

EARLY PREDICTION OF DROUGHT

Submitted in partial fulfillment of the requirements
for the award of

Bachelor of Engineering degree in Computer Science and Engineering

by

K. ANAND

(Reg. No. 37110333)

K. S. V. ARAVIND

(Reg. No. 37110344)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC JEPPIAAR NAGAR, RAJIV
GANDHI SALAI, CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **K. ANAND (Reg. No. 37110333)** and **K. S. V. ARAVIND (Reg. No. 37110344)** who carried out the project entitled "**EARLY PREDICTION OF DROUGHT**" under my supervision from **August 2020 to March 2021**.

Internal Guide

Dr.R. Sethuraman..Ph.D.,

Head of the Department

Dr. S. Vigneswari,M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **K. Anand** and **K. S. V. Aravind** here by declare that the Project Report entitled “**Prediction of Drought**” is done by me under the guidance of **Dr.R.Sethuraman.ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.R.Sethuraman. Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

One of the major challenges of agricultural systems is how to mitigate the impacts of droughts. To mitigate the impacts of drought an effective and timely monitoring system is required. Effective monitoring of droughts can aid in developing an early warning system. Data mining extracts the valuable information from such large data archives in an automated and efficient way. Most of the predictions primarily depends on precipitation rate. Precipitation rate is a major factor in causing drought but when predicting for a localized area there are certain other factors which can also have a potential impact on drought. So this methodology involves basic parameters like humidity, temperature, ground water level, rainfall, wind speed for predicting drought. This is also a less complex method in predicting drought comparing with existing drought predicting methodologies.

TABLE OF CONTENTS

CHAPTER NO **TITLE** **PAGE NO**

1	INTRODUCTION	1
2	LITERATURE SURVEY	3
3	PRESENT SYSTEM AND IT'S DRAWBACK'S	
	3.1 EXISTING SYSTEM 3.2 DISADVANTAGES OF EXISTING SYSTEM 3.3 PROPOSED SYSTEM	12
4	SYSTEM ARCHITECTURE	14
5	SYSTEM REQUIREMENTS 5.1 HARDWARE REQUIREMENTS 5.2 SOFTWARE REQUIREMENTS	15
6	INTRODUCTION OF JAVA 6.1 APPLICATION OF JAVA 6.2 FEATURES OF JAVA 6.3 COLLECTION FRAMEWORK	16
7	SOFTWARE ENVIRONMENT 7.1 Python 7.2 History of Python 7.3 Python Features	21

8	REQUIREMENT ANALYSIS 8.1 FUNCTIONAL REQUIREMENTS 8.2 NON- FUNCTIONAL REQUIREMENTS	30
9	FEASIBILITY STUDY 9.1 ECONOMICAL FEASIBILITY 9.2 TECHNICAL FEASIBILITY 9.3 SOCIAL FEASIBILITY	33
10	MODULES 10.1 Analysis Module 10.2 Drought Prediction Module 10.3 Calculate Severity Index 10.4 View information Module	35
11	SYSTEM DESIGN AND TESTING PLAN 11.1 INPUT DESIGN 11.2 OUTPUT DESIGN	37
12	SYSTEM TESTING 12.1 Types of testing	39
13	UML DIAGRAM 13.1 ADVANTAGES 13.2 USECASE DIAGRAM 13.3 SEQUENCE DIAGRAM 13.4 COLLABORATION DIAGRAM	43

	13.5 ACTIVITY DIAGRAM	
14	RESULTS AND OUTPUT	48
15	CONCLUSION	50
16	FUTURE SCOPE 16.1 REFERENCES	51

CHAPTER 1

INTRODUCTION

India is an agricultural country and every year it faces drought conditions in the states of Maharashtra, Karnataka, Andhra Pradesh, Odisha, Gujarat, Telangana and Rajasthan which obstructs agricultural production and affects farmers and livestock adversely. India mainly experiences three types of drought- meteorological drought, hydrological drought and agricultural drought. Currently, there is no active and robust drought prediction system implemented in Maharashtra, only drought assessment and monitoring systems are available, for example, NADAMS viz. National Agricultural Drought Assessment and Monitoring System, provides real-time information on persistence and severity of agricultural droughts, which are not able to predict drought beforehand. It is important to analyze the impacts of drought since it directly affects the agriculture, food security, livestock, humanity, etc. Droughts are such a calamity which has been not studied and analyzed properly in Maharashtra, however studies and research are done for the same. Drought indices are the numerical depictions of the drought severity based on the inputs provided such as climatological and meteorological inputs. There are several drought indices available and no single index is applicable to all kinds of drought. The selection of index is dependent on the climatic conditions that trigger drought in a region, basin or location and if the same climatic conditions are inputs for calculation of indices as well. For the climatic conditions of Maharashtra, Standard Precipitation Index (SPI) and Normalized Difference Vegetation Index (NDVI) are mainly used for drought assessment. SPI was found by McKee et. al, in 1992 in Colorado state university [19]. It uses historical precipitation data to compute SPI for any

number of timescales and is used to monitor meteorological drought.

NDVI was developed by Tarpley et. al. and Kogan with National Oceanic and Atmospheric Administration (NOAA). It is used to effectively assess and monitor agricultural drought. A model or system is needed to accurately predict A drought is a prolonged shortages in the water supply, whether atmospheric, ground water or surface water in an area. It can substantially affect the biological community of the influenced area and damage to the nearby ecosystem. Continuous drought and water levels below the minimum leading to direct danger for amphibian life. Drought is an adverse environmental phenomenon that influences almost all aspects of society. It is one of the features of climate and its occurrence is unavoidable. among all extreme climate events, drought is the most complex phenomenon affecting more people. Identifying past droughts and predicting future ones is very vital in limiting their effects. however, the random and nonlinear nature of drought variables makes accurate drought prediction remain a challenging scientific problem. This method predicts drought by assessing impact of individual factors on drought ,analyze the relationship in them and draws the conclusion regarding the drought.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

Short term drought prediction by Anteneh Belanyeh and Jan Malinowski

Medium range soil moisture drought forecast helps in decision making in the

field of agriculture and water resources management. Part of skills in medium

range drought forecast comes from precipitation. Proper evaluation and correction of precipitation forecast may improve drought predictions. Here, we evaluate skills of ensemble mean precipitation forecast from Global Ensemble Forecast System (GEFS) for medium range drought predictions over India. Climatological mean (CLIM) of historic data (OBS) are used as reference forecast to evaluate GEFS precipitation forecast. Analysis was conducted based on forecast initiated on 1st and 15th dates of each month for lead up to 7-days. Correlation and RMSE were used to estimate skill scores of accumulated GEFS precipitation forecast from lead 1 to 7-days. Volumetric indices based on the 2X2 contingency table were used to check missed and falsely predicted historic volume of daily precipitation from GEFS in different regions and at different thresholds. GEFS showed improvement in correlation of 0.44 over CLIM during the monsoon season and 0.55 during the winter season. Lower RMSE was showed by GEFS than CLIM. Ratio of RMSE in GEFS and CLIM comes out as 0.82 and 0.4 (perfect skill is at zero) during the monsoon and winter season, respectively. We finally used corrected GEFS forecast to derive the Variable Infiltration Capacity (VIC) model, which was used to develop short-term forecast of hydrologic and agricultural (soil moisture) droughts in India.

