A PROJECT REPORT ON

APPLICATION CODE ANALYSIS

SUBMITTED TO SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILMENT FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY**

BY

SACHIN SHINDE	**B120058633**
SEJAL JAGTAP	**B120058546**
PIYUSH WANI	**B120058649**
ASHISH MALANI	**B120058576**

UNDER THE GUIDANCE OF

Ms. K. A. SULTANPURE

**DEPARTMENT OF INFORMATION TECHNOLOGY
PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE
SR. NO. 27, PUNE-SATARA ROAD, DHANKAWADI, PUNE – 43**

2016-2017

CERTIFICATE

This is to certify that the project report entitled

Application Code Analysis

Submitted by

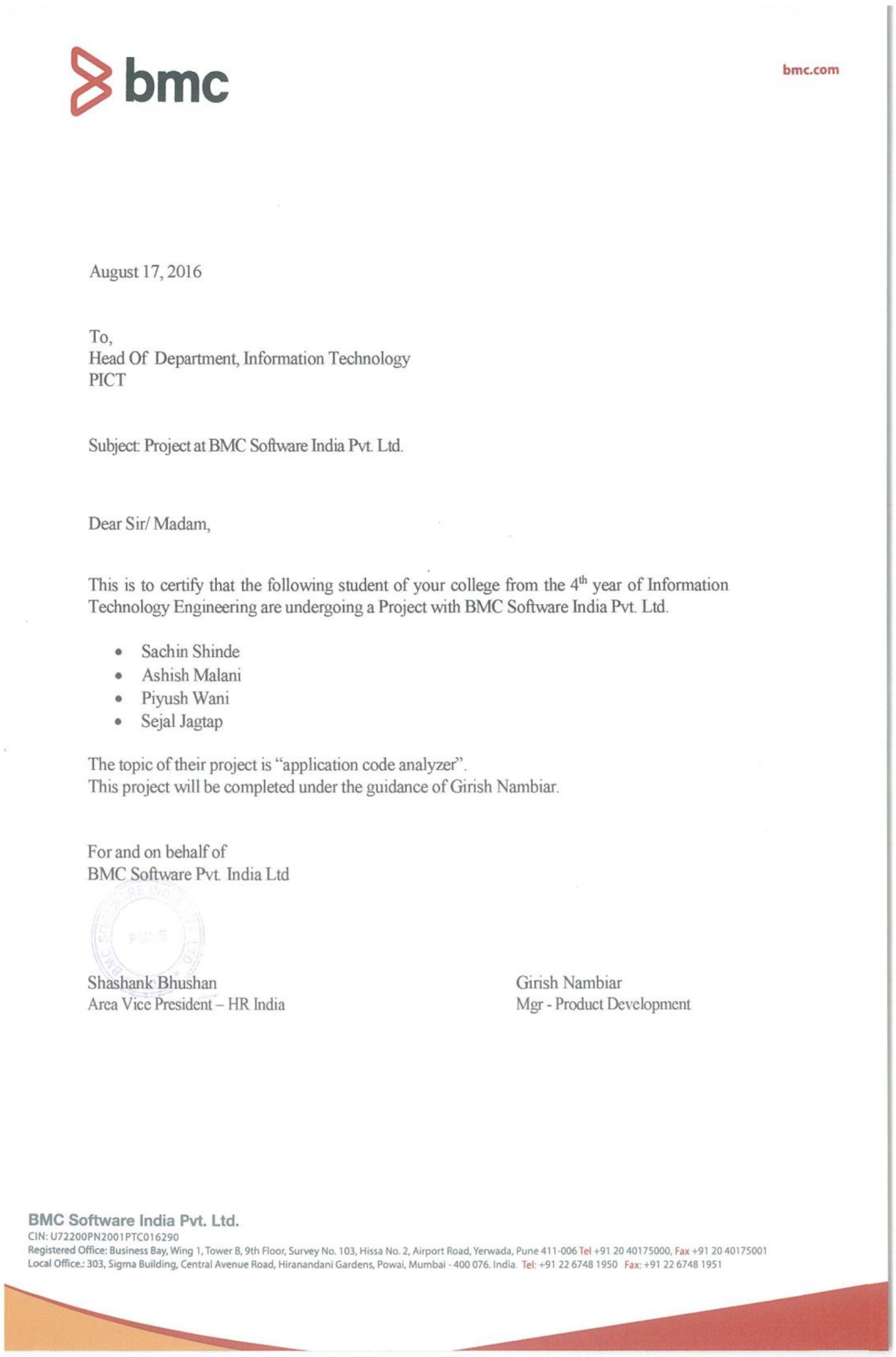
SACHIN SHINDE	**B120058633**
SEJAL JAGTAP	**B120058546**
PIYUSH WANI	**B120058649**
ASHISH MALANI	**B120058576**

is a bonafide work carried out by them under the supervision of Prof. S. B. Deshmukh and it is approved for the partial fulfilment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology)

This project report has not been submitted earlier to any other Institute or University for the award of any degree or diploma.

Ms. K.A.Sultanpure Internal Guide Department of Information Technology	Prof. S.B.Deshmukh Project Coordinator Department of Information Technology
Head of Department Department of Information Technology	Dr. P. T. Kulkarni Principal Pune Institute of Computer Technology, Pune
External Examiner	Place: Date:

SPONSORSHIP LETTER



bmc

bmc.com

August 17, 2016

To,
Head Of Department, Information Technology
PICT

Subject: Project at BMC Software India Pvt. Ltd.

Dear Sir/ Madam,

This is to certify that the following student of your college from the 4th year of Information Technology Engineering are undergoing a Project with BMC Software India Pvt. Ltd.

- Sachin Shinde
- Ashish Malani
- Piyush Wani
- Sejal Jagtap

The topic of their project is "application code analyzer".
This project will be completed under the guidance of Girish Nambiar.

For and on behalf of
BMC Software Pvt. India Ltd

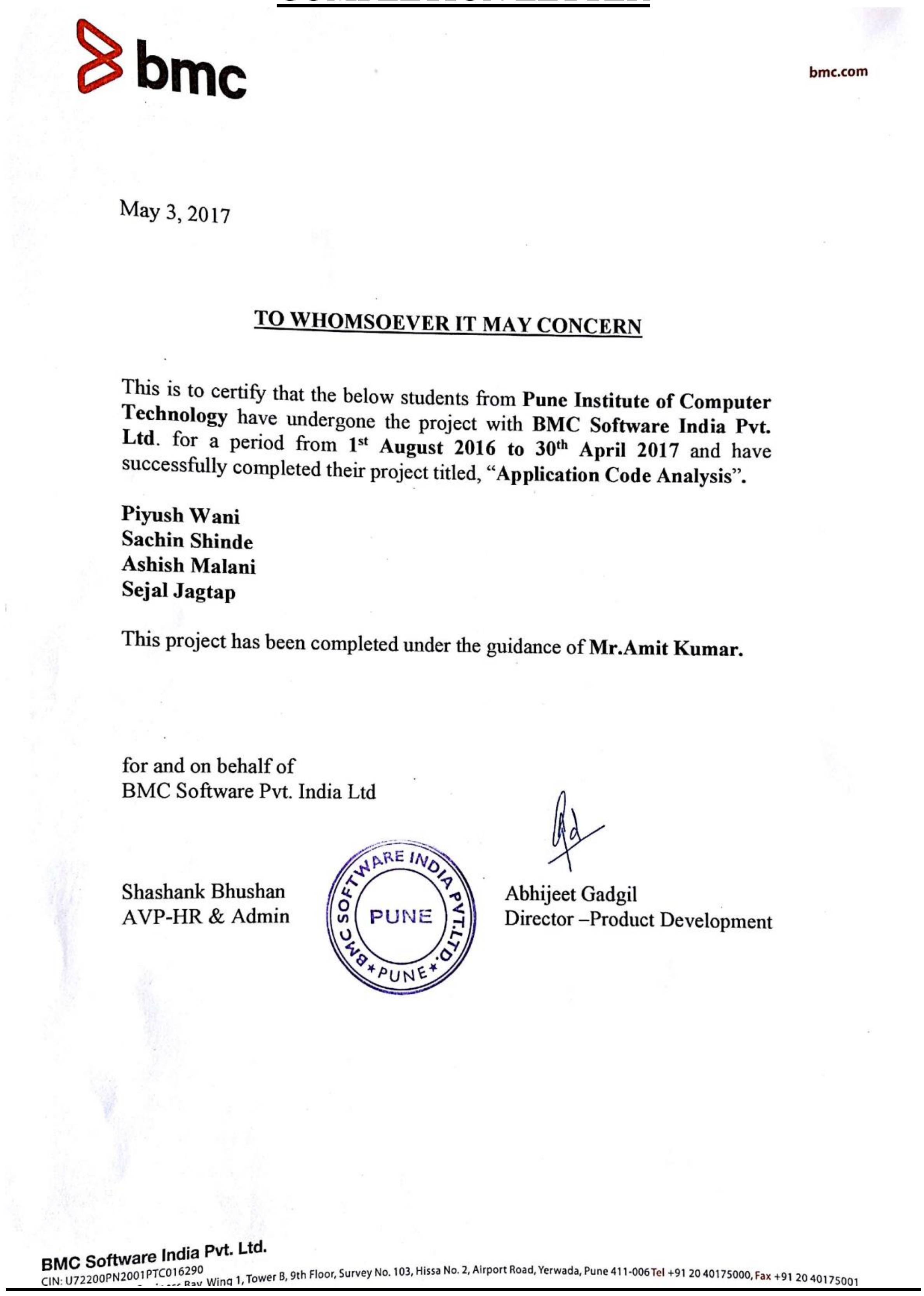
BMC SOFTWARE INDIA PVT. LTD. PUNE

Shashank Bhushan
Area Vice President – HR India

Girish Nambiar
Mgr - Product Development

BMC Software India Pvt. Ltd.
CIN: U72200PN2001PTC016290
Registered Office: Business Bay, Wing 1, Tower B, 9th Floor, Survey No. 103, Hissa No. 2, Airport Road, Yerwada, Pune 411-006 Tel +91 20 40175000, Fax +91 20 40175001
Local Office.: 303, Sigma Building, Central Avenue Road, Hiranandani Gardens, Powai, Mumbai - 400 076. India. Tel: +91 22 6748 1950 Fax: +91 22 6748 1951

COMPLETION LETTER



bmc

bmc.com

May 3, 2017

TO WHOMSOEVER IT MAY CONCERN

This is to certify that the below students from **Pune Institute of Computer Technology** have undergone the project with **BMC Software India Pvt. Ltd**. for a period from **1st August 2016 to 30th April 2017** and have successfully completed their project titled, "**Application Code Analysis**".

Piyush Wani
Sachin Shinde
Ashish Malani
Sejal Jagtap

This project has been completed under the guidance of **Mr.Amit Kumar.**

for and on behalf of
BMC Software Pvt. India Ltd

[illegible]

BMC SOFTWARE INDIA PVT.LTD. PUNE *PUNE*

Shashank Bhushan
AVP-HR & Admin

Abhijeet Gadgil
Director –Product Development

BMC Software India Pvt. Ltd.
CIN: U72200PN2001PTC016290
Wing 1, Tower B, 9th Floor, Survey No. 103, Hissa No. 2, Airport Road, Yerwada, Pune 411-006 Tel +91 20 40175000, Fax +91 20 40175001

ACKNOWLEDGEMENT

We would like to thank our guide, Ms. K.A. Sultanpure for her constant support and encouragement during the course of this project.

We would also like to acknowledge with much appreciation to our project coordinator Mr. S. B. Deshmukh.

We would also like to thank the in charge HOD-IT, Dr. A. M. Bagade, for always being co-operative and helpful.

We also thank the principal, Dr. P. T. Kulkarni, for his support without which it would not have been possible to complete this project.

We would also like to thank our previous project guide Dr. S.C. Dharmadhikari for her help in initial stage of project.

We thank the management of our college for providing us excellent infrastructure and all the facilities required for the project.

We are grateful to the lab attendants for their help in setting up and using the computers in the department laboratories.

Lastly, we thank our friends and family for their support and encouragement, which propelled us to the completion of this project.

Sachin Shinde (B120058633)
Sejal Jagtap (B120058546)
Piyush Wani (B120058649)
Ashish Malani (B120058576)

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Description**
SAST	Static Application Security Testing
SDL	Software Development lifecycle
SQL	Software Quality Objectives
ACR	Automated Code Review
CLI	Command Line Interface
AWT	Abstract Window Toolkit
JOB	Java Objects for Business

CONTENTS

ABSTRACT

The proposed application will enable systematic aggregation of rules, and enable users to choose what they want, when they want. This application will display static errors for a given application package. The framework will analyze programs by carrying out static code analysis on the source code.

The main benefits lie in improving software quality by detecting potential defects and problematic code constructs in early development stages. The framework provides features for extracting and processing textual patterns found in native JavaScript and other javascript frameworks.

Keywords : Aggregation, Framework, Textual pattern, Static code analysis

CHAPTER 1

INTRODUCTION

Code analysis is a method of an automated testing of any programming language source code for the purpose of debugging or testing the program before using it in deployment phase. The source code can be dynamic or static. It is comparatively easy to do static source code analysis. In this part we are focusing on static source code analysis.

1.1 Relevance:

- Most of the developers face problem while developing application based on javascript framework. As it is an interpreted language, it is quite impossible to index everything and apply analysis on it.
- So companies require a generic framework that can provide ability to analyze any javascript source code framework along with flexibility.

1.2 Project Undertaken:

1.2.1 Project Idea

A application package is nothing but set of files which are brought together and linked to each other to carry out particular task. Application package can be web application and a software. Application package can be written in various languages like Java, C, C++, Python, Ruby etc.

1.2.2 Need of the project

BMC Application packages created by Innovation Studio which are deployed on BMC's AR servers are not verified and vulnerable to run time errors also the development style of developers varies at its a big corporation. When one wants to integrate the module to specific application package, it can be hectic task if there is no uniformity in the code.

Static code analysis techniques can be used to solve this problem. Our analyzer tool will actually set the rules which should be followed at the time of development and some optional rules which will not harm uniformity of the code. This rules will be checked at the time of development and deployment using static code analyzer and deployment decisions will be made according to that.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 Background

Although there are lots of frameworks available for javascript code analysis. There are some problems that still exists with them.

