6. Testing....28
7. Screenshots....33
8. Conclusion and Future Enhancements....44
9. Bibliography....46

## List of Figures


| Figure No | Figure Name | Page No |
|---|---|---|
| 1.1 | System Hierarchy | 4 |
| 2.1 | Product Perspective | 7 |
| 2.2 | Product Functions | 7 |
| 3.1 | Use Case Diagram | 16 |
| 3.2.1 | Sequence Diagram for Login | 17 |
| 3.2.2 | Sequence Diagram for Map Interaction | 18 |
| 3.2.3 | Sequence Diagram for Map Printing | 19 |
| 3.3 | Flow Chart for the system | 20 |
| 3.4 | Deployment Diagram | 21 |
| 3.5 | Timeline diagram | 22 |
| 4.1.2 | Texas Counties Shapefile Contents | 25 |
| 7.1 | Home Page | 34 |
| 7.2 | Login Page | 34 |
| 7.3 | Page Orientation Selection | 35 |
| 7.4 | Multiple Layer Selection | 35 |
| 7.5 | Map Preview | 36 |
| 7.6 | Filling Details | 37 |
| 7.7 | Sample Pdf | 38 |
| 7.8 | Code Snippets | 40 |

## Abstract

The unit function of a Geographic Information System (GIS) is to represent data in mapped form. This data can be in the form of table, spreadsheet, CSV etc. Visualization of such spatial data is more convenient in analyzing and deducing conclusions regarding land, agriculture, population, weather, amenities, transports etc. Each country provides an online service where users view spatial data regarding the country and need to take the necessary printouts. Our objective is to get an essence of the customized dynamic printing according to the user's choice. We will make use of popular GIS software - Geoserver along with Mapfish Print and Jaspersoft Studio for this purpose.

Geoserver is extensively used in BISAG, a research organization working in GIS field. Geoserver, although versatile, has a drawback in that it requires a user to go through a number of steps in order to accomplish a task. For example, in order to print spatial data in Geoserver as a map in pdf format, we need to create a workspace and publish a layer and then we need to print it manually over there and top of that one cannot customize printing options. These steps however consume time and hinder the efficiency of BISAG scientists. Therefore, in order to overcome this limitation, we have designed a user friendly tool to automate the functionalities of printing of maps in Geoserver using Mapfish Print. Using the Geoserver REST API, we can program Dynamic Map Composition without accessing Geoserver explicitly. This tool has been tailored to meet the requirements of the BISAG scientists who can use it for efficient research and development.

## 1.1 Project Details

One of the important feature after accessing the maps using the Geoserver software is of printing. The employees of BISAG have extensive use for printing the map. So to ease out the painstaking steps followed, we have developed Dynamic Map Composition Tool. Thus, the tool helps one in printing the maps with celerity, top of that, according to their customizations.

## 1.2 Technologies Used

### 1.2.1 - Geoserver 2.14.1

- Open source application server written in Java for visualizing spatial data.
- Works with all modern OS and has powerful REST API to interact with its functionalities. It is deployed locally.
- A convenient web UI for publishing layers via shapefiles, loading PostGIS tables and previewing layers.

### 1.2.2 - PostgreSQL 9.4

- Open source database used to perform CRUD operations and converting shapefiles into tables.
- Used to connect with Geoserver.
- Has a very useful configuration tool called pgAdmin III for easy and quick operation.

### 1.2.3 - PostGIS 2.1

- A useful extension which helps to load shapefiles as database tables in PostgreSQL.

### 1.2.4 - Eclipse Jee Oxygen with Tomcat 8.0

- Versatile IDE for Java based development. Supports J2EE features.
- Possesses a local server Apache Tomcat which handles the backend and deploys the application on the local browser.

### 1.2.5 – MapFish Print 3.0

- The purpose of Mapfish Print is to create reports that contain maps and its related components like legends, scale bar etc. within them.
- The distribution used here is the *war* distribution where Mapfish Print is a servlet and can be run in a servlet container like Apache Tomcat.

### 1.2.6 - TIBCO Jaspersoft Studio 6.6.0

- TIBCO Jaspersoft Studio is open source reporting engine, entirely written in Java.
- It is able to use data coming from any kind of data source and produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel, OpenOffice and Word.
- Jaspersoft Studio is used for making the layouts of the PDF reports to be generated.

**Diagram Tool:** All the UML diagrams and various other diagrams corresponding to this project are made using **draw.io**

## 1.3 System Modules

**User End Module: (Dynamic Map Composition module):**

Contains code related to printing of maps with customizations of end users.

## 1.4 System Hierarchy

The system hierarchy as shown below is divided into two main components: -

- **Module Hierarchy :**

  The Root Module is composed of the User namely Dynamic Map Printing Module. The Root module is controlled by the Eclipse IDE.

- **Tools Hierarchy :**

  The Eclipse IDE handles all the backend and UI code. It sits above the tools such as OpenLayers, Servlets MapFish Print etc. In the order and it integrates all the tools into a unit. The Eclipse IDE then proceeds to interact with the Root module and accordingly make changes in it. The Eclipse IDE is the main controller of the system and is at the heart of the system. It deploys the system on local machine.

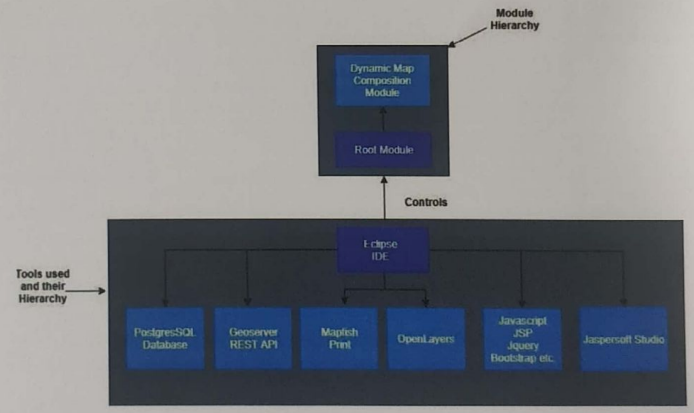



**Fig 1.1: System Hierarchy**

Figure shows the system hierarchy where the main modules and the tools have been shown along with their respective hierarchies.

# Chapter 2

## About the System

# SOFTWARE REQUIREMENT SPECIFICATION

## 2.1 Introduction

### 2.1.1 Purpose

The purpose of preparing this document is to provide a thorough analysis of the Dynamic Map Composition Using MapFish Print, also to list out the operating constraints, environment, and the effect of the system when exposed to external factors.

### 2.1.2 Document Conventions

This document serves as a reference to the software development team as well as the users of this system. The priorities for the points are mentioned accordingly. We have followed IEEE standard to prepare this document. This document is formatted in natural language sentence.

### 2.1.3 Intended Audience and Reading Suggestions

The document is intended for everyone involved with the software in one way or the other i.e. developers, users, admin.