Meteorological Drought Prediction Using a Multi-Model Ensemble Approach

The current NMME SPI prediction framework is similar to the CFSv2 SPI prediction system that developed by Yoon et al. (2012). For each model, monthly-mean precipitation (P) forecasts were first bias corrected and spatially downscaled (BCSD) to regional grids of 0.5-degree resolution over the contiguous United States based on the probability distribution functions (PDFs)

derived from the hindcasts. As a result, BCSD scheme corrects both the

climatological mean and standard deviation of the hindcasts. Specifically, for each month and lead time, the PDF at each grid point is computed based on model hindcasts excluding the target year. The bias-corrected percentile for the target year is then obtained from the inverse PDF of the P analysis based on the percentile calculated from the PDF of the hindcasts. The BCSD method was applied to each member and each lead of the P hindcasts. The corrected P forecasts were then appended to the CPC Unified Precipitation Analysis (Chen et al. 2008) to form a P time series for computing 1-month, 3-month, 6-month, and 12-month SPIs. The NMME-ensemble SPI forecasts are the equally weighted mean of the six model forecasts. Two performance measures, the anomaly correlation coefficient (ACC) and root-meansquare errors (RMSE) against the observations, are used to evaluate forecast skill. In this study, CPC Unified Precipitation Analysis is used as the observations for forecast evaluation.

DROUGHT INDICES Andrej Ceglar Biotechnical faculty

Analyzes climate of each location and adjusts duration coefficients (empirical factors) accordingly

- Duration factors affect the sensitivity of the index to moisture deficits by determining how much weight is given to the current moisture anomaly and the previous index value
- Two sets of duration coefficients are calculated for every location (one for wet spells, other for dry spells)
- climate characteristic coefficient calibration: 2 % of PDSI values fall below -4.0 and 2 % rise above 4.0 => this gives index an upper and lower bounds

- 2 % of extreme dry events and 2 % of extreme wet events

- Self calibrated PDSI performs much more consistently than the original PDSI (Wells et al., 2004)* and provides more realistic metric of relative periods of drought or excessive soil moisture.

Methods for predicting drought occurrences

A comprehensive assessment of future dry events in a region is essential for finding sustainable solutions for water-related problems concerning water management and preventive risk assessment of drought. It is also a first step in developing a common European strategy for facing up to the negative impacts of present and future dry climatic events. A useful index for drought monitoring, based only on monthly precipitation, is the Standardized Precipitation Index (SPI); here we assume that it properly describes the climatic condition of a particular region. By applying an appropriate forecast method to the precipitation time series and then computing the SPI, it is possible to predict future drought occurrences. Forecasting time series is a well-known field of Statistics and methods are well documented in the scientific literature. In the present paper we apply a standard technique and we contrast it with a new method here presented. As a test case, we use rain gauge data for Sicily, which may be considered a key region for understanding climatic conditions in the Mediterranean basin.

Drought forecasting and monitoring

Drought forecasting and monitoring can be defined as data collection, analysis, and reporting to develop information on current and future drought conditions. Observing and understanding current and historical weather, soil, and hydrologic conditions during droughts goes hand-in-hand with predicting future droughts, as these observations are the main inputs for computer models used in

drought forecasting. Drought reports derived from monitoring data and forecasts

simulated in computer models typically report drought in terms of the extent, severity, and duration of drought conditions. There is not a universally accepted way of measuring/forecasting drought; however the various measures used to create maps are largely calculated from temperature, precipitation, streamflow, and soil moisture data. Canada's official drought reports (Canadian Drought Monitor, CDM) rely on multiple calculations and datasets, including satellite imagery and well-established indices, such as the Palmer Drought Severity Index (PDSI) and the Standardized Precipitation Index (SPI). The CDM is a map of current drought conditions while forecasts use the same calculations and include future inputs.

Prospects for Advancing Drought Understanding, Monitoring, and Prediction

This paper summarizes and synthesizes the research carried out under the NOAA Drought Task Force (DTF) and submitted in this special collection. The DTF is organized and supported by NOAA's Climate Program Office with the National Integrated Drought Information System (NIDIS) and involves scientists from across NOAA, academia, and other agencies. The synthesis includes an assessment of successes and remaining challenges in monitoring and prediction capabilities, as well as a perspective of the current understanding of North American drought and key research gaps. Results from the DTF papers indicate that key successes for drought monitoring include the application of modern land surface hydrological models that can be used for objective drought analysis, including extended retrospective forcing datasets to support hydrologic reanalyses, and the expansion of near-real-time satellite-based monitoring and

analyses, particularly those describing vegetation and evapotranspiration. In the

area of drought prediction, successes highlighted in the papers include the development of the North American Multimodel Ensemble (NMME) suite of seasonal model forecasts, an established basis for the importance of La Niña in drought events over the southern Great Plains, and an appreciation of the role of internal atmospheric variability related to drought events. Despite such progress, there are still important limitations in our ability to predict various aspects of drought, including onset, duration, severity, and recovery. Critical challenges include (i) the development of objective, science-based integration approaches for merging multiple information sources; (ii) long, consistent hydrometeorological records to better characterize drought; and (iii) extending skillful precipitation forecasts beyond a 1-month lead time.