- Single Runtime
- Disabling Rules
- Documentation
- Configuring rules

2.2 Existing Methodology

2.2.1 Cross language Code analysis and Refactoring[3]

Software composed of packages are normally written in multiple programming languages. These types of packages are suitable for today's enterprise, desktop and mobile applications. Those types of packages usually forms united system with different languages referencing each other. It creates semantic cross-language links. Because of their structure such links are usually difficult to identify and such links are usually ignored by language-specific tools. Those types of errors are checked only at runtime.

This is very time consuming task since it requires additional testing, leads to unmaintained code. So, code analysis of such packages require understanding, analysis and refactoring of cross-language code. It also requires explicitly specifying and exploiting sematic links. It gives developers control over multi language programming as if they are developing it in single language. [1]

2.2.2 Multilingual source code analysis [4]

The importance of using multi-language software systems in a project is continuously growing. Increase in demand of multiple language systems caused increase in need to develop multilingual source code analyzers. The use of multiple language tools causes generation of complex systems which are usually difficult to analyze.

Getting output of source code wrong on deployed systems cause project failure .The main strategy to carry out additional assurance analysis is static code analysis. It also consists of identifying current software version as well as future updates. This kind of source code analysis mainly focuses on quality linking between multiple languages. [2]

The current solutions currently suffers from certain limitations, some of which are:

- The frameworks are not generic. Those frameworks only focuses on identifying core problem of a single-language.
- The frameworks are not extensible. They cannot be used as an extension to newly developed frameworks.
- Most of them are hard-coded. They do not provide flexibility to programmer to apply required constraints.
- Writing custom logic is difficult. It's difficult to integrate custom logic to those frameworks.

2.3 Proposed Methodology

The project would develop a system which will able to sustain change in code as well as will fulfill requirements mentioned by developer at the time of deployment .By doing so, it will enable application developers to maintain clean and refactored code on cloud servers. It will also reduce chances of runtime errors in deployed package.

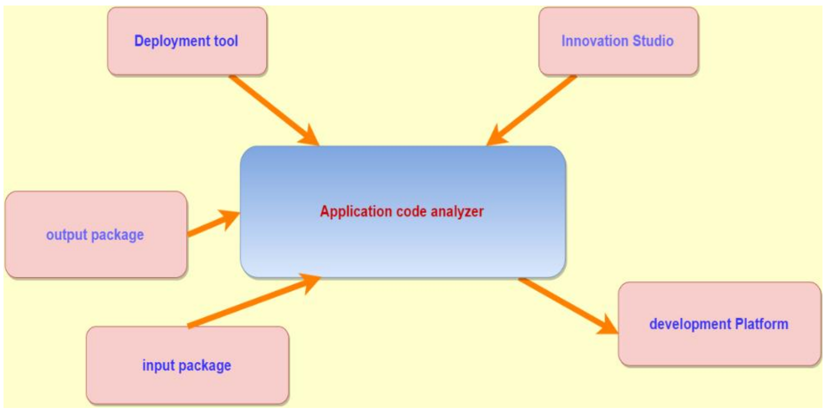


Fig. 2.1 System Interpretation

The framework will meet following conditions based on the requirements.

- The framework is flexible and extensible and will check whether all standards are met for an app before deployment.
- App developer can extend the framework to add new rules on top of existing framework.
- The framework will generate a report problem wise so that it can be corrected by the app developer. The purpose of this framework is also to analyze errors in code which doesn’t satisfy standards, so that it can handle and correct them if possible.
- The framework will do deployment time validation of rules and constraints corresponding to package version and will make the deployment decision.

CHAPTER 3

REQUIREMENT SPECIFICATION AND ANALYSIS

3.1 Problem Definition

To create a flexible framework to analyze and check all standards of given package with respect to some basic set of rules given as arguments and also upgrade it to detect and handle errors. The framework enables users to add new rules on top of existing framework. In short, the framework is reusable and generic. The framework will generate a report problem wise so that it can be corrected by the app developer. The purpose of this framework is also to analyze the errors in code which doesn't satisfy standards, so that it can handle them.

Static code analysis technique usually analyze programs by examining the source code without actually executing them. The main benefit is developer gets set of errors generated in early development stages which contribute to improving more software quality. There is also another aspect called software security which comes into picture when developer need to deploy highly confidential application on required server. So, this code analyzer can also be used to detect security vulnerabilities during development phase.

The current problem is to develop a framework in java and python for analyzing javascript code. The javascript analysis supports analysis of pure javascript, common-script, typescript, Angular.js and stylistic javascript.

The framework is developed on the basis of javascript linting modules. Those linting modules supports code analysis by using set of defined rules. It requires a lot efforts to set up everything, so real-deal is to make it more user-friendly and understandable.

3.2 Statement of Scope

- Understand the innovation studio framework
- Choose appropriate javascript linting framework and writing down code that will analyse the Javascript souce code.
- Develop Custom Rule generator tool and tool for Editing set of rules available in Rule file. Develop sublime plugin with linting as a basic framwwork.
- Integration of Rule generator tool, Rule editor and sublime plugin.

3.3 Hardware and system requirements

Hardware:

- Intel i3 Processor or greater
- Ram 2 GB or greater

Software:

- Python >= 2.7
- Java SE 8
- Sublime >=2
- Node.js >= 5.2.3

CHAPTER 4

DESIGN

4.1 Introduction

4.1.1 Goals

- To create a flexible/extensible framework to analyse and check whether all standards are met for an app before deployment.
- App developer can extend the framework to add new rules on top of existing framework. In short, the framework is reusable and generic.

4.1.2 Objectives

- The framework/tool will do the code analysis in order to verify the standards and any errors occurring at run time and decide about the app deployment.
- The framework will generate a report problem wise so that it can be corrected by the app developer. The purpose of this framework is also to analyse the errors in code which doesn’t satisfy standards, so that it can handle them and correct them if possible.

4.1.3 Statement of scope

The application code analysis will cover a wide range of features. The scope of the project consists of the following details:

- Rule generator file to add the rules (constraints).
- Checking/Unchecking of the optional rules.
- Static code analysis of the source to validate the standards according to constraint.
- Generation of errors if the constraints are not met.
- Showing suggestions to the developer if the errors are present.
- Validating the code standards and making deployment decisions according to that.

4.1.4 Purpose

- To main purpose is to provide a uniqueness in modularity, quality and standard of each application. It makes simple for IT companies to identify where improvements can be made to boost effectiveness or performance and identify undetected problems before they bring down server resources or create potential harm to security of organization.
- The secondary purpose is to provide flexibility of choice with set of rules the developer wishes to apply for code analysis of a particular package.
- To overcome problems faced in Traditional frameworks which cannot effectively detect vulnerable or defective code within a multi-tiered infrastructure.

4.2 Usage scenario

4.2.1 User profiles

1. Developer
Developers will use the innovation studio to create the web application.

2. User
Users will use the web app that is deployed on Innovation studio

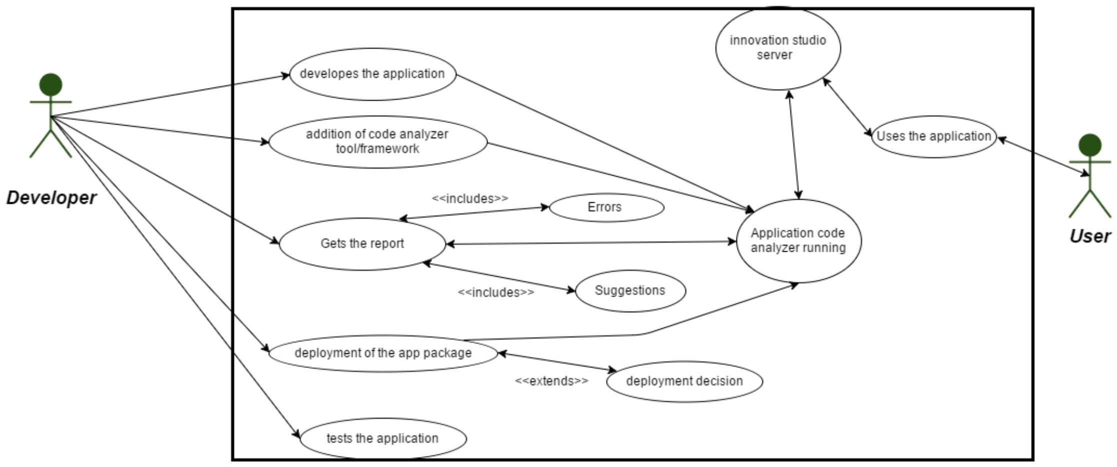


Fig. 4.1 Use case model

Table 4.1 Use case description

Use-cases	**Use case description**
Develops the application	Developer write the code for the application package to develop an innovation studio app
Adds the App Code Analyzer	Developer adds app code analyzer tool along with application package to set the rules and constraints.
Runs the application package	Developer runs the application at the time of development
Test the app	Testing of the application is done by tester using some test cases to check the proper working and app code analyzer to check the package's standard.
Generate the test results	Test results are done to check the working of application and to identify errors in the functionality
Gets the errors and suggestions	App will go through static code analysis and will be given some output to the developer if there is any errors regarding rules and constraints decided while adding app code analyzer
Deployment of the app package	Developer deploys the package on the server
Application code analyzer	Application code analyzer runs implicitly though the code whenever package is tested, Changed and deployed.
Application use	User uses the application deployed on Innovation studio

4.2.2 Use Case Specification

Name: Adds the App Code Analyzer

Description: Developer adds app code analyzer tool along with application package to set the rules and constraints

Primary actors: Developer.

Preconditions:

1. The input package should be ready to be tested on franework.
2. The framework must contain all the APIs to evaluate the package.

Basic Flow:

1. Developer adds the rule generator file via commandline instructions.
2. Rule.pro file will be generated.
3. Default rules will be displayed in file but can’t be edited .
4. User can add more set of rules in order to meet the requirements.

4.3 Data Model and Description

4.3.1 Data Model

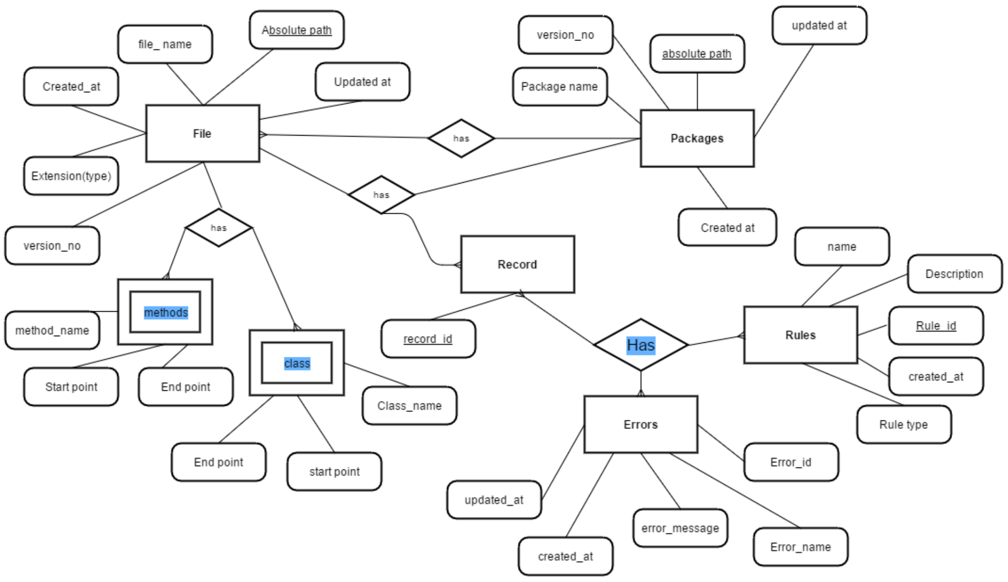


Fig.4.2 Data Model (ER diagram)

Data model can be represented in the form of ER-diagram. Entities can be shown with their attributes and related to each other in data model. Above diagram shows generalized data model of the framework.

4.3.2 Data Dictionary

4.3.2.1 Package Data Dictionary:

Table 4.2 Package Data Dictionary

P/F	**Field Name**	**Caption**	**Data Type**	**Field size**	**Description**
P	Absolute_path	Path	Varchar	50	Absolute path of package
	Package name	P_name	Varchar	30	Name of the application package
	Version_no	Version_no	Integer		Version no of the package
	Created_at	Created_at	DATETIME		Timestamp of the package creation
	Updated_at	Updated_at	DATETIME		Timestamp of the package updation

4.3.2.2 File Data Dictionary

Table 4.3 File Data Dictionary

P/F	Field Name	Caption	Data Type	Field size	Description
	File_name	Fname	Varchar	50	Name of file
P	Absolute_path	A_path	Varchar	50	Absolute path for file
	Extension	F_type	Varchar	20	File type
	Version_no	Version_no	Integer		Version no of the package
	Created_at	Created_at	DATETIME		Timestamp of the file creation
	Updated_at	Updated_at	DATETIME		Timestamp of the file update

4.3.2.3 Rules Data Dictionary

Table 4.4 Rules Data Dictionary

P/F	Field Name	Caption	Data Type	Field size	Description
P	Rule_id	R_id	Integer		Rule ID
	Rule name	R_name	Varchar	30	Rule name
	Description	R_Desc	Varchar	100	Rule description
	Rule type	R_type	Boolean		Rule type optional or default
	Created_at	Created_at	DATETIME		Timestamp of the rule creation

4.3.2.4 Errors Data Dictionary

Table 4.5 Error Data Dictionary

P/F	Field Name	Caption	Data Type	Field size	Description
P	Error_id	E_id	Integer		Error ID
	Error name	E_name	Varchar	30	Error name
	Error message	E_message	Varchar	100	Error message (cause of error and suggestion)
	Created_at	Created_at	DATETIME		Timestamp of the error creation
	Updated_at	Updated_at	DATETIME		

4.3.2.5 Methods Data Dictionary

Table 4.6 Methods Data Dictionary

P/F	Field Name	Caption	Data Type	Field size	Description
F	File_absolute_path	F_path	Varchar	50	File it is present in
	Method name	M_name	Varchar	50	Method name
	Start point	M_start_point	Integer		Line no. in the file where method starts
	End point	M_end_point	Integer		Line no. in the file where method ends

4.3.2.6 Class Data Dictionary

Table 4.7 Classes Data Dictionary

P/F	Field Name	Caption	Data Type	Field size	Description
F	File_absolute_path	F_path	Varchar	50	File it is present in
	Class name	C_name	Varchar	50	Method name
	Start point	C_start_point	Integer		Line no. in the file where Class starts
	End point	C_end_point	Integer		Line no. in the file where class ends

4.3.2.7 Records Data Dictionary

Table 4.8 Records Data Dictionary

P/F	Field Name	Caption	Data Type	Field size	Description
F	File_absolute_path	F_path	Varchar	50	File it is present in
F	Package_absolute_path	F_path	Varchar	50	File it is present in
	Record id	R_id	Integer		Record id
	Rule id	Rule_id	Integer		Rule id
	Error id	E_id	Integer		Error id

4.4 Functional Model and Description

4.4.1 Description

- **Java/ JavaScript package generation:**
  - For writing custom code on server, developer needs to create a custom java or JavaScript package. This package can contain multi language references which needs to handled by static code analyzer tool.