### 2.1.4 Product Scope

This system is designed to capture the essence of Dynamic Map Composition in GIS. This system has been tailored to aid the development process of the research organization BISAG. It is not meant for industrial/commercial use and its scope is limited to BISAG use.

## 2.2. Overall Description

### 2.2.1 Product Perspective

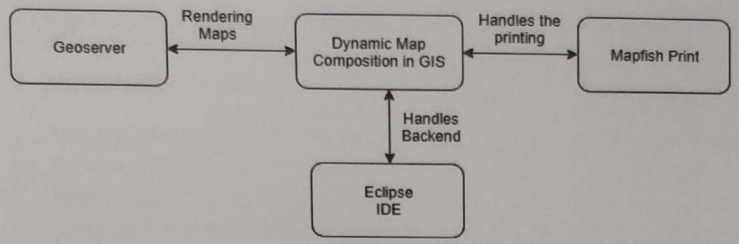



**Fig 2.1 - Product Perspective**

The main module uses Geoserver to show map, while Mapfish Print takes the maps and provides the platform for Printing. Eclipse handles the backend and integrates the system

### 2.2.2 Product Functions

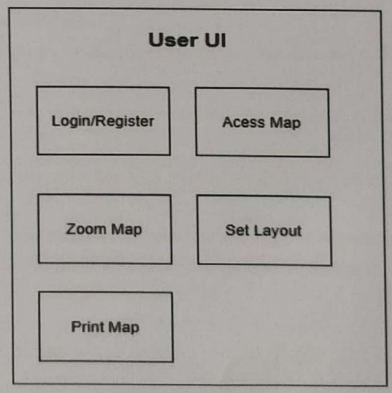



**Fig 2.2 - Product Functions**

A brief overview of the functionalities to be implemented on both the admin as well as user end.

## 2.3 Operating Environment

- **HARDWARE**

1. Intel Core2Duo, i3, i5 with a minimum of 4 GB Ram and 256 GB HDD.

- **SOFTWARE AND OS:**

1. Windows 7 and above
2. Geoserver 2.14.1
3. Openlayers 3
4. pgAdmin 3 (PostgreSQL 9.4, PostGIS 2.0)
5. Eclipse JEE Oxygen
6. MapFish Print 3
7. TIBCO Jaspersoft Studio 6.6.0

## 2.4 Design and Implementation Constraints

1. The system should run on all the latest versions of Chrome, Firefox, Opera and Microsoft Edge.
2. The system should support all modern operating systems( iOS, Windows, Linux)
3. The system should be able to run on any machine having at least 4GB of RAM and 256 GB HDD.
4. The system should be implemented in Java language using either Eclipse or NetBeans.
5. The system should be reliable, efficient in loading, have proper GUI and navigation and should have an average level of maintenance.

## 2.5 User Documentation

A CD will be provided which contains a presentation of the features and working of the system.

## 2.6 Assumptions and Dependencies

1. Users must have Internet connectivity to operate the software.
2. User can interact with one map database at a time.
3. We assume that shape file provided to the user is correct.

## 2.7 External Interface Requirements

### 2.7.1 User Interfaces

1. The user interface includes a well-designed web page of login and registration. This, as well as other constituent webpages are designed using HTML, CSS, Bootstrap and JavaScript.
2. Once the user is authenticated, they are directed to their respective home page. The back-end for the user interface will be handled by the Eclipse IDE.

### 2.7.2 Hardware Interfaces

The hardware interfaces will be handled by the IDE itself.

### 2.7.3 Software Interfaces

**Operating System:**

The software is designed to run on all modern OS.

**Database:**

1. The software uses PostgreSQL DB(Version 9.4).
2. pgAdmin is free and open source GUI administration tool for PostgreSQL.

**Geoserver:**

1. Geoserver is an open-source application server written in Java that allows users to share, process and edit geospatial data.
2. It provides features such as creating and maintaining workspaces, publishing layers and viewing layers in multiple formats.

**Jaspersoft Studio:**

1. Jaspersoft Studio is an open source application written in Java that allows to create sophisticated layouts containing charts, images, sub reports and much more.
2. The reports allows the user to have a customized printing functionality.

Backend code written in Java using Eclipse

### 2.7.4 Communications Interfaces

The system will operate on the internet via a secure (HTTPS) protocol and all its features will be accessible through the website.

## 2.8 System Features

### 2.8.1 User Registration

#### 2.8.1.1 Description and Priority

This is high priority function which performs user registration. Users have to register to use the system. This registration can be of a new user or employee account.

#### 2.8.1.2 Stimulus/Response Sequences

In registration user will provide details of them namely a username and a password. If all details satisfy the conditions then account will be successfully created and system will store all the details in database otherwise it will send error message for required fields.

#### 2.8.1.3 Functional Requirements

**R 2.8.1.3.1: Registration**

I/P : Username and password with details like e-mail id.
O/P : A message with successful registration.

Process: System will take username, password and details. It will check all details are complete or missing. If details are verified, system will show successful message to user and will be redirect to login page so user can login from there and also, system will store user details in database.

### 2.8.2 Map Printing

#### 2.8.2.1 Description and Priority

This is a key feature of user home page. It includes dynamic selection of series of layers, the page size and orientation. Thus, allows exquisite customization to the user.

#### 2.8.2.2 Stimulus/Response Sequences

When selecting the layers to be printed, user will select various map layers that needs to be print, and then the layer preview is shown on the screen. Further the user will select the page size and orientation through a dropdown list and fill out all the text fields for printing the details regarding the map on the pdf. After clicking the button a pdf file is been generated with the layers and according to the page layouts by the user.

#### 2.8.2.3 Functional Requirement

**R 2.8.2.3.1: Fetching Layers**

I/P : Selecting the layers from checkbox group (i.e. all the layers you want to be printed)

O/P : Redirected to the next page previewing the layers selected by the user

Process : Layers are selected and on clicking of button called "Get Selected Layers", the page is updated with the all the selected layers on a certain base layer.

**R 2.8.2.3.2: Selecting the page size and orientation**

I/P : Selecting the page size (i.e. A0, A1, A3 currently used) and orientation (i.e. Portrait and Landscape) from first dropdown.

O/P : -

Process : User just selects the page orientation and size for their customization.

**R 2.8.2.3.3: Filling the Map related details**

I/P : Filling out the text fields regarding the map (i.e. Map title, Junior Engineer Name, District etc.).

O/P : On clicking the button "Click here to Save Map" pdf is successfully generated with a unique name.

Process : After filling out the details of customization, on click on the button for saving the map, on next webpage shows the pdf generation status Success.

### 2.8.3: Zoomed Map Printing

#### 2.8.3.1 Description and Priority

This is an important feature of user home page which enables user to print the map up to their desired level of zooming.