Drought Forecasting Using the Standardized Precipitation Index

Unlike other natural disasters, drought events evolve slowly in time and their impacts generally span a long period of time. Such features do make possible a more effective drought mitigation of the most adverse effects, provided a timely monitoring of an incoming drought is available. Among the several proposed drought monitoring indices, the Standardized Precipitation Index (SPI) has found widespread application for describing and comparing droughts among different time periods and regions with different climatic conditions. However, limited efforts have been made to analyze the role of the SPI for drought forecasting. The aim of the paper is to provide two methodologies for the seasonal forecasting of SPI, under the hypothesis of uncorrelated and normally distributed monthly precipitation aggregated at various time scales k . In the first methodology, the auto-covariance matrix of SPI values is analytically derived, as a function of the statistics of the underlying monthly precipitation process, in order to compute the transition probabilities from a current drought condition to another in the future. The proposed analytical approach appears particularly valuable from a practical stand point in light of the difficulties of applying a frequency approach due to the limited number of transitions generally observed even on relatively long SPI records. Also, an analysis of the applicability of a Markov chain model has revealed the inadequacy of such an approach, since it leads to significant errors in the transition probability as shown in the paper. In the second methodology, SPI forecasts at a generic time horizon M are

analytically determined, in terms of conditional expectation, as a function of

past values of monthly precipitation. Forecasting accuracy is estimated through an expression of the Mean Square Error, which allows one to derive confidence intervals of prediction. Validation of the derived expressions is carried out by comparing theoretical forecasts and observed SPI values by means of a moving window technique. Results seem to confirm the reliability of the proposed methodologies, which therefore can find useful application within a drought monitoring system.

AGRICULTURAL DROUGHT FORECASTING AND WARNING SYSTEM

Damage from agricultural droughts spread gradually on wide areas over a long period. In addition, due to climate change, the frequency and intensity of drought have increased. The drought of 2015 in Korea has been the worst due to lowest rainfall. It has been suggested that Korean government revisit the agricultural water management policy and incorporate in it the drought forecasting and warning system for effective drought management. In this study, new information of preparedness and mitigation planning for agricultural drought in Korea, including a long term plan and water saving strategy has been suggested. It analyses the previous drought status using the agricultural drought map, the prediction information related to drought damages, and the drought-related response capability of irrigation facilities. The system can be used to provide current and future drought conditions and the comprehensive data of drought evaluation for supporting decision makers. Also, the agricultural drought forecasting and warning system can give an opportunity to build the proper response for agricultural drought holistically and systematically.

Methods and Tools for Drought Analysis and Management

Implementation of effective drought management policies requires both

advanced technologies and appropriate methods. Monitoring and forecasting

systems, practical tools for risk assessment, as well as simple and objective criteria to select and implement appropriate drought mitigation measures are key elements for a successful drought management strategy. These key issues have been tackled by universities and public agencies involved in the EU projects Sedemed and Sedemed II (Programme Interreg IIIB MEDOCC), aimed at the definition of an integrated network for real time monitoring of drought, the development of common methodologies for drought analysis and forecasting, as well as the definition of proper mitigation strategies for the Mediterranean countries. The book presents the main outcomes of such projects with a special focus on: drought monitoring and forecasting techniques at different spatial scales; new or modified agrometeorological indices and remote sensing technique for drought identification and characterization; tools to improve surface water resources management under drought conditions; methods and tools for groundwater resources monitoring and management, based on hydrogeological and hydrodynamics characteristics of aquifers; general criteria to select and implement mitigation strategies to prevent or minimize drought impacts.

MONITORING AND MANAGEMENT OF GROUNDWATER UNDER DROUGHT CONDITIONS

California is the only state in the nation with no statewide groundwater regulation. It is clear that for California to address current and future water challenges, the state must adopt a more comprehensive approach to groundwater management. California has recently taken bold steps to re-think its approach to groundwater. Yet, a commitment to sustainable groundwater

management is only as good as the metrics that will be used to assess progress. Further, climate change is already altering the timing and reliability of California's water supplies. Groundwater management must incorporate the state's changing hydrology and its impact on groundwater recharge and use. This paper reviews a number of approaches to groundwater management. California's groundwater policy can benefit from the lessons learned elsewhere. Specifically, California's groundwater policy should provide clear metrics to define and measure sustainable yield, explicitly incorporating climate change.

CHAPTER 3

PRESENT SYSTEM AND IT'S DRAWBACK'S

3.1 EXISTING SYSTEM

- The existing system for drought forecasting uses many techniques like data mining, machine learning and deep learning.
- The technique of data mining is introduced to find associations between drought and several oceanic and climatic indices and thus predicting the drought.
- Machine learning methods have also been used recently to forecast droughts, using drought indicators such as the Standardized precipitation Index (SPI) as data input.
- Machine learning techniques like artificial neural networks (ANNs), support vector regression (SVR), and coupled wavelet-ANNs, pre-process input data using wavelet analysis and the forecast results of all three methods were compared using two performance measures.
- Drought can be forecasted by using machine learning based on remote sensing and long range forecasting data.
- Another method is Deep Belief Network consisting of two Restricted Boltzmann Machines for long-term drought prediction using lagged values of Standardized Streamflow Index (SSI) as inputs. This method is applied to predict different time scale drought indices.

3.2 DISADVANTAGES OF EXISTING SYSTEM

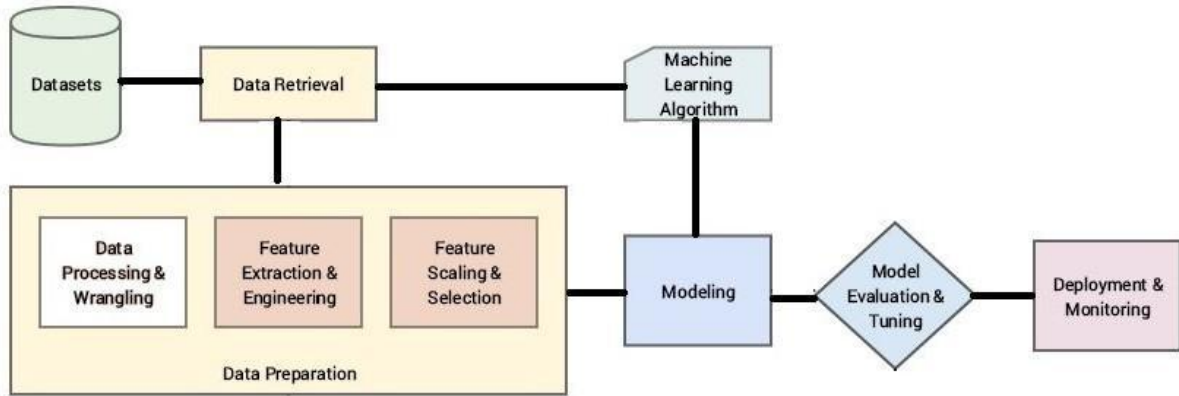
- Individual assessment of each and every causative parameters is not involved in the existing methods.
- There is a need for a tool to predict the water scarcity in an area, sometimes as small as size of a range of 1000 ha.
- Most of the predictions primarily depends on precipitation rate .Precipitation rate is a major factor in causing drought but when predicting for a localized area there are certain other factors which can also have a potential impact on drought .