- **Package + rules file generation**:
  - Developer defines certain rules that needs to be applied on input package. Code analyzer identifies those rules from rules file and apply it accordingly.

- **Report generation**:
  - The code analyzer tool will perform code analysis and will generate report in accordance with rules defines in rules file.

- **Output package generation:**
  - The final output package is generated after code analysis. It contains clean and optimized code.

- **Package validation and requirement:**
  - After final output package is generated, its time to deploy it on AR server. The output package is validated with respect to its version.

- **Testing:**
  - The package deployed on server is tested by tester in order to verify whether it meets all of functional requirements or not.

4.4.2 Function flow diagram

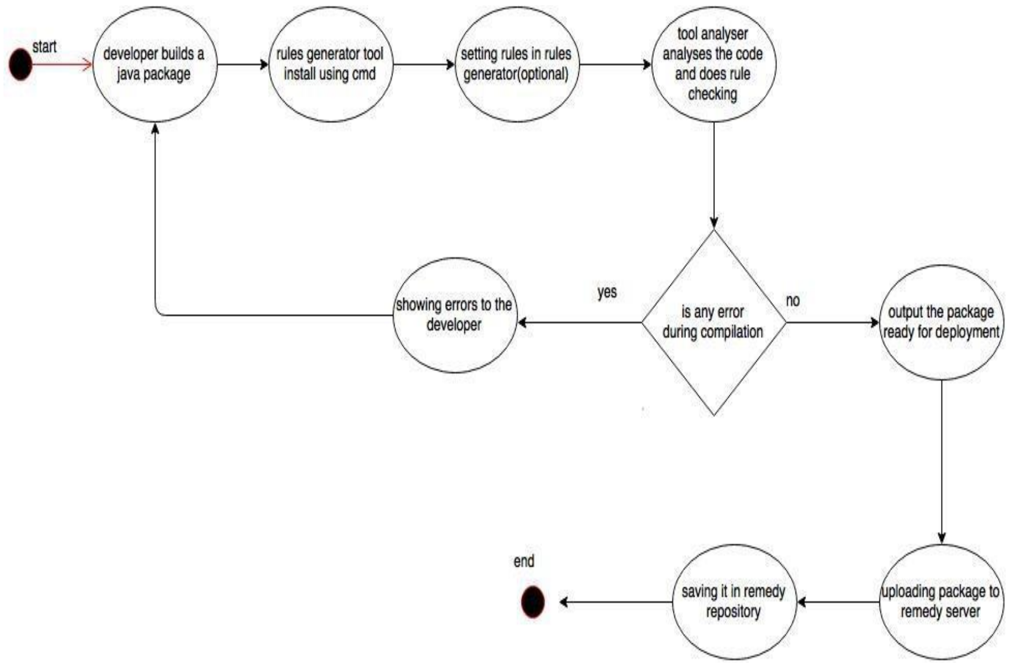

start
developer builds a java package
rules generator tool install using cmd
setting rules in rules generator(optional)
tool analyser analyses the code and does rule checking
is any error during compilation
yes
no
showing errors to the developer
output the package ready for deployment
uploading package to remedy server
saving it in remedy repository
end

Fig 4.3 Function flow diagram

4.4.3 Performance Issues

The proposed system that we are going to develop will be used as framework for testing innovation studio apps before deployment.

- The performance of framework must be accurate, fast and reliable.
- The framework must be able to handle multiple packages at the same time and it should be able to handle multi language input.

4.4.4 Design Constraints

This program is created using java programming language and RX-SDK provided by BMC. So minimum PC having at least 2GB of RAM and CPU over 1.7GHz is required to run the program with good speed. Also the program use 15 megabytes of hard disk space to store program libraries.

4.4.5 External system interfaces

Table 4.9 Software description table

Software used	Description
Operating System	We have chosen Windows 7 as it is provided by BMC
Java	To implement the project we have chosen JAVA language because it is object oriented and more flexible to use.

4.4.6 Control flow description

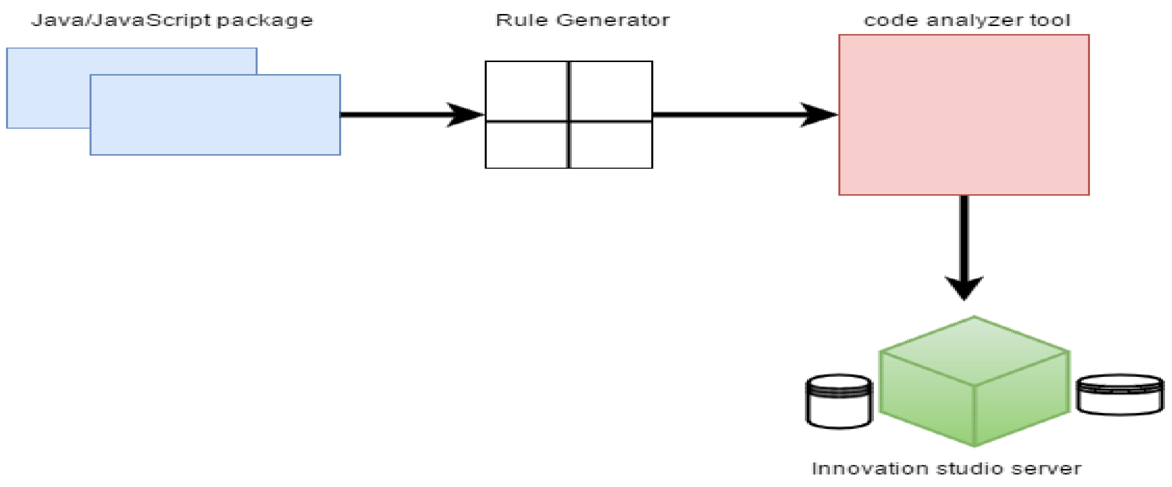


Fig 4.4 Control flow description

4.5 Behavioral model and Description

4.5.1 Description for software behavior

States of our software system are as follow:

1. Java / Javascript package generated

2. Rules + package generated

3. Report generated

4. Report and error validated

5. Package deployed on innovation studio server

4.5.2 State Transition Diagram

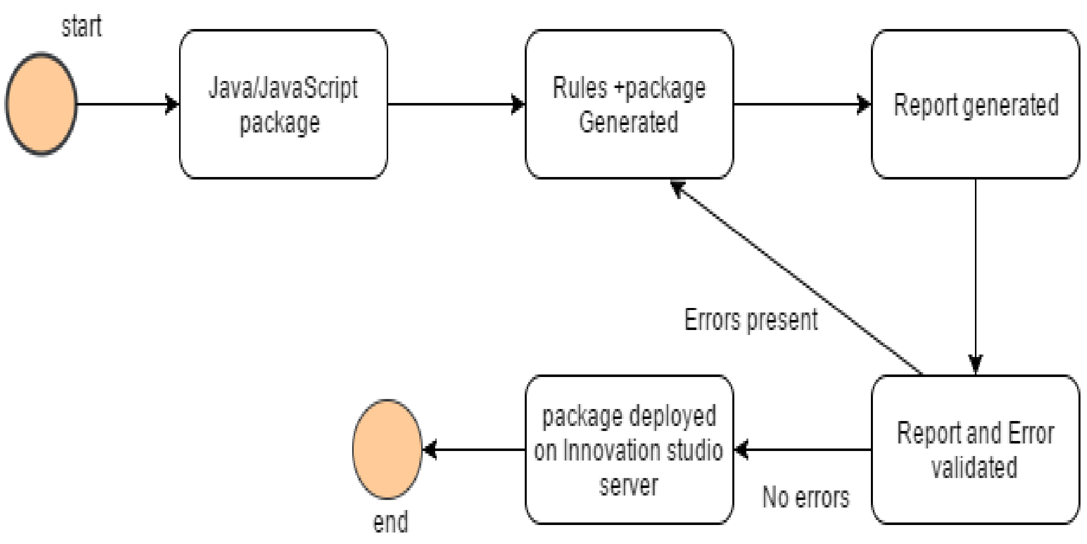


Fig 4.5. State transition diagram

CHAPTER 5

IMPLEMENTATION

5.1 High Level Design of Application Code Analysis

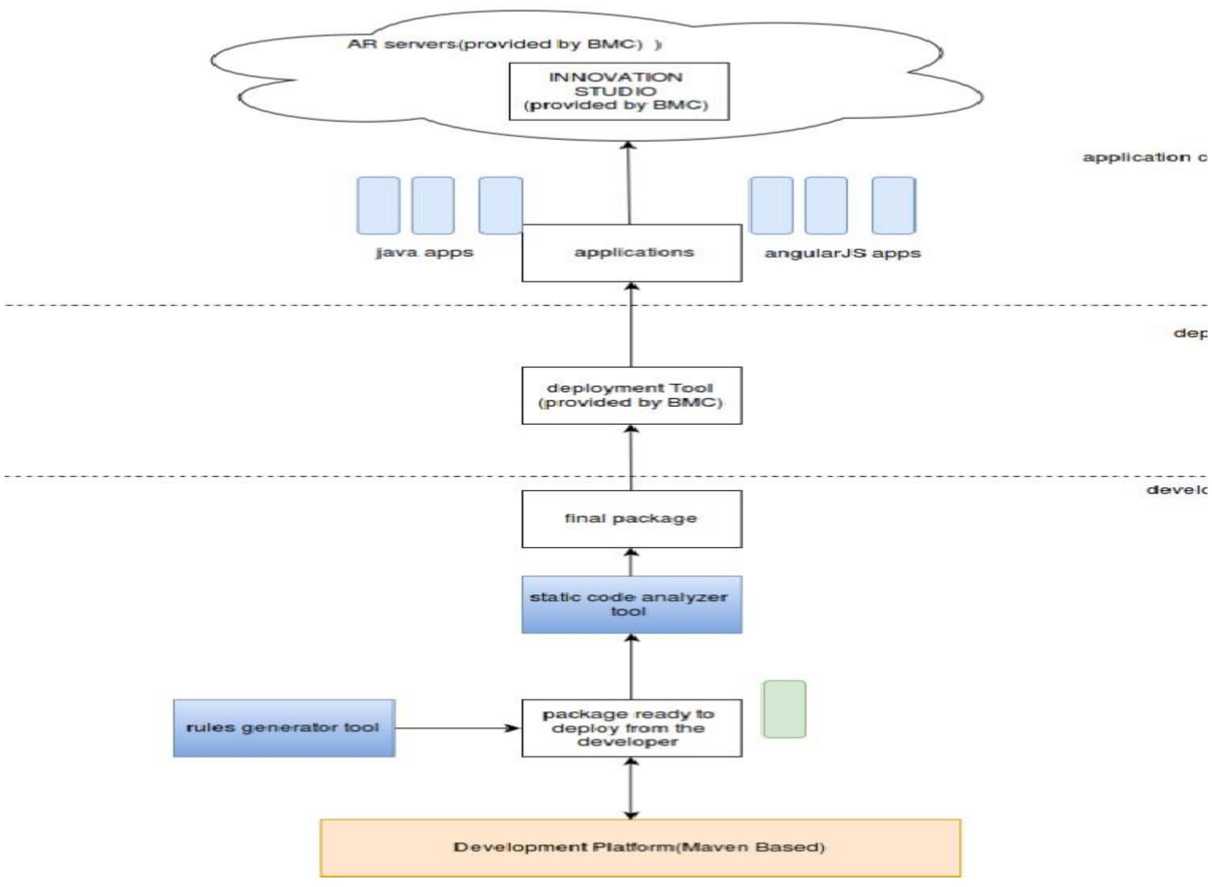

AR servers(provided by BMC)
INNOVATION STUDIO (provided by BMC)
application c
java apps
applications
angularJS apps
dep
deployment Tool (provided by BMC)
develc
final package
static code analyzer tool
rules generator tool
package ready to deploy from the developer
Development Platform(Maven Based)

Fig. 5.1: High level design of the Application code analysis

5.2 Methodology:

The methodology that this project requires is the Waterfall Methodology.Waterfall is a linear approach to software development. In this methodology, the sequence of events is:

1. Gather and document requirements
2. Design
3. Code and unit test
4. Perform system testing
5. Perform user acceptance testing (UAT)
6. Fix any issues
7. Deliver the finished product

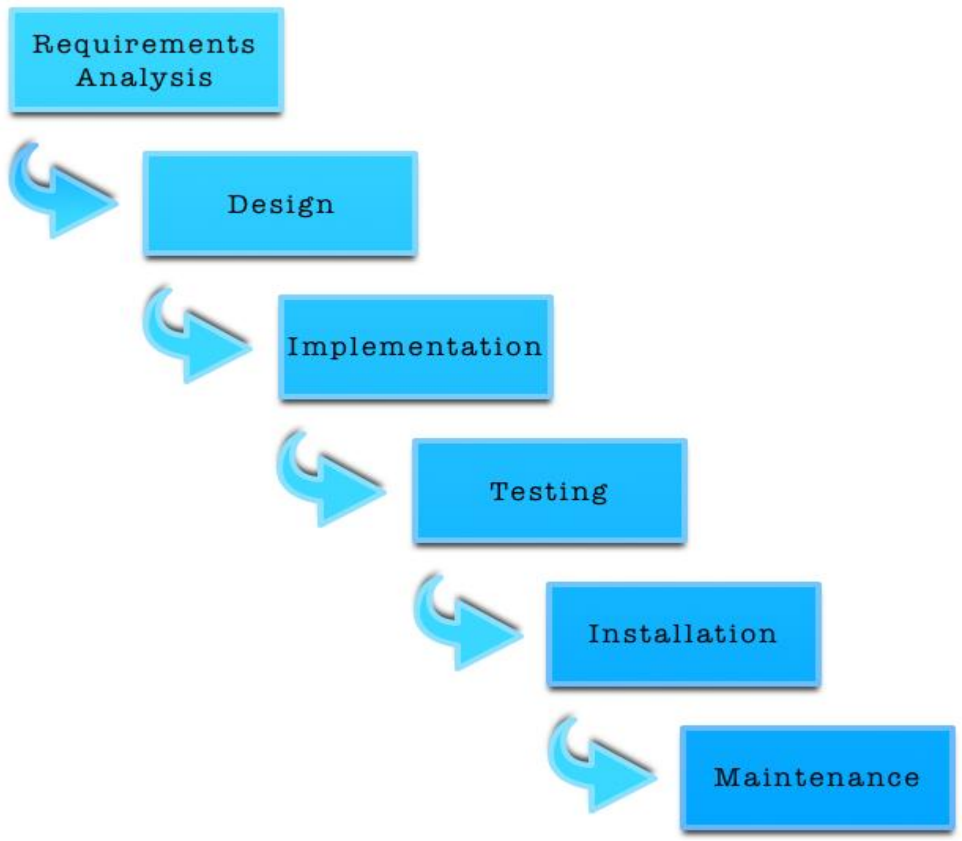


Fig 5.2 Waterfall model

5.3 Module Based Flow Diagrams

1. Rule generator module –

   This module is nothing but a script that defines set of standard rules as well as exclusions that can be applied on input package. The packages are divided into two different types depending on the source codes. Depending on java and javascript package, the rules script will generate set of rules in the form of rules.pro file. This file is embedded inside package so that both package and script can be given as input to application code analyzer.