#### 2.8.3.2 Stimulus/Response Sequences

There will be buttons "+" and "-" for the user to zoom in and out respectively by the user for printing a particular part of the map.

#### 2.8.3.3 Functional Requirement

**R 2.8.2.3.1: Fetch Layer**

I/P : Click "+" or "-" buttons numerously till they get the desired region of the map for printing.

O/P : Pdf generated successfully with bounding box region of the user only.

Process: When user clicks on the button of "+" and "-" and after going on the same procedure of customization the pdf file with zoomed feature is successfully created.

## 2.9. Non-functional Requirements

### 2.9.1 Performance Requirements

**Scalability:** System should be able to handle a number of users.

**Usability:** Simple interface that a layman can understand.

**Speed:** Speed of the system should be responsive i.e. response to a particular action should be available in short period of time. For e.g. in home page of user we have checkbox group for displaying the layers of maps, thus that layers should be fetched swiftly.

### 2.9.2 Safety Requirements

The database may face a risk of crash due to operating system failure. Therefore, it is required to take the database backup or to use any standby server so that there is no chance that our database will destroy. There should not be any type of SQL injection.

### 2.9.3 Security Requirements

The system shall only allow authenticated and authorized users to access the system

# Chapter 3
## System Analysis

# 3 System Analysis

## 3.1 Use Case Diagram:

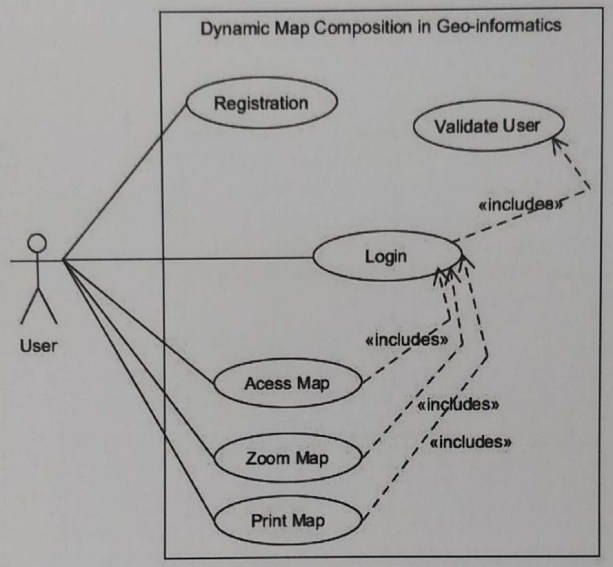



**Figure 3.1**

**Use Case diagram for Dynamic Map Composition in GIS**

This diagram shows the interaction of the different use cases for the system such as Login, Registration, Zoom and Print Map etc.

## 3.2 Sequence Diagram

### 3.2.1 Login

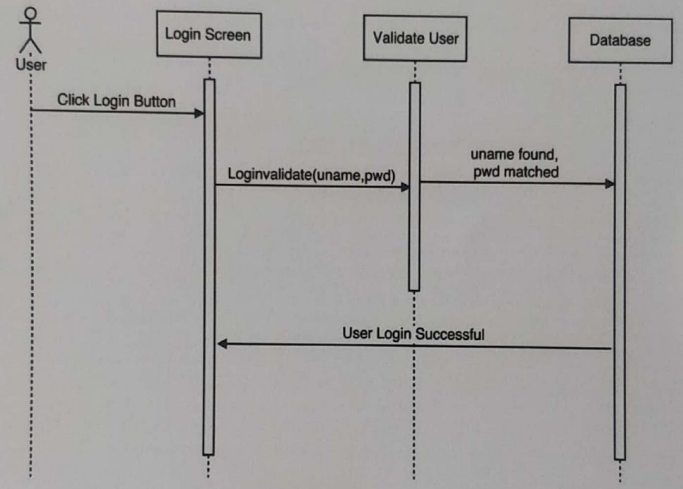



**Figure 3.2.1**

**Sequence diagram for Login process**

This diagram shows the sequence of operations involved in Login process

### 3.2.2 Map Interaction

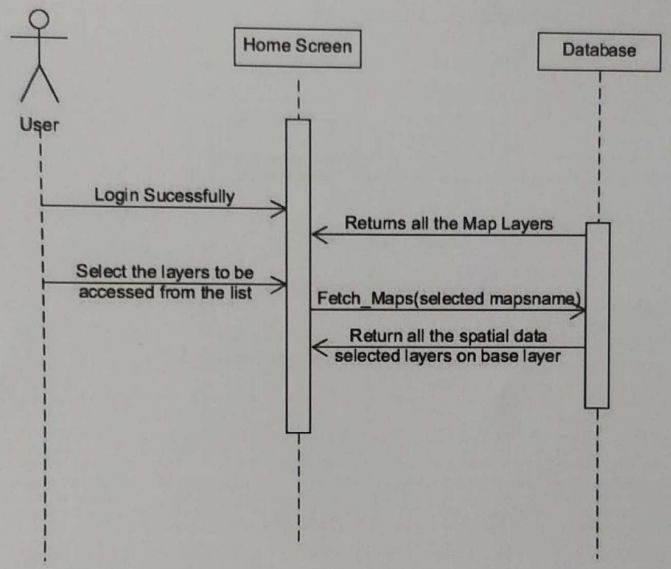



**Figure 3.2.2**

**Sequence diagram for Map Interaction process**

This diagram shows the sequence of operations involved while the user interacts with the map

### 3.2.3 Map Printing

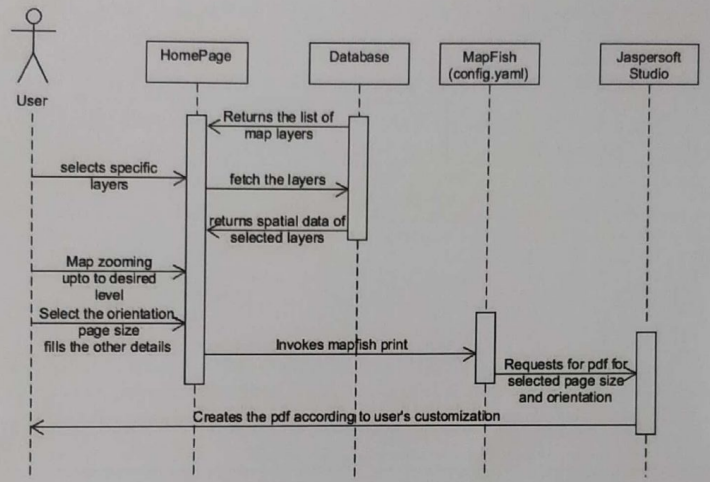



**Figure 3.2.3**

**Sequence diagram for Printing Maps**

This diagram shows the sequence of operation when a customized printing is to be done.