3.3 PROPOSED SYSTEM

- ☐ The proposed system assess the parameters contributing to drought and predicts the probability of drought occurrence.
- ☐ If the occurrence of the drought is detected, then the severity of the drought is determined.
- ☐ This is done by analysing the historical values of the causative parameters and predicting the upcoming values.
- ☐ Machine learning algorithm is used for analytic purpose.
- ☐ Thus the predicted values is assessed to determine the drought probability.

CHAPTER 4

SYSTEM ARCHITECTURE



CHAPTER 5

SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS:

- System - Pentium-IV
- Speed - 2.4GHZ
- Hard disk - 40GB
- Monitor - 15VGA color
- RAM - 512MB

5.2 SOFTWARE REQUIREMENTS:

- Operating System - Windows XP
- Coding language - Java
- IDE - Eclipse

CHAPTER 6

INTRODUCTION OF JAVA

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown.

6.1 APPLICATION OF JAVA

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

6.2 FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features,

there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

3) Robust

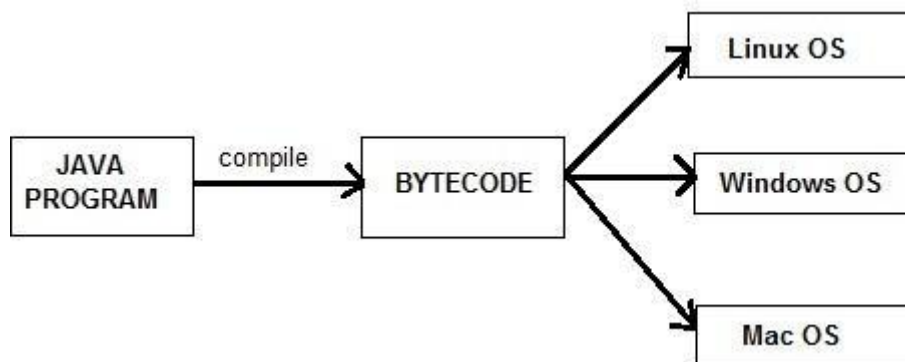
Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once,

run-anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime Environment can run Java Programs.



5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7) Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

8) Portable

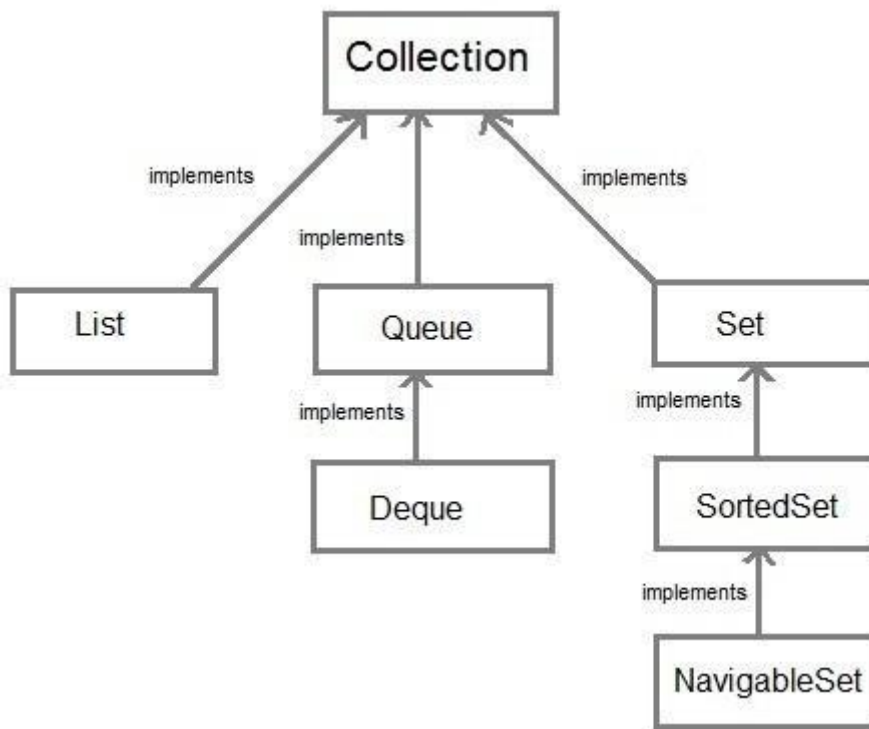
Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types.

10) High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

6.3 COLLECTION FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.



CHAPTER 7

SOFTWARE ENVIRONMENT

7.1 Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

7.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

7.3 Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.

- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.

- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>> print "Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods –

Sr.No	Methods & Description
1	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<form action="http://localhost:5000/login" method="post">

<p>Enter Name:</p>

<p><input type="text" name="nm"/></p>

<p><input type="submit" value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask import Flask, redirect, url_for, request

app=Flask( name )

@app.route('/success/<name>')

def success(name):