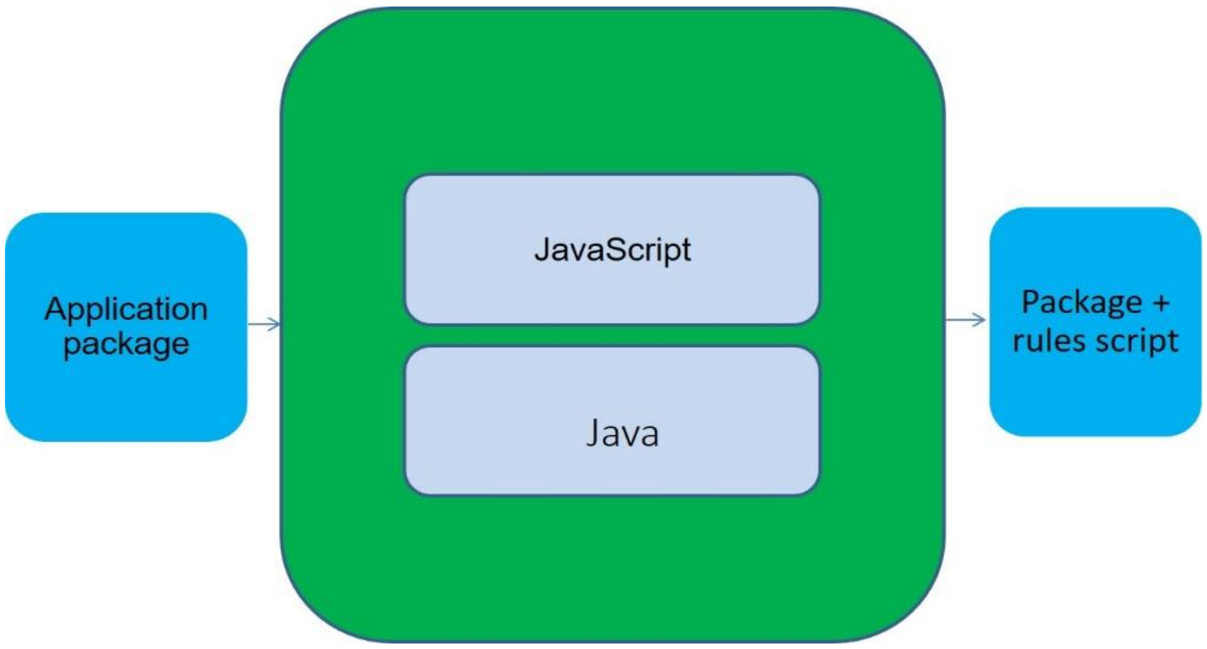


Fig 5.3 Rule Generator flow diagram

2. Code analysis module –

   The output package generated by rule generator module is given as input to code analysis modules. The code analysis modules reads the rules defined in rules.pro file. If any exclusions are found, then it will avoid to run rules on that particular set of files. This module consists of different components. The functionality provided by module is rules validation coded lines, blank lines and comment section recognition, identification of unused and deprecated code. This module will generate reports accordingly.

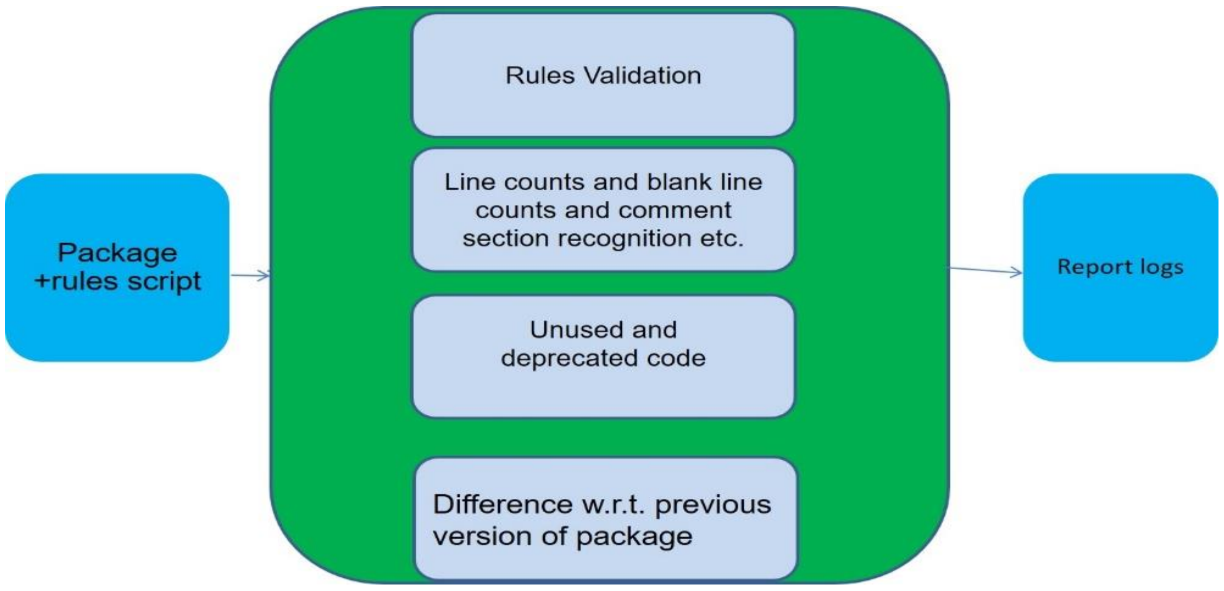

Package
+rules script
Rules Validation
Line counts and blank line counts and comment section recognition etc.
Unused and deprecated code
Difference w.r.t. previous version of package
Report logs

Fig 5.4 Code analysis flow diagram

3. Code suggestion module –

The errors generated can be reported by code analysis module. But, there are some sets of errors which can be corrected by module. The code suggestion module does the same thing. It will show the suggestions that needs to be corrected if possible. The final corrected package will be generated by code suggestion module

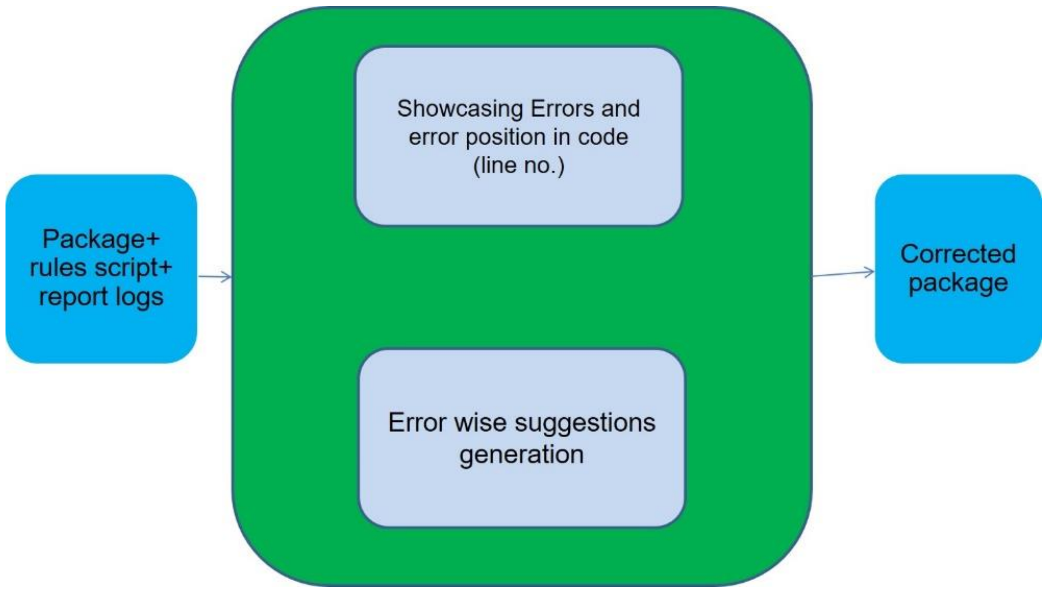

Package+
rules script+
report logs
Showcasing Errors and error position in code (line no.)
Error wise suggestions generation
Corrected package

Fig.5.5 Code Suggestions flow diagram

4. Package validation module –

   The package is now ready for deployment. The final check is identifying version of package that needs to be deployed on innovation server. This module also takes responsibility of reliable uploading of package on the server, so that even if package uploading fails at certain point, it can be resumed in any situation.

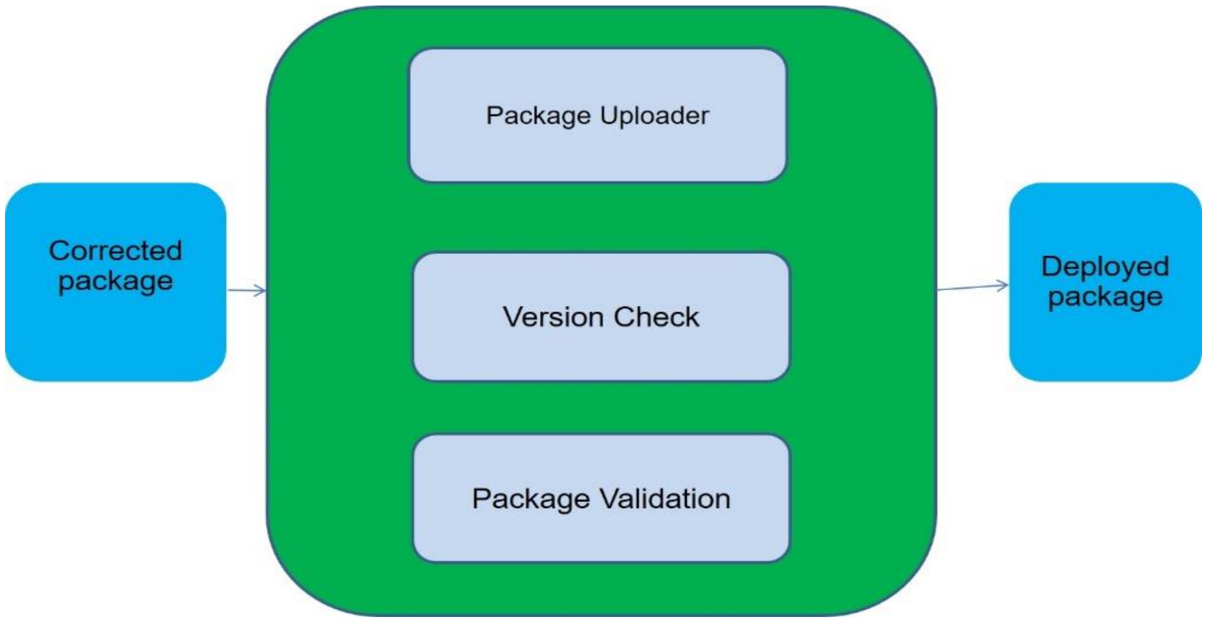


Fig.5.6 Package Validations flow diagram

5.4 Module Based Split Up

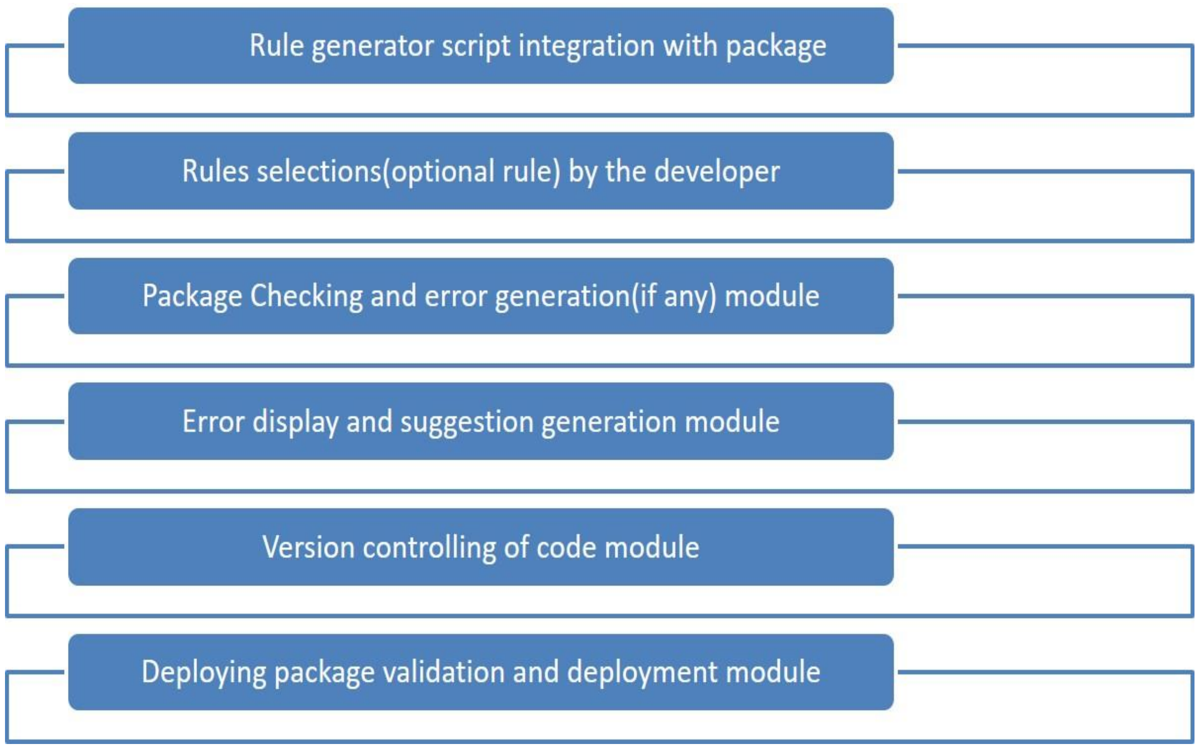

Rule generator script integration with package
Rules selections(optional rule) by the developer
Package Checking and error generation(if any) module
Error display and suggestion generation module
Version controlling of code module
Deploying package validation and deployment module