## 3.3 Flow Chart

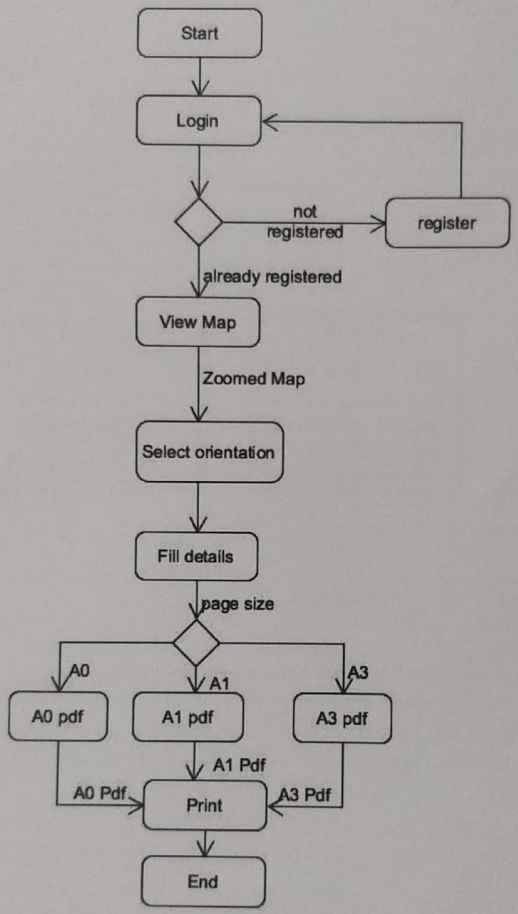



**Figure 3.3**

**System Flow Chart**

This diagram shows the flow of execution of the entire system.

## 3.4 Deployment Diagram

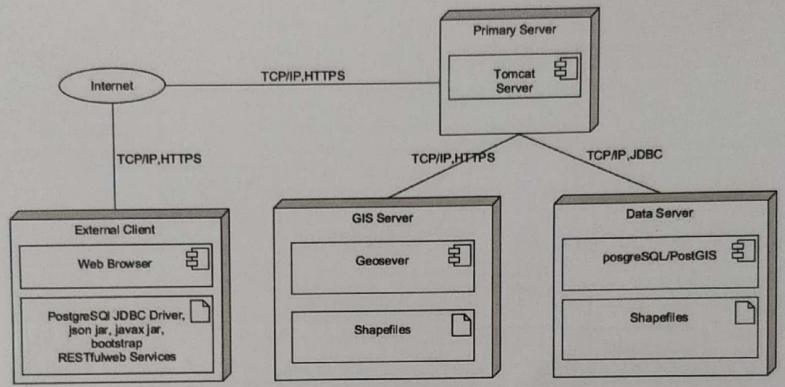



**Figure 3.4**

**Deployment diagram for Dynamic Map Composition**

This diagram is used for describing the hardware components, where software components are deployed. It includes the servers and the protocols used for communication. The servers consist of the software comprising it.

## 3.5 Timeline Diagram

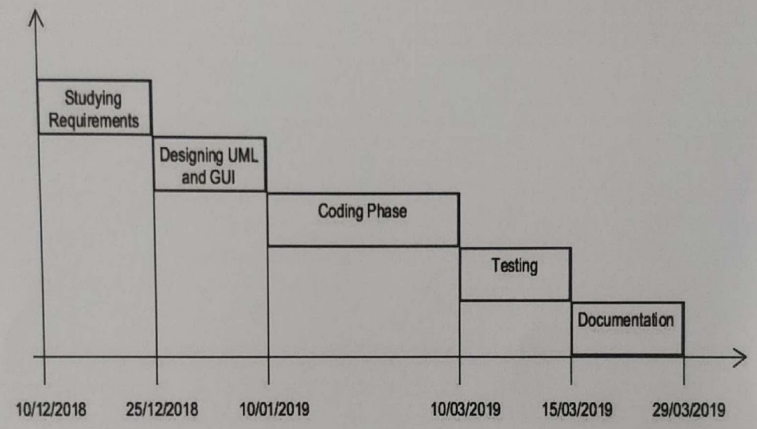



**Figure 3.5**

**Timeline Diagram**

This diagram shows the various phases of the system development starting with requirement study up to final documentation.

# Chapter 4

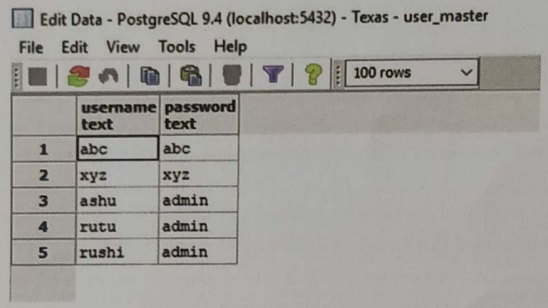
## System Design

## 4.1 Tables

The following tables are part of this systems database:

1. **User table :** Stores the user login/ register information.
2. **Data Store table** : Shapefile ( Texas Counties) loaded as a table in pgAdmin

### 4.1.1 User Table




| | username text | password text |
|---|---|---|
| 1 | abc | abc |
| 2 | xyz | xyz |
| 3 | ashu | admin |
| 4 | rutu | admin |
| 5 | rushi | admin |

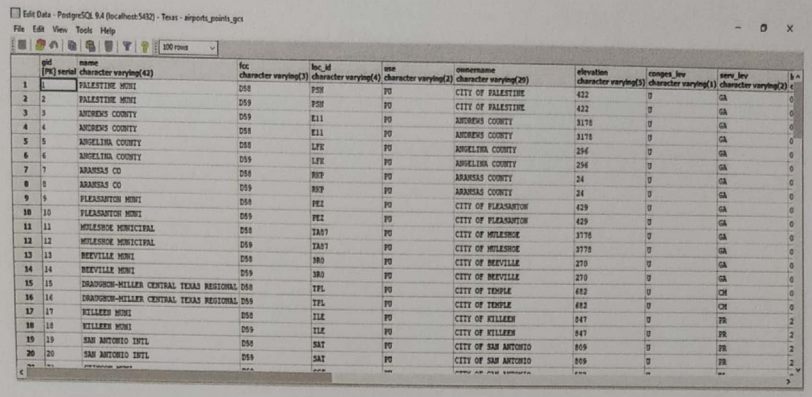
**Fig 4.1.1 -User Table**
Table containing basic user credential details

### 4.1.2 Data Store Table

This being a GIS application, the most used data format is shape file. This shapefile is composed of rows and columns and can be converted into a table by adding a datastore in Geoserver.

Here we have added Texas State Shapefile as a data store table.