return 'welcome %s'% name
```

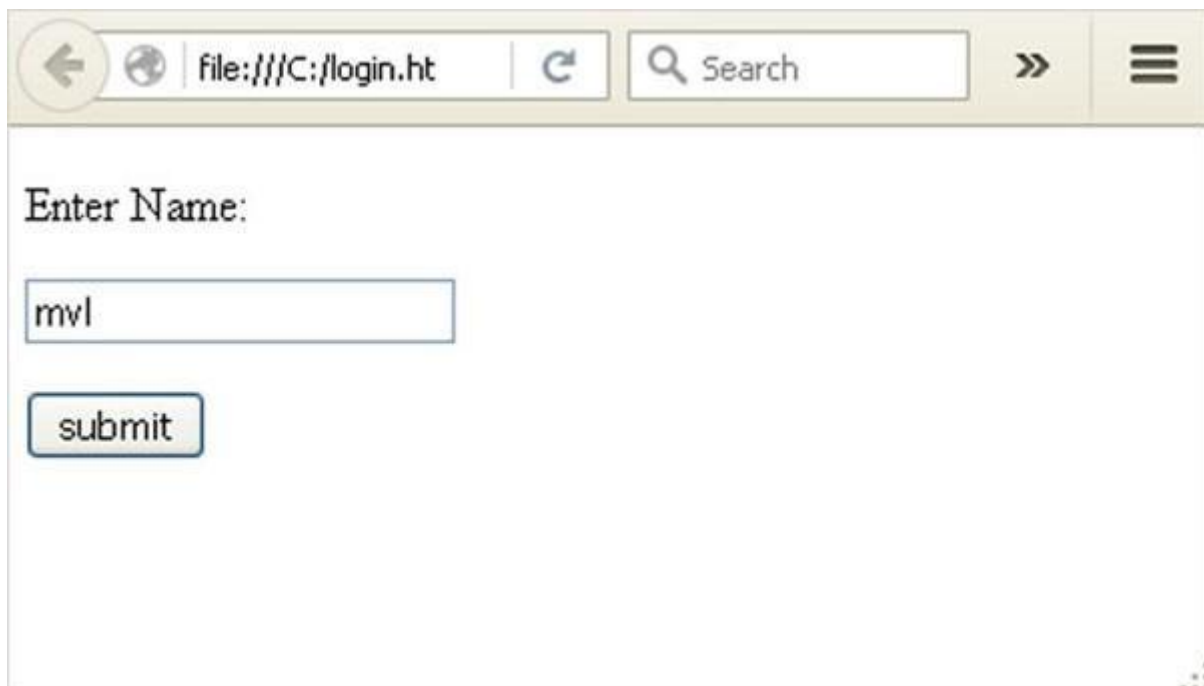
```

@app.route('/login',methods=['POST','GET'])
def login():
    if request.method=='POST':
        user=request.form['nm']
        return redirect(url_for('success',name=
        user))
    else:
        user=request.args.get('nm')
        return redirect(url_for('success',name=
        user))
if __name__ == '__main__':

```

```
app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.



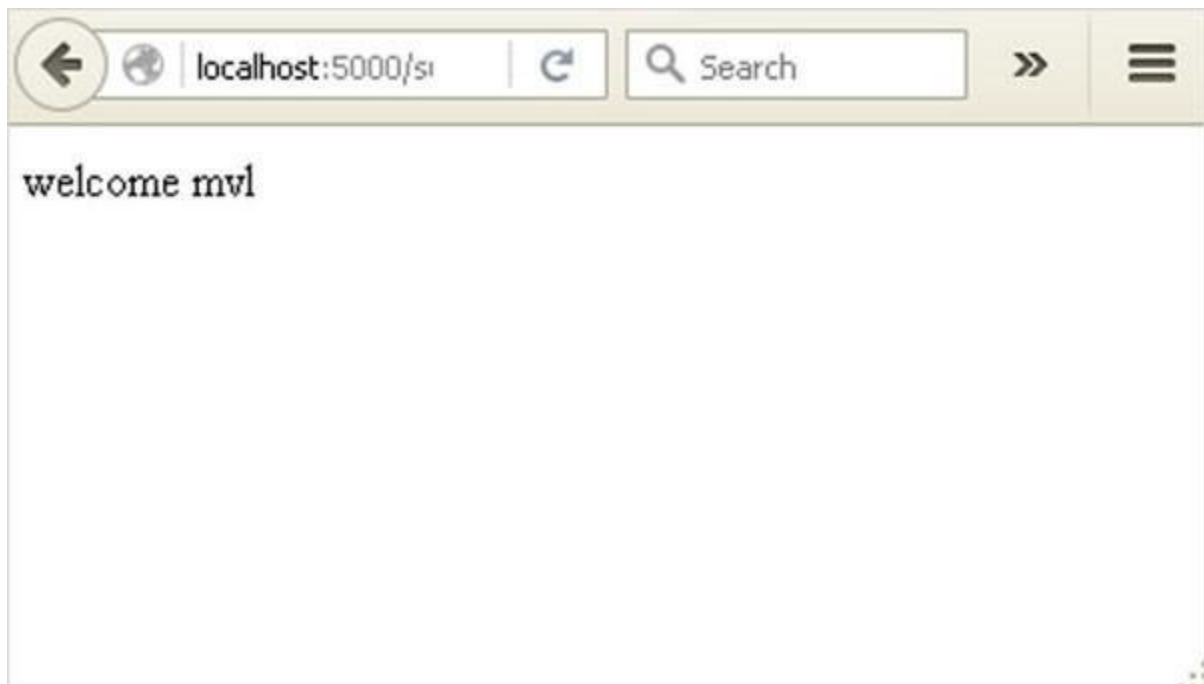
The screenshot shows a web browser window with the address bar displaying 'file:///C:/login.ht'. The main content area contains a form with the label 'Enter Name:'. Below the label is a text input field with the value 'mvl'. Underneath the input field is a button labeled 'submit'.

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by –

```
user = request.form['nm']
```

It is passed to **‘/success’** URL as variable part. The browser displays a **welcome** message in the window.



Change the method parameter to **‘GET’** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by –

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to **‘/success’** URL as before.

CHAPTER 8

REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

8.1 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is has the ability to run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

8.2 NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The

project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

CHAPTER 9

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

9.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour

into the research and development of the system is limited. The expenditures

must justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

9.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

9.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

CHAPTER 10

MODULES

- Analysis Module
- Drought Prediction Module
- Calculate Severity index
- View Information Module

10.1 Analysis Module

In this Module, analysing the historical values of the causative parameters and predicting the upcoming values. Machine learning algorithm is used for analytic purpose.

10.2 Drought Prediction Module

This method predicts drought by assessing impact of individual factors on drought, analyze the relationship in them and draws the conclusion regarding the drought. Most of the predictions primarily depends on precipitation rate, and basic parameters like humidity, temperature, ground water level, rainfall, wind speed for predicting drought. This is also a less complex method in predicting drought comparing with existing drought predicting methodologies.

10.3 Calculate Severity Index

In this Module, Calculate the severity index based on drought prediction parameters like humidity, temperature, ground water level, rainfall, wind speed for predicting drought.

10.4 View information Module

In this Module, user can view the information about how many percentage drought will be occurred.

CHAPTER 11

SYSTEM DESIGN AND TESTING PLAN

11.1 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

11.2 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the

- Future.

- Signal important events, opportunities, problems, or warnings.

- Trigger an action.

- Confirm an action.

CHAPTER 12

SYSTEM TESTING

Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

12.1 Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the

specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

CHAPTER 13

UML

DIAGRAM

UML is simply another graphical representation of a common semantic model. UML provides a comprehensive notation for the full lifecycle of object-oriented development.