Fig 5.7: Module Split-up

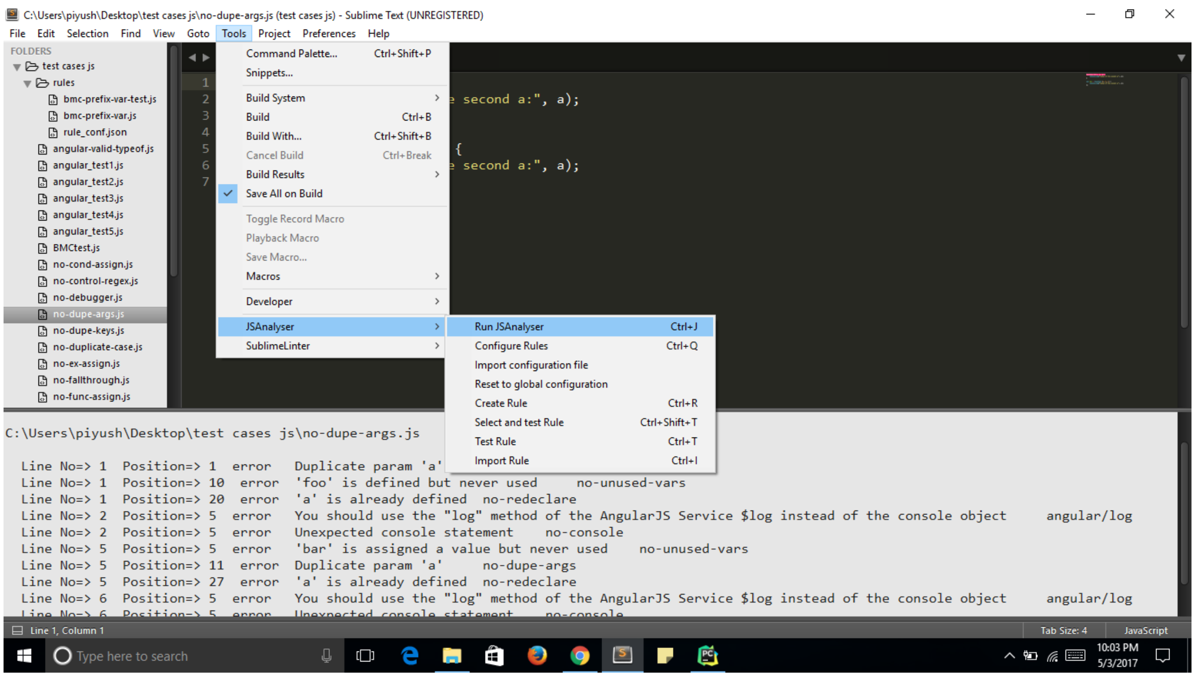
CHAPTER 6

RESULTS AND EVALUATION

6.1 Experimental Setup

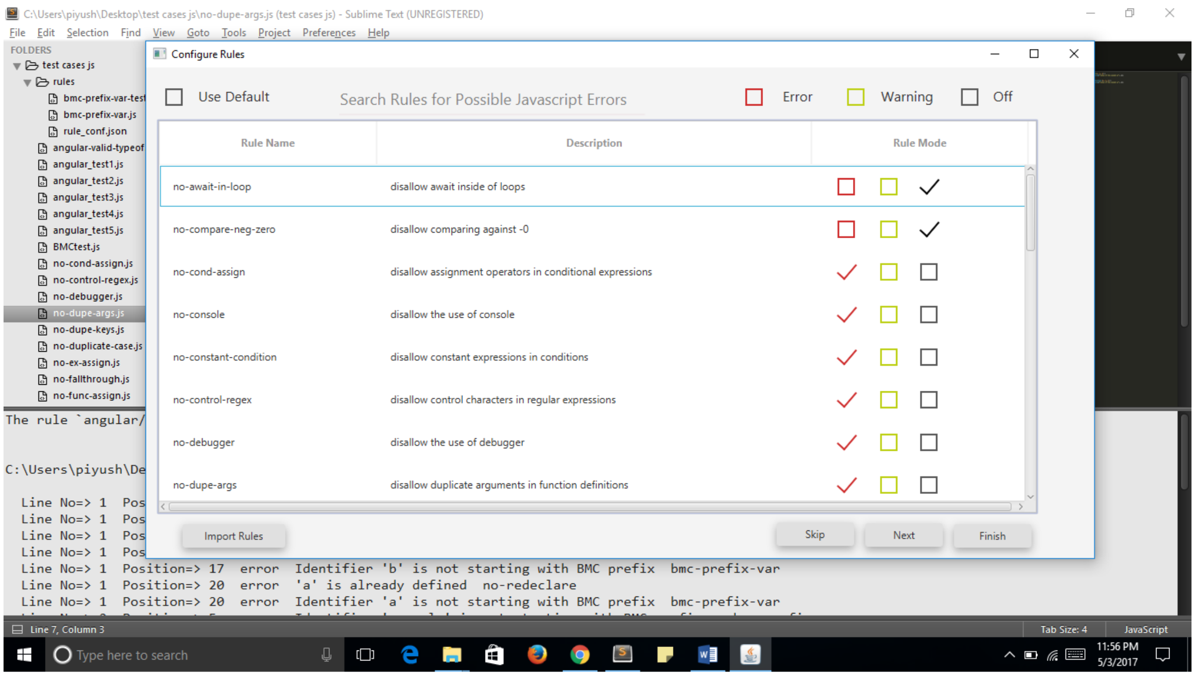
- The project requires latest version of java and Node.js for working. Just make sure they are installed properly. The project is built on top of eslint as a underline moduler framework for javascript linting. So, it is also required to install eslint globally via Node Pacakge Manager.
- After installation of dependencies, clone repo from https://github.com/EslintSublimePlugin/JSubLint

6.2 Screenshots of user interface


C:\Users\piyush\Desktop\test cases js\no-dupe-args.js (test cases js) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Command Palette... Ctrl+Shift+P
Snippets...
Build System
Build Ctrl+B
Build With... Ctrl+Shift+B
Cancel Build Ctrl+Break
Build Results
Save All on Build
Toggle Record Macro
Playback Macro
Save Macro...
Macros
Developer
JSAnalyser
SublimeLinter
Run JSAnalyser Ctrl+J
Configure Rules Ctrl+Q
Import configuration file
Reset to global configuration
Create Rule Ctrl+R
Select and test Rule Ctrl+Shift+T
Test Rule Ctrl+T
Import Rule Ctrl+I
Line 1, Column 1
Tab Size: 4
JavaScript

```
C:\Users\piyush\Desktop\test cases js\no-dupe-args.js

  Line No=> 1  Position=> 1  error  Duplicate param 'a'
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 2  Position=> 5  error  You should use the "log" method of the AngularJS Service $log instead of the console object     angular/log
  Line No=> 2  Position=> 5  error  Unexpected console statement    no-console
  Line No=> 5  Position=> 5  error  'bar' is assigned a value but never used    no-unused-vars
  Line No=> 5  Position=> 11  error  Duplicate param 'a'     no-dupe-args
  Line No=> 5  Position=> 27  error  'a' is already defined  no-redeclare
  Line No=> 6  Position=> 5  error  You should use the "log" method of the AngularJS Service $log instead of the console object     angular/log
  Line No=> 6  Position=> 5  error  Unexpected console statement    no-console
```

Snapshot 6.1 Sublime Plugin



C:\Users\piyush\Desktop\test cases js\no-dupe-args.js (test cases js) - Sublime Text (UNREGISTERED)

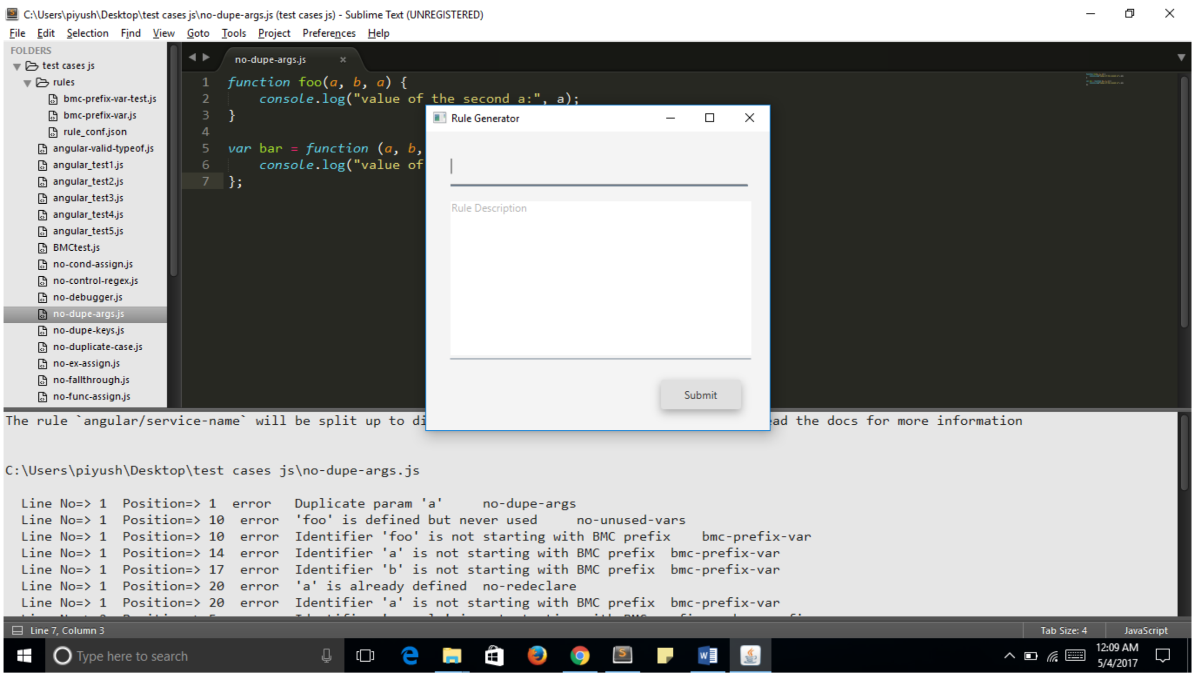
File Edit Selection Find View Goto Tools Project Preferences Help
Configure Rules
Import Rules
Skip
Next
Finish

☐ Use Default   Search Rules for Possible Javascript Errors   ☐ Error ☐ Warning ☐ Off

Rule Name	Description	Rule Mode		
no-await-in-loop	disallow await inside of loops	☐	☐	✓
no-compare-neg-zero	disallow comparing against -0	☐	☐	✓
no-cond-assign	disallow assignment operators in conditional expressions	✓	☐	☐
no-console	disallow the use of console	✓	☐	☐
no-constant-condition	disallow constant expressions in conditions	✓	☐	☐
no-control-regex	disallow control characters in regular expressions	✓	☐	☐
no-debugger	disallow the use of debugger	✓	☐	☐
no-dupe-args	disallow duplicate arguments in function definitions	✓	☐	☐

```
The rule `angular/

C:\Users\piyush\De

  Line No=> 1  Pos
  Line No=> 1  Pos
  Line No=> 1  Pos
  Line No=> 1  Pos
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```

Snapshot 6.2 Rule Editor tool



C:\Users\piyush\Desktop\test cases js\no-dupe-args.js (test cases js) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help
no-dupe-args.js
Rule Generator
Rule Description
Submit

```
function foo(a, b, a) {
    console.log("value of the second a:", a);
}

var bar = function (a, b,
    console.log("value of
};
```

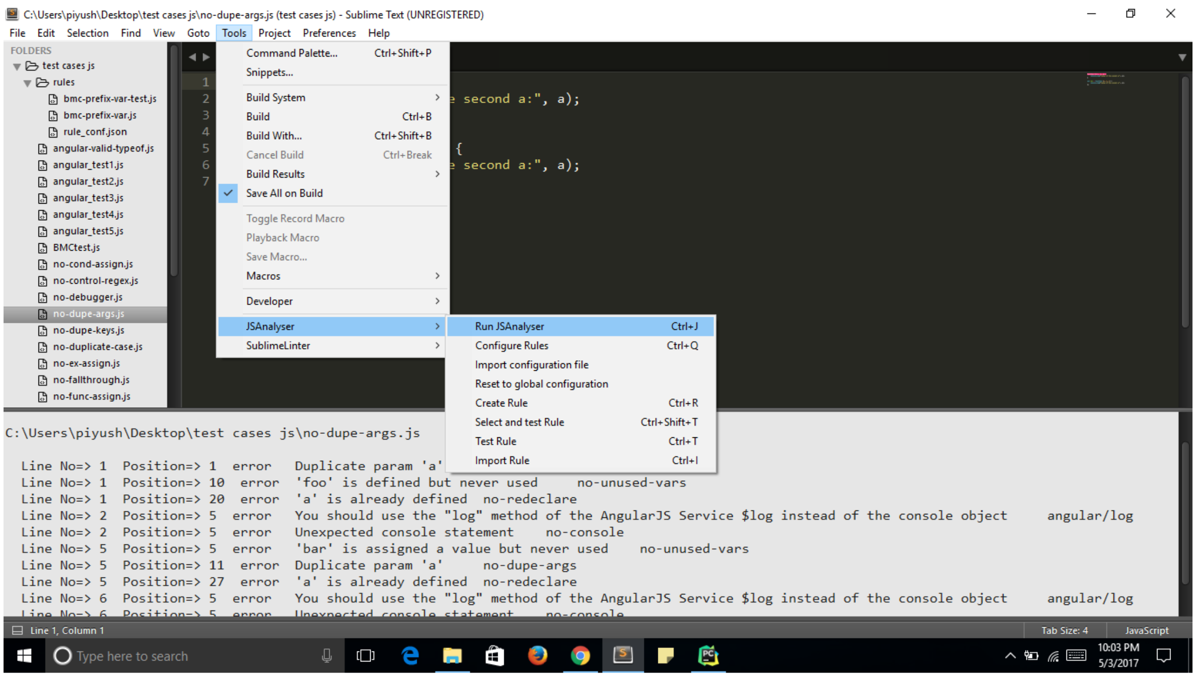
```
The rule `angular/service-name` will be split up to di                                    ad the docs for more information

C:\Users\piyush\Desktop\test cases js\no-dupe-args.js

  Line No=> 1  Position=> 1   error   Duplicate param 'a'     no-dupe-args
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 10  error  Identifier 'foo' is not starting with BMC prefix    bmc-prefix-var
  Line No=> 1  Position=> 14  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```

Snapshot 6.3 Rule Generator tool

6.3 Test Cases:

TEST CASE 1 : Rule analyser on javascript file

Description : This is first option given in plugin to run analyser manually. Automatic event for analysis can also be triggered by ctrl+s event.