### 4.1.2.1 Texas Counties shapefile


Edit Data - PostgreSQL 9.4 (localhost:5432) - Texas - airports_points_gcs

File Edit View Tools Help

100 rows

| | gid [PK] serial | name character varying(42) | fcc character varying(3) | loc_id character varying(4) | use character varying(2) | ownername character varying(29) | elevation character varying(5) | conges_lev character varying(1) | serv_lev character varying(2) | b |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | PALESTINE MUNI | D58 | PSN | PU | CITY OF PALESTINE | 422 | U | GA | 0 |
| 2 | 2 | PALESTINE MUNI | D59 | PSN | PU | CITY OF PALESTINE | 422 | U | GA | 0 |
| 3 | 3 | ANDREWS COUNTY | D59 | E11 | PU | ANDREWS COUNTY | 3178 | U | GA | 0 |
| 4 | 4 | ANDREWS COUNTY | D58 | E11 | PU | ANDREWS COUNTY | 3178 | U | GA | 0 |
| 5 | 5 | ANGELINA COUNTY | D58 | LFK | PU | ANGELINA COUNTY | 296 | U | GA | 0 |
| 6 | 6 | ANGELINA COUNTY | D59 | LFK | PU | ANGELINA COUNTY | 296 | U | GA | 0 |
| 7 | 7 | ARANSAS CO | D58 | RKP | PU | ARANSAS COUNTY | 24 | U | GA | 0 |
| 8 | 8 | ARANSAS CO | D59 | RKP | PU | ARANSAS COUNTY | 24 | U | GA | 0 |
| 9 | 9 | PLEASANTON MUNI | D58 | PEZ | PU | CITY OF PLEASANTON | 429 | U | GA | 0 |
| 10 | 10 | PLEASANTON MUNI | D59 | PEZ | PU | CITY OF PLEASANTON | 429 | U | GA | 0 |
| 11 | 11 | MULESHOE MUNICIPAL | D58 | TAD7 | PU | CITY OF MULESHOE | 3778 | U | GA | 0 |
| 12 | 12 | MULESHOE MUNICIPAL | D59 | TAD7 | PU | CITY OF MULESHOE | 3778 | U | GA | 0 |
| 13 | 13 | BEEVILLE MUNI | D58 | HBO | PU | CITY OF BEEVILLE | 270 | U | GA | 0 |
| 14 | 14 | BEEVILLE MUNI | D59 | HBO | PU | CITY OF BEEVILLE | 270 | U | GA | 0 |
| 15 | 15 | DRAUGHON-MILLER CENTRAL TEXAS REGIONAL | D58 | TPL | PU | CITY OF TEMPLE | 682 | U | CH | 0 |
| 16 | 16 | DRAUGHON-MILLER CENTRAL TEXAS REGIONAL | D59 | TPL | PU | CITY OF TEMPLE | 682 | U | CH | 0 |
| 17 | 17 | KILLEEN MUNI | D58 | ILE | PU | CITY OF KILLEEN | 847 | U | FR | 2 |
| 18 | 18 | KILLEEN MUNI | D59 | ILE | PU | CITY OF KILLEEN | 847 | U | FR | 2 |
| 19 | 19 | SAN ANTONIO INTL | D58 | SAT | PU | CITY OF SAN ANTONIO | 809 | U | FR | 2 |
| 20 | 20 | SAN ANTONIO INTL | D59 | SAT | PU | CITY OF SAN ANTONIO | 809 | U | FR | 2 |

**Fig 4.1.2 – Texas Counties Shapefile loaded as a table**

The Texas counties shapefile is a loaded as table and as seen in the figure it contains spatial data such as coordinates, name etc. in a tabular form.

# Chapter 5

# Implementation

# 5. Implementation

## 5.1 Implementation Environment

- A Web Browser (Chrome,Firefox etc.)
- Language - Java
- IDE - Eclipse Oxygen with Tomcat 8
- Javascript (for OpenLayers)
- PostgreSQL 9.4 (database)
- Jaspersoft Studio 6.6.0 (for .jrxml files)

### 5.1.1 Key Implementation Objectives

The implementation objectives are as follows:

- Simple and User-friendly UI should be developed.
- Ensure smooth working of all functionalities of application.
- Ensure that all the Users are familiar with System.

## 5.2 Programmer/Module Specifications

Main modules of system are:

- **User End Module :**

  Sample Dynamic Map Composition (Texas shapefile)

  Contains code files for printing layers of Texas Shapefile via Geoserver and also contains code for various customizations for the user.

# Chapter 6

## Testing

# 6. TESTING

## 6.1 Testing Overview

- Software Testing is the critical element of software quality assurance and represents the ultimate review of specifications, design and coding. Testing represents an interesting analogy for the software. The testing phase involves testing of the system using various test data. Preparation of the test data plays a vital role in the system testing.

- After preparing the test data, the system under study is tested using those data. Errors found were corrected and corrections were recorded for future references. Thus, a series of testing is performed on the system before it is ready for implementation

- The development of the software system involves a series of production activities where opportunities for injection of human fallibility are enormous. Errors may begin to occur at very inception of the process where the objectives may be erroneously or imperfectly specified as well as in the later design and development stages. Because of human inability to perform and communicate with perfection, software development is followed by a quality assurance activity.

## 6.2 Testing Strategy

A Correct system must accomplish the following:

- Compute correct results.
- Operate safely and cause the system containing the software to operate safely.

- Perform the tasks required by the system containing the software to operate safely.
- Perform the tasks required by the system containing the software, as explained in the software applications.
- Achieve these goals for all inputs.
- Recognize inputs outside its domain.

We shall see that satisfying these pre-requisites depends on a variety of things. One of these things is to provide clear and correct software specifications.

## Testing for Performance

After you have identified specific performance requirements, you can begin testing to determine whether the application meets those requirements. Performance testing presumes that the application is functioning, stable, and robust. As such, it is important to eliminate as many variables as possible. For example, bugs in the code can create the appearance, performance problem or even mask performance problem. This being a GIS application, the loading of maps and printing of maps should be as efficient as possible across all modern browsers with ample configuration.

## Testing for Reliability

Testing for reliability is about exercising an application so that failures are discovered and removed before the system is deployed because the different combinations of pathways through an application are high; it is unlikely that you can identify all potential failures in complex application. For a GIS application like this, reliability is measured by the ability of the system to save user data while dealing with shapefiles, data stores and workspaces.

## 6.3 Testing Methods

**Black Box Testing:**

Performed on Dynamic Map Composition Module.

### Test Case Design

Test cases have been designed for appropriate number of scenarios. Realistically, the most frequent occurring have been tested. The test cases have been designed keeping in mind the software requirements.