13.1 ADVANTAGES

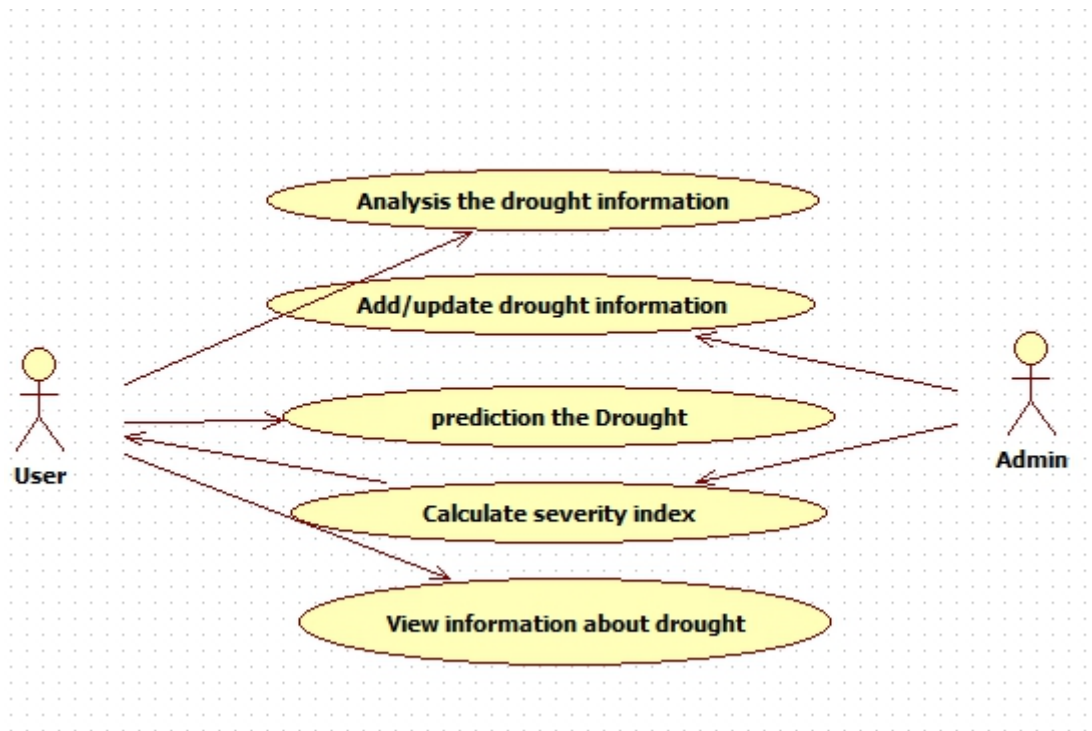
- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To creating a modeling language usable by both humans and machines

UML defines several models for representing systems

- The class model captures the static structure
- The state model expresses the dynamic behavior of objects
- The use case model describes the requirements of the user
- The interaction model represents the scenarios and messages flows
- The implementation model shows the work units
- The deployment model provides details that pertain to process allocation

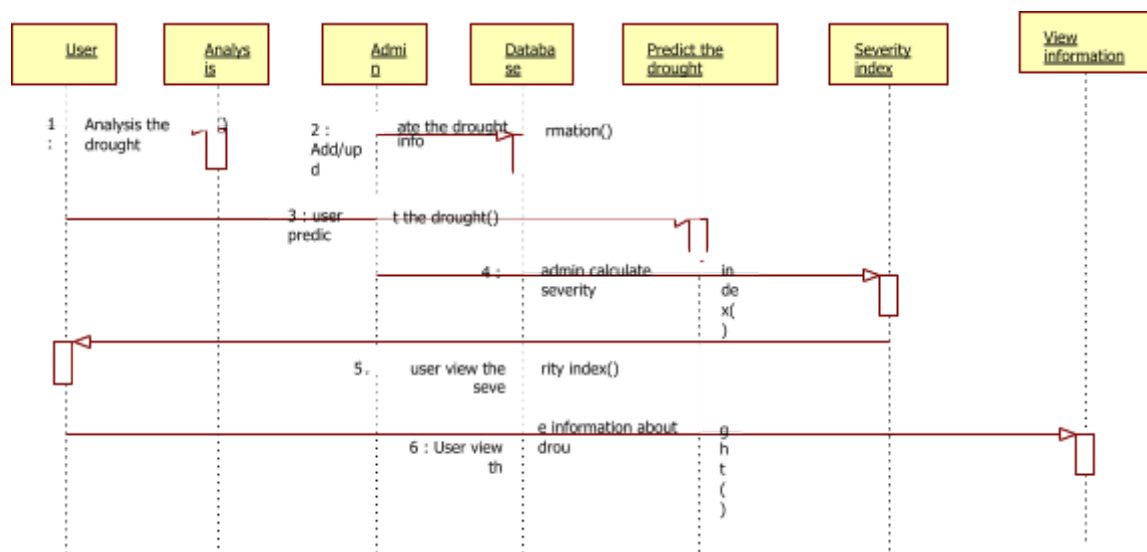
13.2 USECASE DIAGRAM

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse.



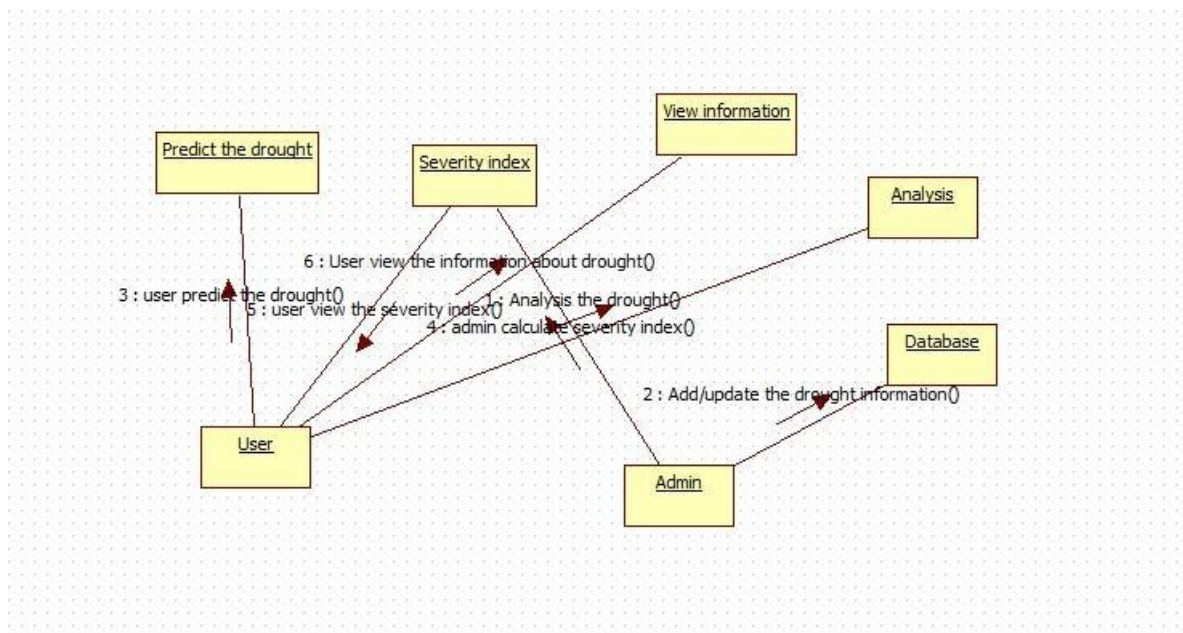
13.3 SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.