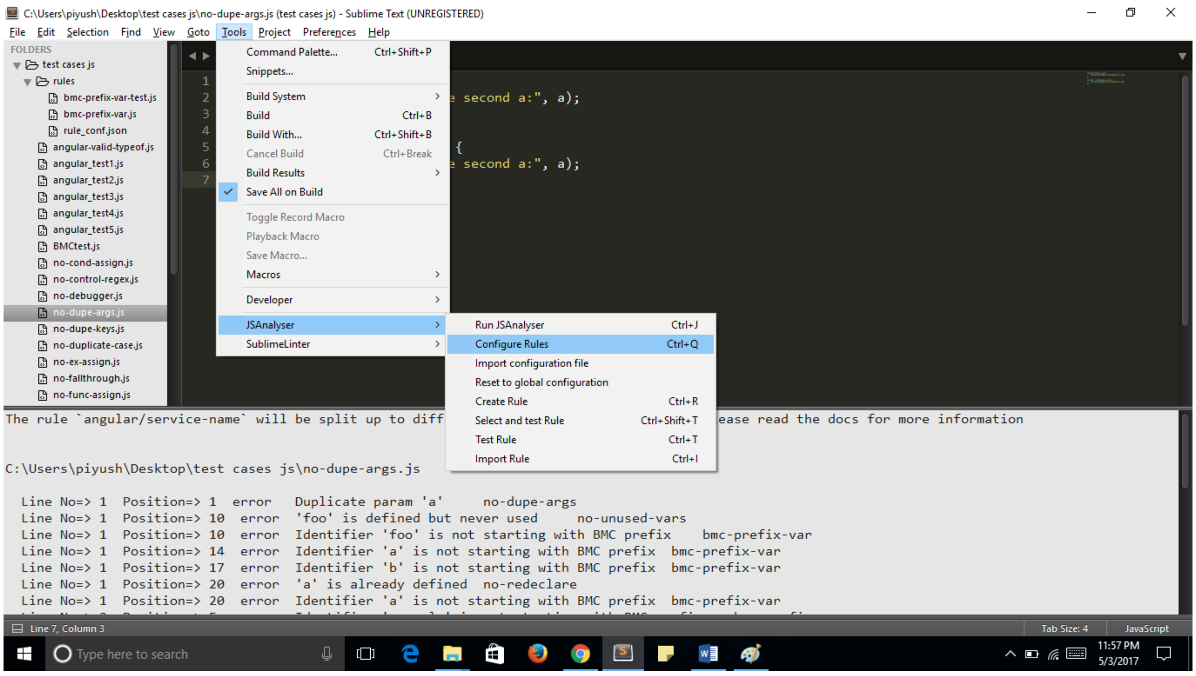

C:\Users\piyush\Desktop\test cases js\no-dupe-args.js (test cases js) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Command Palette... Ctrl+Shift+P
Snippets...
Build System
Build Ctrl+B
Build With... Ctrl+Shift+B
Cancel Build Ctrl+Break
Build Results
Save All on Build
Toggle Record Macro
Playback Macro
Save Macro...
Macros
Developer
JSAnalyser
SublimeLinter
Run JSAnalyser Ctrl+J
Configure Rules Ctrl+Q
Import configuration file
Reset to global configuration
Create Rule Ctrl+R
Select and test Rule Ctrl+Shift+T
Test Rule Ctrl+T
Import Rule Ctrl+I
Line 1, Column 1
Tab Size: 4
JavaScript

```
C:\Users\piyush\Desktop\test cases js\no-dupe-args.js

  Line No=> 1  Position=> 1  error   Duplicate param 'a'
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 2  Position=> 5  error   You should use the "log" method of the AngularJS Service $log instead of the console object     angular/log
  Line No=> 2  Position=> 5  error   Unexpected console statement    no-console
  Line No=> 5  Position=> 5  error   'bar' is assigned a value but never used    no-unused-vars
  Line No=> 5  Position=> 11  error  Duplicate param 'a'     no-dupe-args
  Line No=> 5  Position=> 27  error  'a' is already defined  no-redeclare
  Line No=> 6  Position=> 5  error   You should use the "log" method of the AngularJS Service $log instead of the console object     angular/log
  Line No=> 6  Position=> 5  error   Unexpected console statement    no-console
```

TEST CASE 2 : Configure rules for analysis

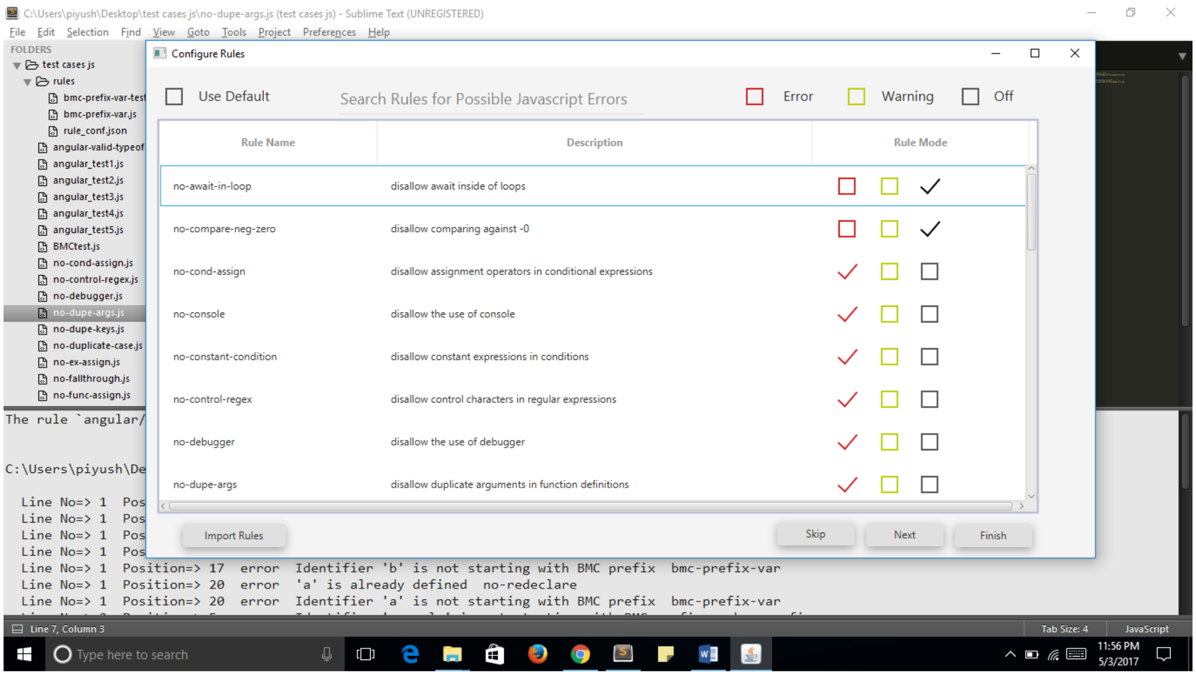
Description : To configure rules

- Click on tools -> JsAnalyser -> Configure rules
- Select the rules and its configuration
- Click on finish and save configuration as JSON files



```
e second a:", a);

{
e second a:", a);
```

```
The rule `angular/service-name` will be split up to diff          ease read the docs for more information

C:\Users\piyush\Desktop\test cases js\no-dupe-args.js

  Line No=> 1  Position=> 1  error   Duplicate param 'a'     no-dupe-args
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 10  error  Identifier 'foo' is not starting with BMC prefix    bmc-prefix-var
  Line No=> 1  Position=> 14  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```


C:\Users\piyush\Desktop\test cases js\no-dupe-args.js (test cases js) - Sublime Text (UNREGISTERED)
Configure Rules
Use Default
Search Rules for Possible Javascript Errors
Error
Warning
Off
Import Rules
Skip
Next
Finish
Line 7, Column 3
Tab Size: 4
JavaScript
Type here to search
11:56 PM
5/3/2017

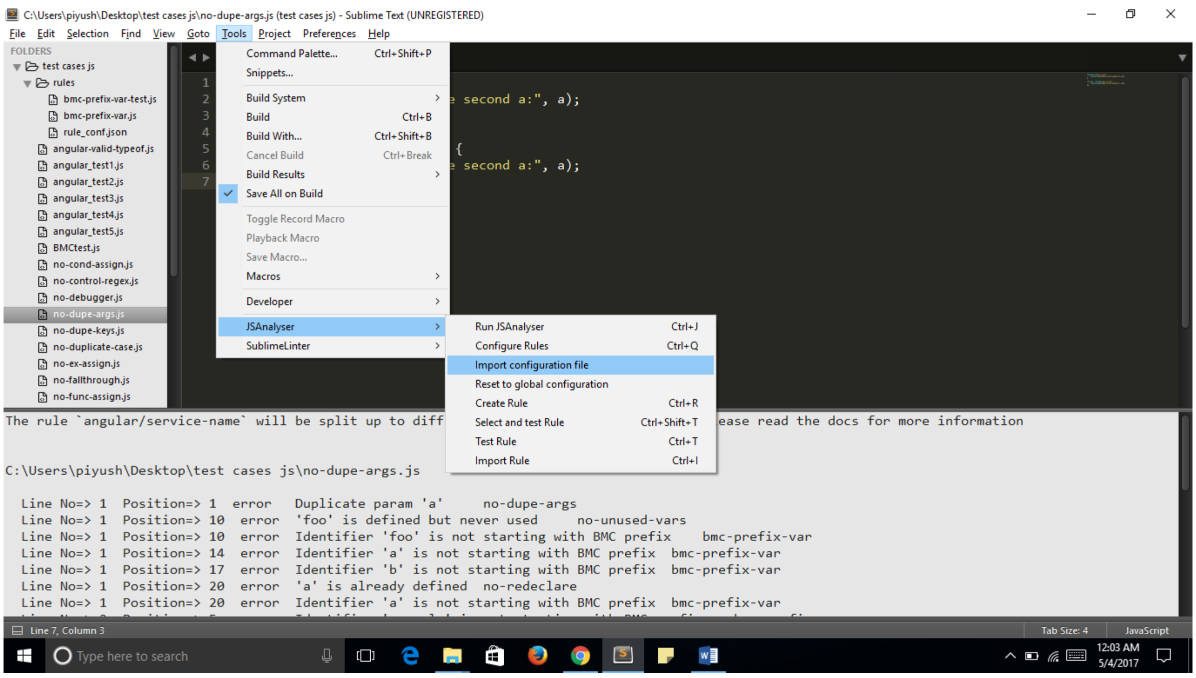
Rule Name	Description	Rule Mode		
no-await-in-loop	disallow await inside of loops	☐	☐	✓
no-compare-neg-zero	disallow comparing against -0	☐	☐	✓
no-cond-assign	disallow assignment operators in conditional expressions	✓	☐	☐
no-console	disallow the use of console	✓	☐	☐
no-constant-condition	disallow constant expressions in conditions	✓	☐	☐
no-control-regex	disallow control characters in regular expressions	✓	☐	☐
no-debugger	disallow the use of debugger	✓	☐	☐
no-dupe-args	disallow duplicate arguments in function definitions	✓	☐	☐

```
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```

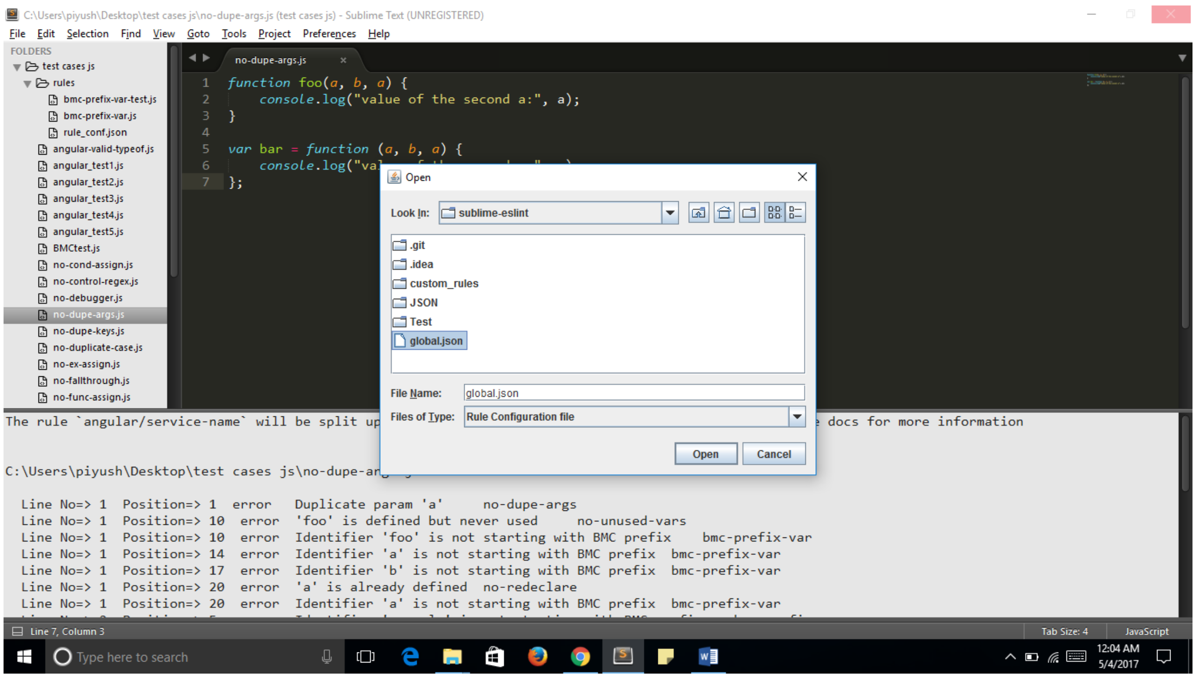
TEST CASE 3 : Import your custom Rules file