In the figure below, test cases as per the module have been ordered. The following order is followed:

**Test Case Nos. (1 - 14):** Dynamic Map Composition scenarios

| Sr. No | Test Case | Input | Expected Output | Actual Output | Remark Test pass or fail? |
|---|---|---|---|---|---|
| 1 | Login | Valid username and password | Redirect to select layout page. | Redirect to select layout page. | Pass |
| 2 | Login | Invalid Username and password | Remain at same page. | Remain at same page. | Pass |
| 3 | Login | Empty username and password | Remain at same page. | Remain at same page. | Pass |
| 4 | Connection to PgAdmin III | Valid Query | Shows list of Layers. | Shows list of Layers. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| 5 | Connection to PgAdmin III | Invalid Query | Shows Empty Page | Shows Empty Page | Pass |
| 6 | Select Layer | Atleast one layer selected | Map is displayed | Map is displayed | Pass |
| 7 | Select Layer | No layer selected | Shows alert " Please select atleast one layer" | Shows alert " Please select atleast one layer" | Pass |
| 8 | Select DPI and Page size | Valid Dpi and Page size | Map is downloaded | Map is downloaded | Pass |
| 9 | Select DPI and Page size | Dpi not selected and Valid Page size | Alert "Please select dpi" | Alert "Please select dpi" | Pass |
| 10 | Select DPI and Page size | Page size not selected and Valid Dpi | Alert "Please select Page Size" | Alert "Please select Page Size" | Pass |
| 11 | Select DPI and Page size | Page size and Dpi not selected | Alert "Please fill required details" | Alert "Please fill required details" | Pass |
| 12 | If an empty layer is selected | Selected Layer (Empty) | Base Layer is displayed only | Base Layer is displayed only | Pass |
| 13 | Geoserver is Off | Select Layer | No Map is displayed (Error in console" Failed to load resource"). | No Map is displayed (Error in console" Failed to load resource"). | Pass |
| 14 | Geoserver is on | Select Layer | Map is Displayed | Map is Displayed. | Pass |

**Fig 6.2 - Test Case Design for Dynamic Map Composition in GIS**

Figure shows test cases for the key features of the system. It has columns for input, expected output and actual outputs and a remarks column for determining the success of the test case. Data in each cell is self-explanatory. The order is as mentioned previously.

# Chapter 7

# Screenshots

## 7.1 - Home Page

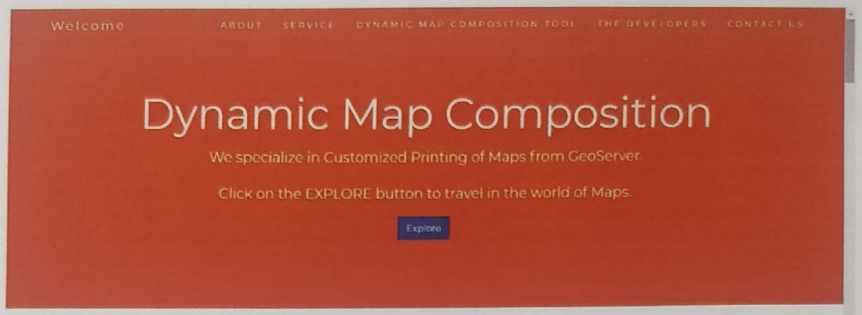



**Fig 7.1 - Home Page – Dynamic Map Composition in GIS**
Home page for the system from where user can navigate to the features