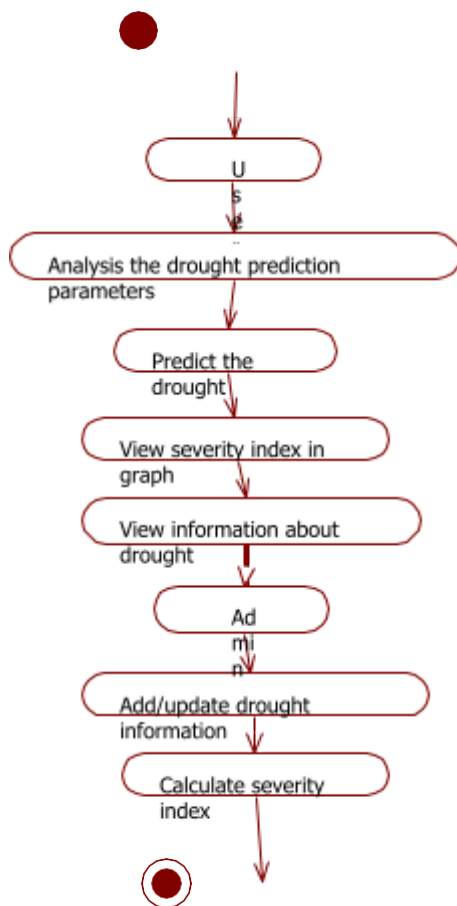
13.4 COLLABORATION DIAGRAM

Another type of interaction diagram is the collaboration diagram. A collaboration diagram represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchange among the objects within the collaboration to achieve a desired outcome.



13.5 ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.