Description : To import custom rules file

- Click on tools -> JsAnalyser ->Import configuration file
- Select JSON file



```
The rule `angular/service-name` will be split up to diff                                    ease read the docs for more information

C:\Users\piyush\Desktop\test cases js\no-dupe-args.js

  Line No=> 1  Position=> 1  error   Duplicate param 'a'     no-dupe-args
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 10  error  Identifier 'foo' is not starting with BMC prefix    bmc-prefix-var
  Line No=> 1  Position=> 14  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```


C:\Users\piyush\Desktop\test cases js\no-dupe-args.js (test cases js) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
no-dupe-args.js
Open
Look In: sublime-eslint
.git
.idea
custom_rules
JSON
Test
global.json
File Name: global.json
Files of Type: Rule Configuration file
Open
Cancel
Line 7, Column 3
Tab Size: 4
JavaScript
Type here to search
12:04 AM
5/4/2017

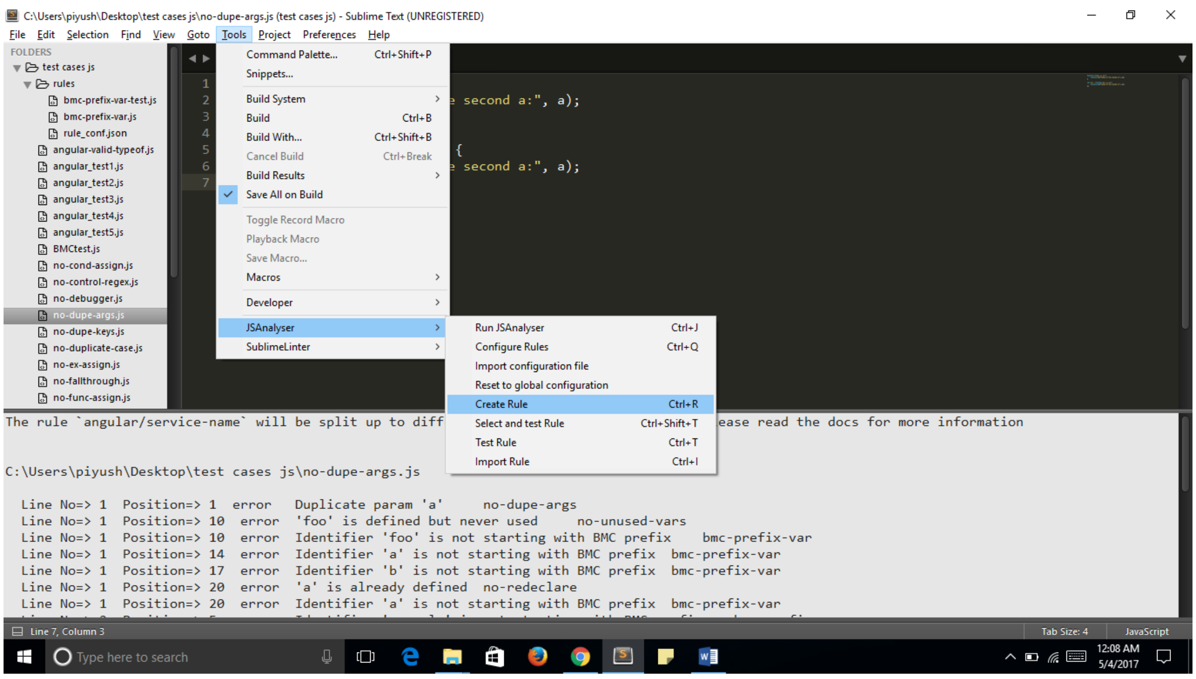
```
function foo(a, b, a) {
    console.log("value of the second a:", a);
}

var bar = function (a, b, a) {
    console.log("va
};
```

```
The rule `angular/service-name` will be split up                                    e docs for more information

C:\Users\piyush\Desktop\test cases js\no-dupe-ar

  Line No=> 1  Position=> 1  error   Duplicate param 'a'     no-dupe-args
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 10  error  Identifier 'foo' is not starting with BMC prefix    bmc-prefix-var
  Line No=> 1  Position=> 14  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```

TEST CASE 4 : Create Rule

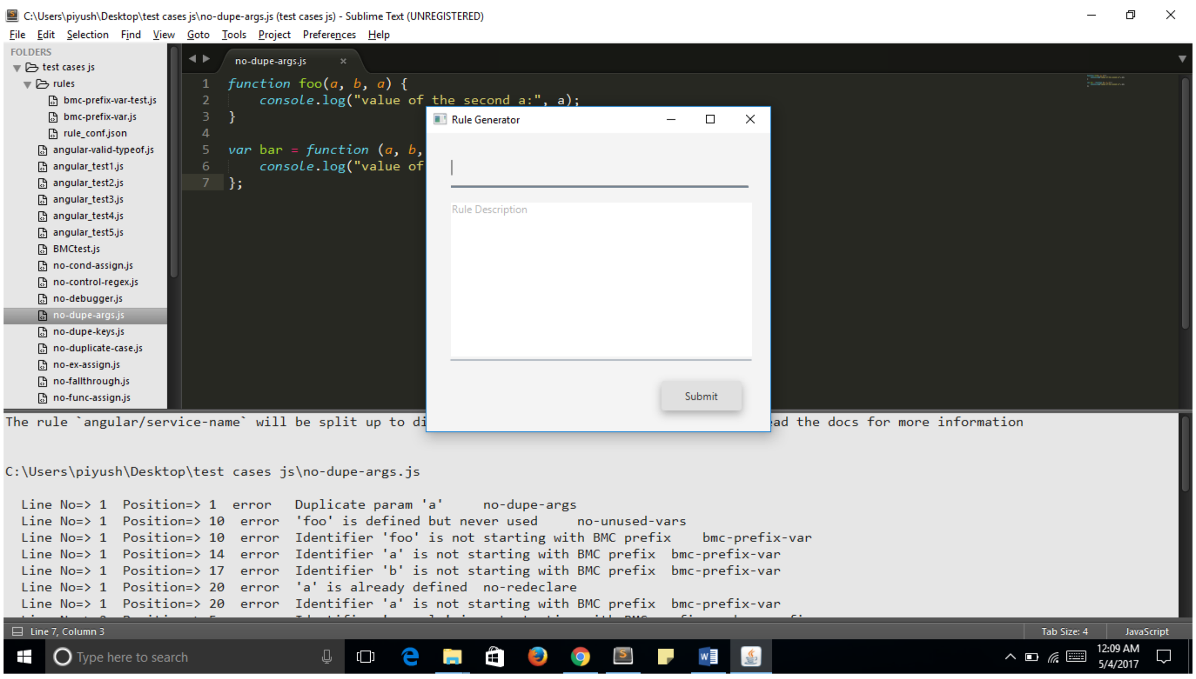
Description : It prompts for Rule Name and Description. Output consist of 4 different files

- Rule file (rule_name.js)
- Rule test file (rule_nametest.js)
- Json configuration file for rule (rule_nameconfig.json)
- Json metadata file for rule (rule_namemetadata.json)



```
e second a:", a);

 {
e second a:", a);
```

```
The rule `angular/service-name` will be split up to diff                              ease read the docs for more information


C:\Users\piyush\Desktop\test cases js\no-dupe-args.js

  Line No=> 1  Position=> 1  error   Duplicate param 'a'     no-dupe-args
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 10  error  Identifier 'foo' is not starting with BMC prefix    bmc-prefix-var
  Line No=> 1  Position=> 14  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```



```
function foo(a, b, a) {
    console.log("value of the second a:", a);
}

var bar = function (a, b,
    console.log("value of
};
```

```
The rule `angular/service-name` will be split up to di                                 ad the docs for more information