## 7.2 - Login Page

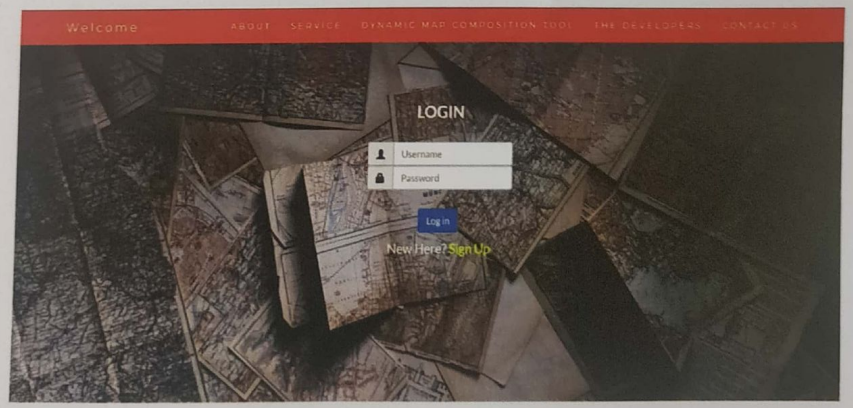



**Fig 7.2 - Login Page for user.**
Contains fields for filling user name and password and a button to log in

### 7.3 - Page Orientation

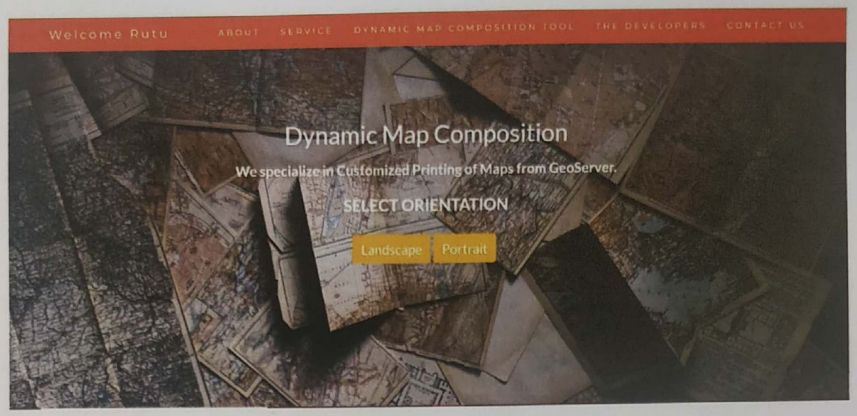



**Fig 7.3- Select between Portrait and Landscape for printing**

### 7.4 - Dynamic Map Layers Checkbox Group

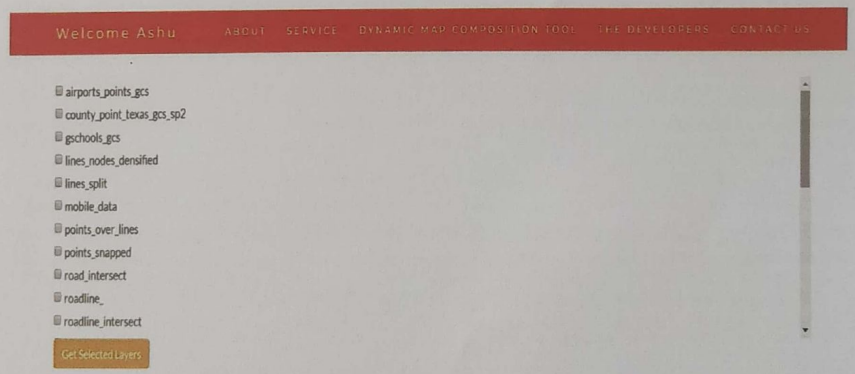



**Fig 7.4 – Tick mark the Map layers, the user want to print**

### 7.5 - Multi-Layer Map View

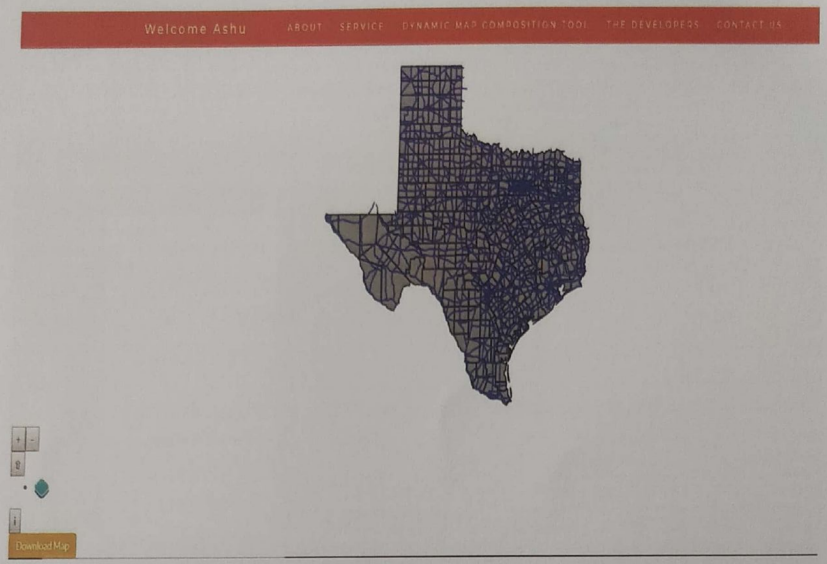



**Fig 7.5 –Preview of the layers user wants to print and using "+" and "-" can zoom in and zoom out for dynamic map printing**

### 7.6 - Filling Map Related Details

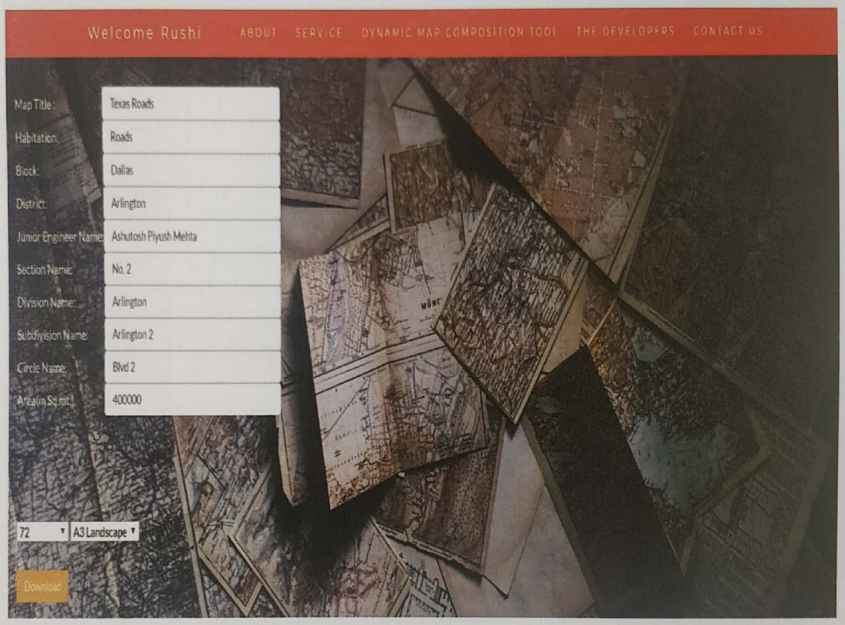



**Fig 7.6 - Adding the Map details like Map Title, Page Resolution, Page Size etc.**

### 7.7 - Sample PDF with Landscape Format

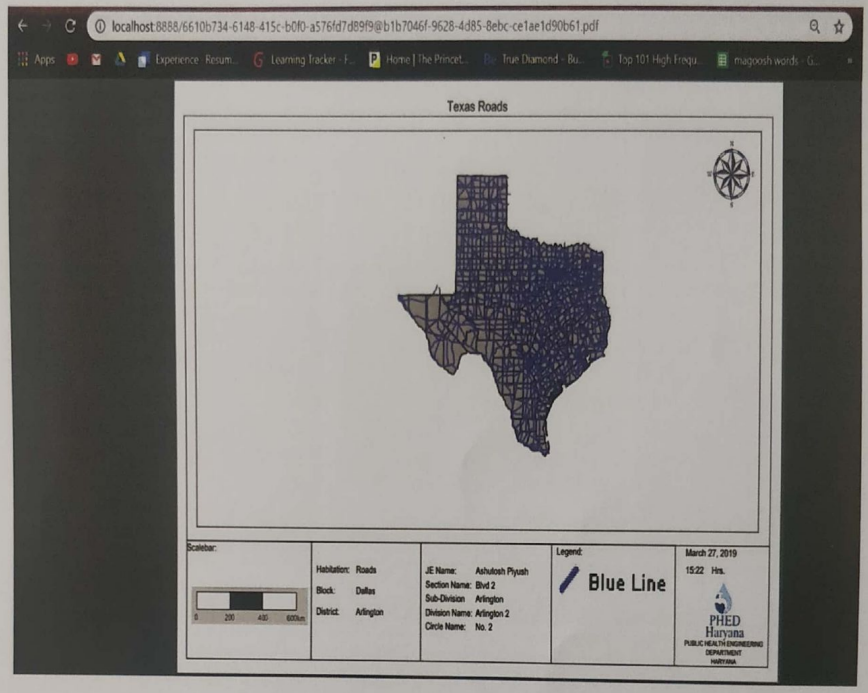



**Fig 7.7 – A3 Landscape PDF Sample with unique name**

### 7.8 - Sample PDF with Portrait Format

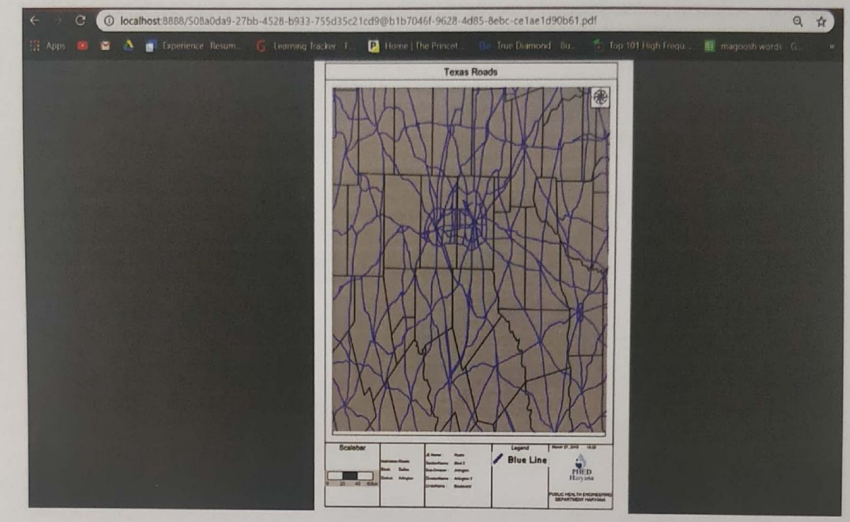



**Fig 7.8 - A0 Portrait PDF Sample with unique name.**

Code Snippet:

### 7.10 Retrieval of various layers

```
try{
    Class.forName("org.postgresql.Driver");
    conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/Texas","postgres","ashutosh");
    conn.setAutoCommit(false);
    stat = conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);

    String qry= "SELECT table_schema,table_name FROM information_schema.tables where table_schema='public' AND";
        qry+="table_type='BASE TABLE' AND table_name!='counties_texas_gcs' AND table_name!='state_texas_gcs' AND";
        qry+="table_name!='spatial_ref_sys' AND table_name!='user_master'  ORDER BY table_schema,table_name";
    rs = stat.executeQuery(qry);
    }
 catch(Exception e){
    out.println("Something went wrong !! Please try again "+e.getMessage());
    }
```

**Fig. 7.10 Code for obtaining all the layers of the database**

7.11 Binding of the selected Maps

```
if(maps.length>0)
{
map = new ol.Map({
     layers:LayersToArray(layers),
     target: 'map',
     view:new ol.View({
           center : newpos,
           zoom: 6
         })
  });
var SelLa=localStorage.getItem("SelLa");
  }else{
  alert("Please select atleast one layer");
}
```

Fig. 7.11 Code that binds the selected Maps and it displays the same

7.12 Getting the Zoomed Value of the Map invoked by user

```
function getbb()
{
     extent = map.getView().calculateExtent(map.getSize());
     mapbb = ol.proj.transformExtent(extent, 'EPSG:3857', 'EPSG:4326');
     alert(mapbb);
     localStorage.setItem("mapbb",mapbb);

}
```

Fig. 7.12 Code for getting the bounding box of the map layers

## 7.13 Parsing the Map related details in a variable

```
var lookup =
{
        "layout":document.myform.page.value,//"A3 portrait" ,
        "outputFormat": "pdf",
        "attributes": {
          "MapTitle":document.myform.MapTitle.value,
          "Habitation":document.myform.Habitation.value,
          "Block":document.myform.Block.value,
          "District":document.myform.District.value,
          "Logo":"file://LOGO_new.png",
          "North":"file://North.png",
          "JEName":document.myform.JEName.value,
          "SectionName":document.myform.SectionName.value,
          "SubdivisionName":document.myform.SubdivisionName.value,
          "DivisionName":document.myform.DivisionName.value,
          "CircleName":document.myform.CircleName.value,
          "map": {
            "projection": "EPSG:4326",
            "dpi": document.myform.dpi.value,
            "rotation": 0,
            "bbox": [mapbb[0],mapbb[1],mapbb[2],mapbb[3]],//[-101.25,33.75,-9
            "longitudeFirst": true,
            "layers": [{
              "type": "WMS",
              "layers":[layerarray[0]],//["airports_points_gcs"],
              "baseURL": "http://localhost:8085/geoserver/cite/wms",
              "imageFormat": "image/png",
              "version": "1.1.0",
              "customParams":{"TRANSPARENT":"true","EXCEPTIONS":"INIMAGE"}
            },
            {
                  "type": "WMS",
                  "layers": [layerarray[1]],//["counties_texas_gcs"],
                  "baseURL": "http://localhost:8085/geoserver/cite/wms",
```

**Fig. 7.13 Variable lookup defined for parsing all the details related to the map**

## 7.14 Request for printing

```
$.ajax({
    url:'http://localhost:8088/print-servlet_student_try2/print/simple/buildreport.pdf',
    type: 'POST',
    data: JSON.stringify(lookup),
    contentType: 'application/json',
    dataType: 'application/pdf',
    crossDomain: true,
    beforeSend: function()
    {
        $('#loader').css("visibility","visible");
        document.getElementById("loader").visibility="visible";
    },
    complete: function(data,textStatus)
    {
            window.location="window.jsp";
    }
    });
```

**Fig. 7.14 AJAX request for printing**

## 7.15 Glimpse of config.yaml file

```
templates:
  #------------------------------------------------------------------------
  A3 portrait: !template
  #------------------------------------------------------------------------
    reportTemplate: A3_P.jrxml
    attributes:
     MapTitle: !string {}
     Habitation: !string {}
     Block: !string {}
     District: !string {}
     JEName: !string {}
     SectionName: !string {}
     SubdivisionName: !string {}
     DivisionName: !string {}
     CircleName: !string {}
     Logo: !string {}
     North: !string {}

     #Use one of the following
     WMSLegend: !string {}


     map: !map
        maxDpi: 800
        width: 730
        height: 850
     scalebar: !scalebar
        width: 195
        height: 70
        default:
          backgroundColor: "rgba(214, 214, 214, 200)"
          align: "left"

     northArrowDef: !northArrow
```

**Fig. 7.15 Code in the file config.yaml, with the aid of which the printing is done**

# Chapter 8

# Conclusion and Future Enhancements

## 8.1 Conclusion

Through the medium of this project, we arrive at the conclusion that this system has been satisfactorily implemented as per the documented requirements with appropriate design and testing procedures followed. There is room for improvement in terms of features once the future extensions are added.

This system has been successfully deployed on the latest version of Firefox (v58.01), as well as Chrome (v65.0) and Edge (v41.16). This system is compatible with all the previous versions as well.

Our lightweight tool is sufficient enough to handle the requirements of BISAG scientists however it can be improved to add further features so as to make Dynamic Map composition as a standalone tool. It is a very fast and powerful approach to configure Printing.

As far as the Dynamic Map Composition module is concerned, the OpenLayers format provides sufficient loading and rendering speed and the performance is not impacted when deployed on any local browser.

Overall, the system has been designed by keeping a Waterfall Model approach in mind such that future changes can be easily made without much ado.

## 8.2 Future Enhancements

- Add Custom Map Databases :

Due to time constraint only Texas state database is added. Later on custom database selection feature will be added.

- Dynamic Legend Display for all selected Layers :

Currently, legends appearing on the pdf are merely of the base layer. Further, the legends of all the layers selected by the user will be displayed.

# Chapter 9

# Bibliography

## 9.1 References


1. www.openlayers.com/examples
2. http://mapfish.github.io/mapfish-print-doc/#/overview
3. https://community.jaspersoft.com/wiki/designing-report-jaspersoft-studio
4. https://docs.geoserver.org
5. www.jquery.com
6. www.w3schools.com
7. www.stackoverflow.com
8. www.getbootstrap.com
9. www.materializecss.com/docs
10. https://gis.stackexchange.com
11. https://www.postgis.net/docs/
12. https://www.postgresql.org/docs/