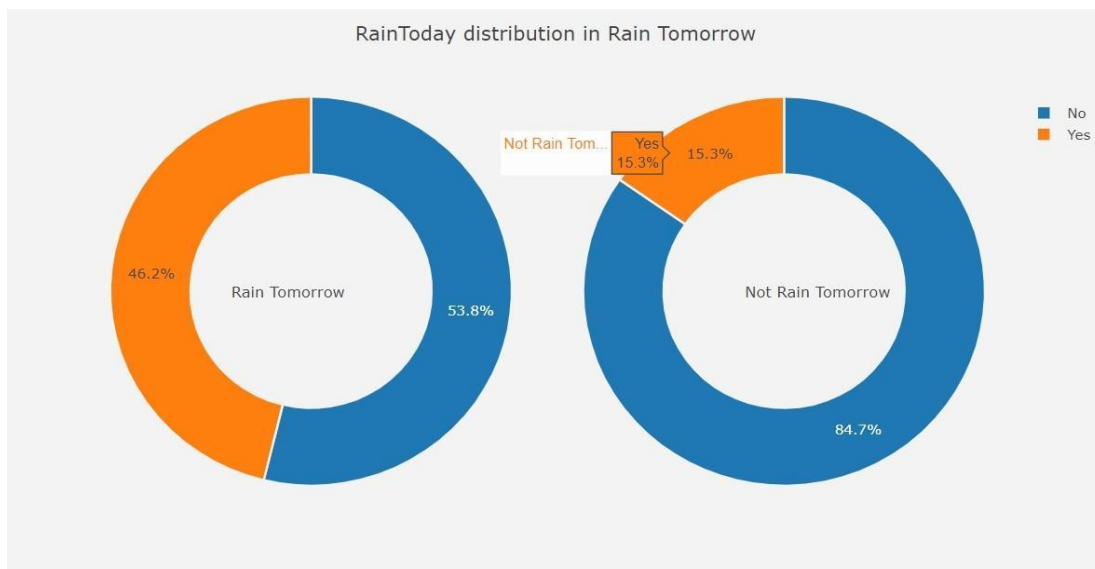


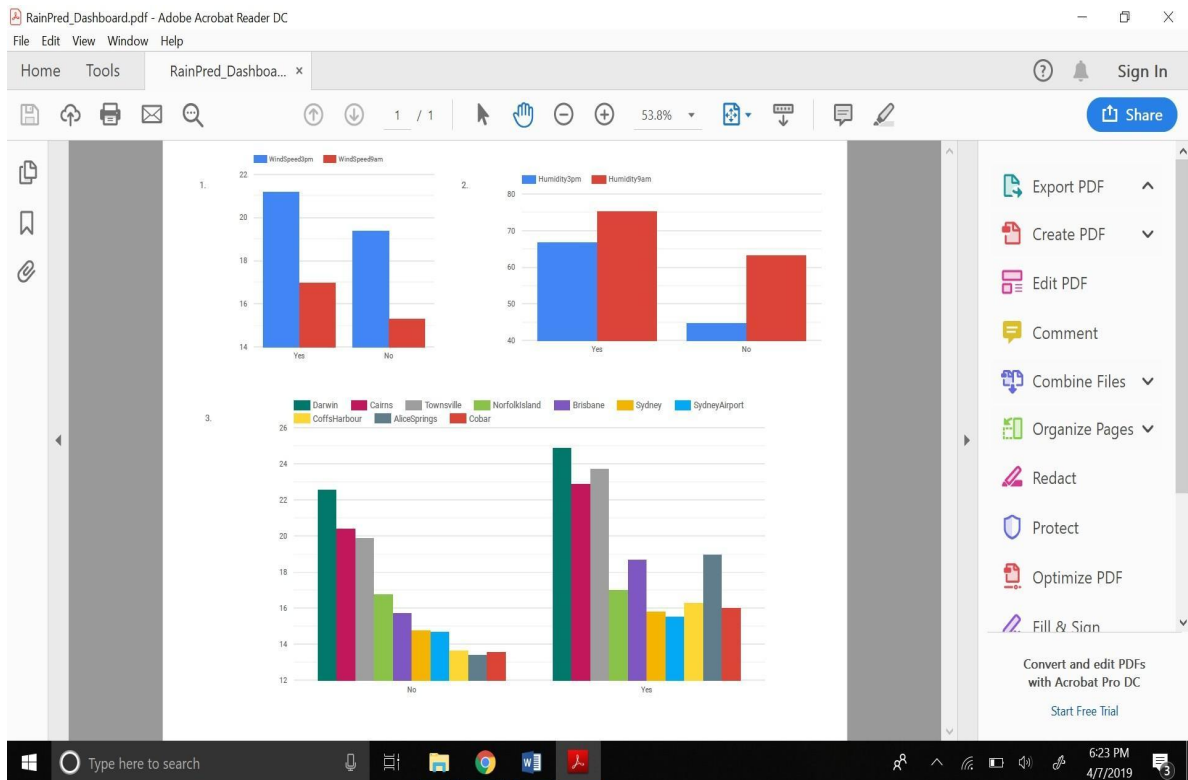
CHAPTER 14

RESULTS

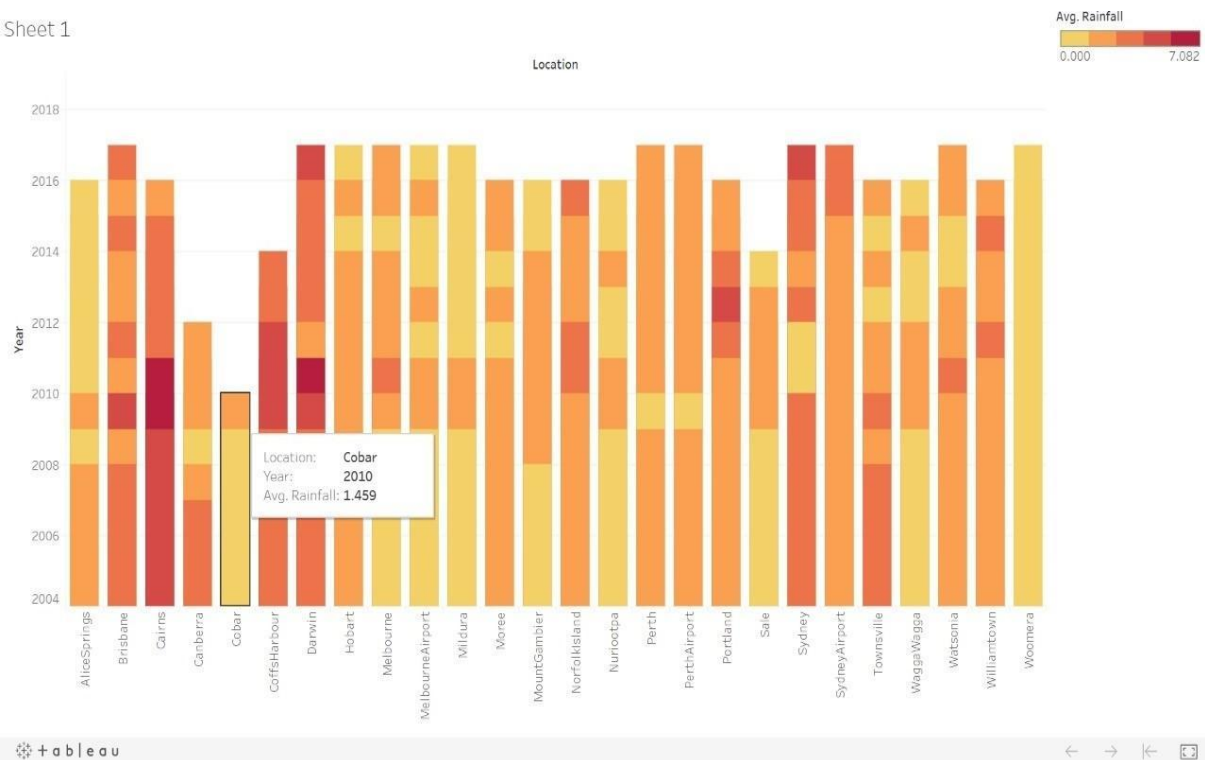
This method predicts drought by assessing impact of individual factors on drought, analyze the relationship in them and draws the conclusion regarding the drought. Analysing and Predicting the drought based on humidity, temperature, ground water level, rainfall, wind speed. To improve drought prediction, efforts are still needed in the understanding of drought mechanisms and predictability in different regions or seasons. Apart from this, several future aspects that may contribute to the improved drought prediction.

OUTPUT





Sheet 1



CHAPTER 15

CONCLUSION

This study reviewed recent developments of seasonal drought prediction methods from statistical and dynamical perspectives. The large scale circulation patterns, land surface interactions, and land initial conditions provide important predictability of drought at the seasonal time scale. Meanwhile, a variety of important contributing factors, including local climate, evaporation changes, and human activities may maintain or amplify different types of drought. Seasonal drought prediction mainly focuses on fully utilizing these predictability sources or contributing factors operating at local, continental, or global scales to facilitate drought prediction at seasonal time scales. For the statistical prediction, these predictability sources and contributing factors are mainly represented as oceanic, atmospheric, or hydrologic predictors and are then fed to a forecast model with the key task to model linear or nonlinear relationships between the predict and and predictors. For the dynamical prediction, GCMs and hydrologic models parameterize physical processes in climatic and hydrologic systems for predicting droughts of different types. Comparison of dynamical and statistical prediction methods reveals the differences in the requirement of computing resources and ability to model complicated interactions. This motivates the integration of both methods to combine drought forecasts from different sources for improving drought prediction.

CHAPTER 16

FUTURE

SCOPE

To improve drought prediction, efforts are still needed in the understanding of drought mechanisms and predictability in different regions or seasons. Apart from this, several future aspects that may contribute to the improved drought prediction.

16.1 REFERENCES

- 1) Short term drought prediction by Anteneh Belanyeh and Jan Malinowski
- 2) Meteorological Drought Prediction Using a Multi-Model Ensemble Approach
- 3) DROUGHT INDICES Andrej Ceglar Biotechnical faculty, University of Ljubljana
- 4) Researchgate.net publication ,238733223 , Methods for predicting drought occurrences
- 5) http://www.dmcsee.org/uploads/file/71_jrc_drought_forecasting_methods_b_kurnik.pdf
- 6) <https://albertawater.com/drought-mitigation/drought-forecasting-and-monitoring>
- 7) <https://drought.unl.edu/Education/DroughtIn-depth/Predicting.aspx>
- 8) <https://www.sciencedirect.com/science/article/pii/S0169809514003858>

9) <https://ieeexplore.ieee.org/document/7925314>