C:\Users\piyush\Desktop\test cases js\no-dupe-args.js

  Line No=> 1  Position=> 1  error   Duplicate param 'a'     no-dupe-args
  Line No=> 1  Position=> 10  error  'foo' is defined but never used     no-unused-vars
  Line No=> 1  Position=> 10  error  Identifier 'foo' is not starting with BMC prefix    bmc-prefix-var
  Line No=> 1  Position=> 14  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 17  error  Identifier 'b' is not starting with BMC prefix  bmc-prefix-var
  Line No=> 1  Position=> 20  error  'a' is already defined  no-redeclare
  Line No=> 1  Position=> 20  error  Identifier 'a' is not starting with BMC prefix  bmc-prefix-var
```

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

In this project, we identified code snippets which doesn’t meet the standards provided by BMC software solutions. We used some of language processing tools available on the internet to find out number of lines of code, commented code, unused code and languages used in given input package. We were also able to identify the difference in code between two versions of deployed packages .We have developed Sublime Plugin, Rule generator tool and Rule editing tool for javascript source code analysis.

7.2 Future scope

The future scope of the project is to provide more extensible custom rule generator. Currently we are providing with basic functionalities required for generating new rule. The additional functionality will consist of providing node selectors for AST formed using JsAnalyzer. The template generation window will provide with set of functions which can be embedded into rendered templates. It will also generate documentation for each function along with it.

The project can also be used to handle and fix the errors properly. Currently it doesn’t support this functionality due to limitation on number of rules and their working types. Fixing errors requires going through the code and fix error part of the code. After adding proper code snippets, this functionality can be achieved.

REFERENCES

[1] Zeineb Zhioua, Stuart Short ,Yves Roudier : “Static code analysis for software security verification: problems and approaches” ,21-25 July 2014.

[2] Herbart Prahofer , Florian Angerer :“Opportunities and challenges of static code analysis of IEC61131-3 programs”, 17-21 Sept. 2012.

[3] P. Mayer, A. Schroeder: “Cross-Language code analysis and refactoring”, 24 December 2012.

[4] Zaigham Mushtaq , Ghulam Rasool : “Multilingual source code analysis: State of the art and challenges”, 17-19 Dec. 2015.

[5] Melina Kulenovic, Dzenana Donko: “A survey of static code analysis methods for security vulnerabilities detection”, 26-30 May 2014.

24.docx

ORIGINALITY REPORT

7%	4%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	datapreservation.usgs.gov Internet Source	1%
2	www.seguetech.com Internet Source	1%
3	dblp.dagstuhl.de Internet Source	1%
4	Mayer, Philip, and Andreas Schroeder. "Cross-Language Code Analysis and Refactoring", 2012 IEEE 12th International Working Conference on Source Code Analysis and Manipulation, 2012. Publication	1%
5	Prahofer, Herbert, Florian Angerer, Rudolf Ramler, Hermann Lacheiner, and Friedrich Grillenberger. "Opportunities and challenges of static code analysis of IEC 61131-3 programs", Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), 2012. Publication	1%
	laptophub.com.pk	

6 Internet Source <1%

7 Zaigham Mushtaq, Ghulam Rasool, Balawal Shahzad. "Extendable and customizable features for the detection of JEA/J2EE patterns", 2016 International Conference on Open Source Systems & Technologies (ICOSST), 2016
Publication <1%

8 Hussein, Sherif. "Education Quality Control Based on System Dynamics and Evolutionary Computation", Modeling Simulation and Optimization - Focus on Applications, 2010.
Publication <1%

9 Binkley, David, Nicolas Gold, Mark Harman, Syed Islam, Jens Krinke, and Shin Yoo. "ORBS: language-independent program slicing", Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering - FSE 2014, 2014.
Publication <1%

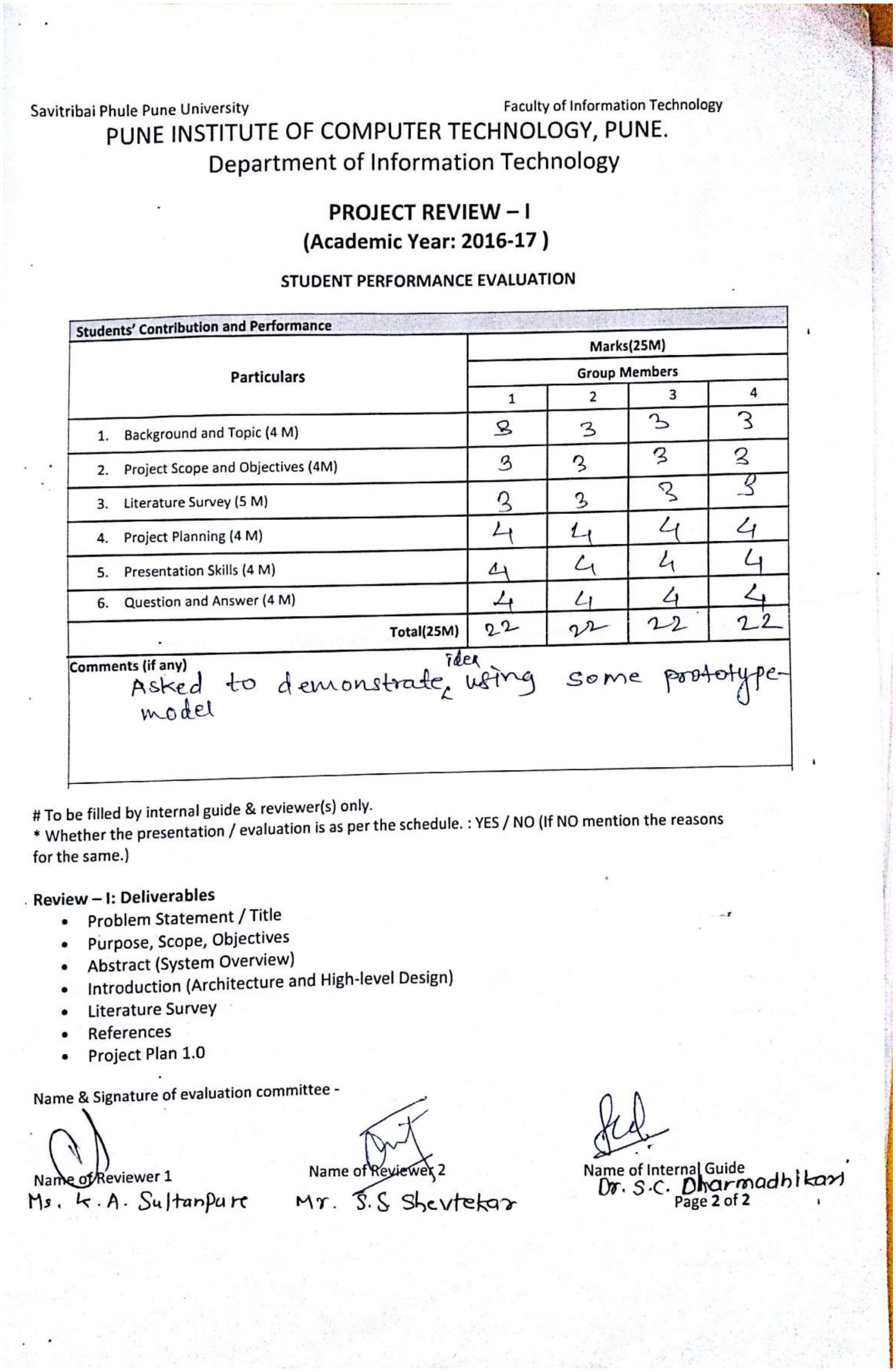
10 www.docstoc.com
Internet Source <1%

11 scholar.lib.vt.edu
Internet Source <1%

12 Lecture Notes in Computer Science, 2016.
Publication <1%

13	Kulenovic, Melina, and Dzenana Donko. "A survey of static code analysis methods for security vulnerabilities detection", 2014 37th International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO), 2014. Publication	<1%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	< 1 WORDS
EXCLUDE BIBLIOGRAPHY	OFF		



Savitribai Phule Pune University — Faculty of Information Technology

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE.

Department of Information Technology

PROJECT REVIEW – I

(Academic Year: 2016-17)

STUDENT PERFORMANCE EVALUATION

Students' Contribution and Performance

Particulars	Marks(25M)			
	Group Members			
	1	2	3	4
1. Background and Topic (4 M)	3	3	3	3
2. Project Scope and Objectives (4M)	3	3	3	3
3. Literature Survey (5 M)	3	3	3	3
4. Project Planning (4 M)	4	4	4	4
5. Presentation Skills (4 M)	4	4	4	4
6. Question and Answer (4 M)	4	4	4	4
Total(25M)	22	22	22	22

Comments (if any)
Asked to demonstrate idea using some prototype-model

To be filled by internal guide & reviewer(s) only.
* Whether the presentation / evaluation is as per the schedule. : YES / NO (If NO mention the reasons for the same.)

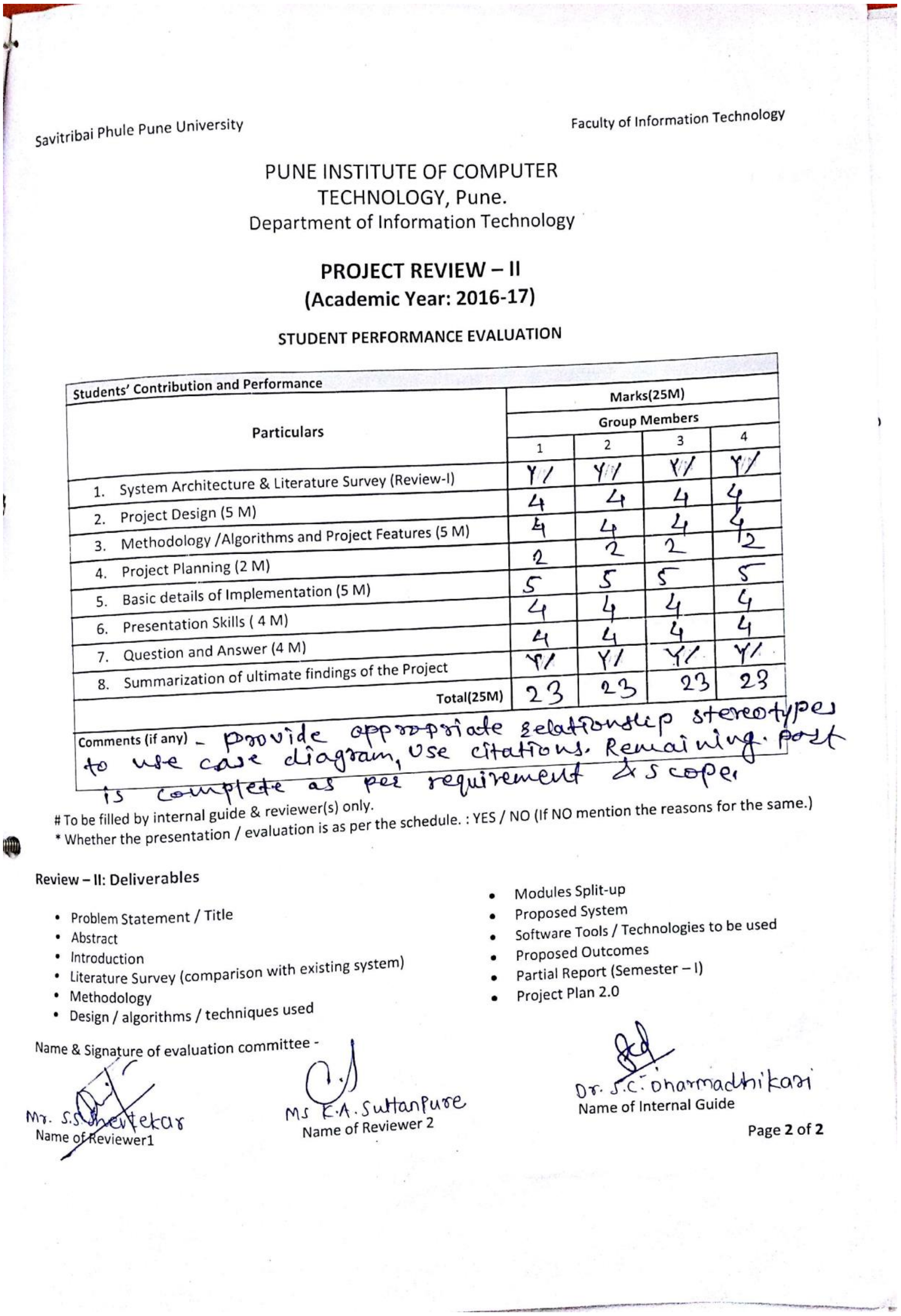
Review – I: Deliverables

- Problem Statement / Title
- Purpose, Scope, Objectives
- Abstract (System Overview)
- Introduction (Architecture and High-level Design)
- Literature Survey
- References
- Project Plan 1.0

Name & Signature of evaluation committee -

Name of Reviewer 1	Name of Reviewer 2	Name of Internal Guide
[illegible]	[illegible]	[illegible]
Ms. K. A. Sultanpure	Mr. S.S Shevtekar	Dr. S.C. Dharmadhikari

Page 2 of 2



Savitribai Phule Pune University

Faculty of Information Technology

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, Pune.

Department of Information Technology

PROJECT REVIEW – II

(Academic Year: 2016-17)

STUDENT PERFORMANCE EVALUATION

Students' Contribution and Performance				
Particulars	**Marks(25M)**			
	Group Members			
	1	2	3	4
1. System Architecture & Literature Survey (Review-I)	Y/	Y/	Y/	Y/
2. Project Design (5 M)	4	4	4	4
3. Methodology /Algorithms and Project Features (5 M)	4	4	4	4
4. Project Planning (2 M)	2	2	2	2
5. Basic details of Implementation (5 M)	5	5	5	5
6. Presentation Skills (4 M)	4	4	4	4
7. Question and Answer (4 M)	4	4	4	4
8. Summarization of ultimate findings of the Project	Y/	Y/	Y/	Y/
Total(25M)	23	23	23	23

Comments (if any) – Provide appropriate relationship stereotypes to use case diagram, Use citations. Remaining part is complete as per requirement & scope.

To be filled by internal guide & reviewer(s) only.

* Whether the presentation / evaluation is as per the schedule. : YES / NO (If NO mention the reasons for the same.)

Review – II: Deliverables

- Problem Statement / Title
- Abstract
- Introduction
- Literature Survey (comparison with existing system)
- Methodology
- Design / algorithms / techniques used
- Modules Split-up
- Proposed System
- Software Tools / Technologies to be used
- Proposed Outcomes
- Partial Report (Semester – I)
- Project Plan 2.0

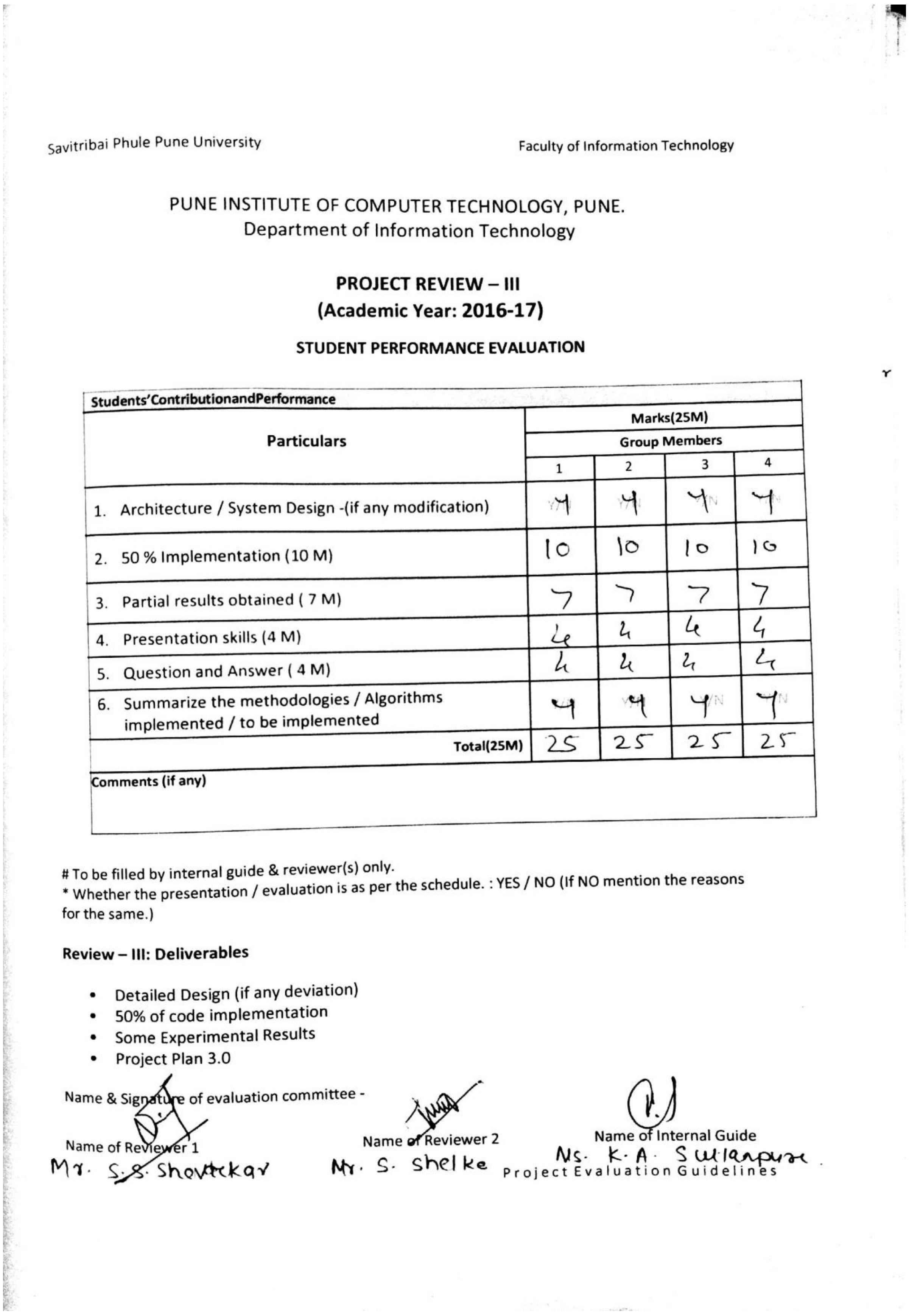
Name & Signature of evaluation committee -

[illegible]
Mr. S.S. Shertekar
Name of Reviewer1

[illegible]
Ms K.A. Sultanpure
Name of Reviewer 2

[illegible]
Dr. S.C. Dharmadhikari
Name of Internal Guide

Page **2** of **2**



Savitribai Phule Pune University

Faculty of Information Technology

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE.
Department of Information Technology

PROJECT REVIEW – III
(Academic Year: 2016-17)

STUDENT PERFORMANCE EVALUATION

Students'ContributionandPerformance

Particulars	**Marks(25M)**			
	Group Members			
	1	2	3	4
1. Architecture / System Design -(if any modification)	4	4	4	4
2. 50 % Implementation (10 M)	10	10	10	10
3. Partial results obtained (7 M)	7	7	7	7
4. Presentation skills (4 M)	4	4	4	4
5. Question and Answer (4 M)	4	4	4	4
6. Summarize the methodologies / Algorithms implemented / to be implemented	4	4	4	4
Total(25M)	25	25	25	25

Comments (if any)

To be filled by internal guide & reviewer(s) only.
* Whether the presentation / evaluation is as per the schedule. : YES / NO (If NO mention the reasons for the same.)

Review – III: Deliverables

- Detailed Design (if any deviation)
- 50% of code implementation
- Some Experimental Results
- Project Plan 3.0

Name & Signature of evaluation committee -

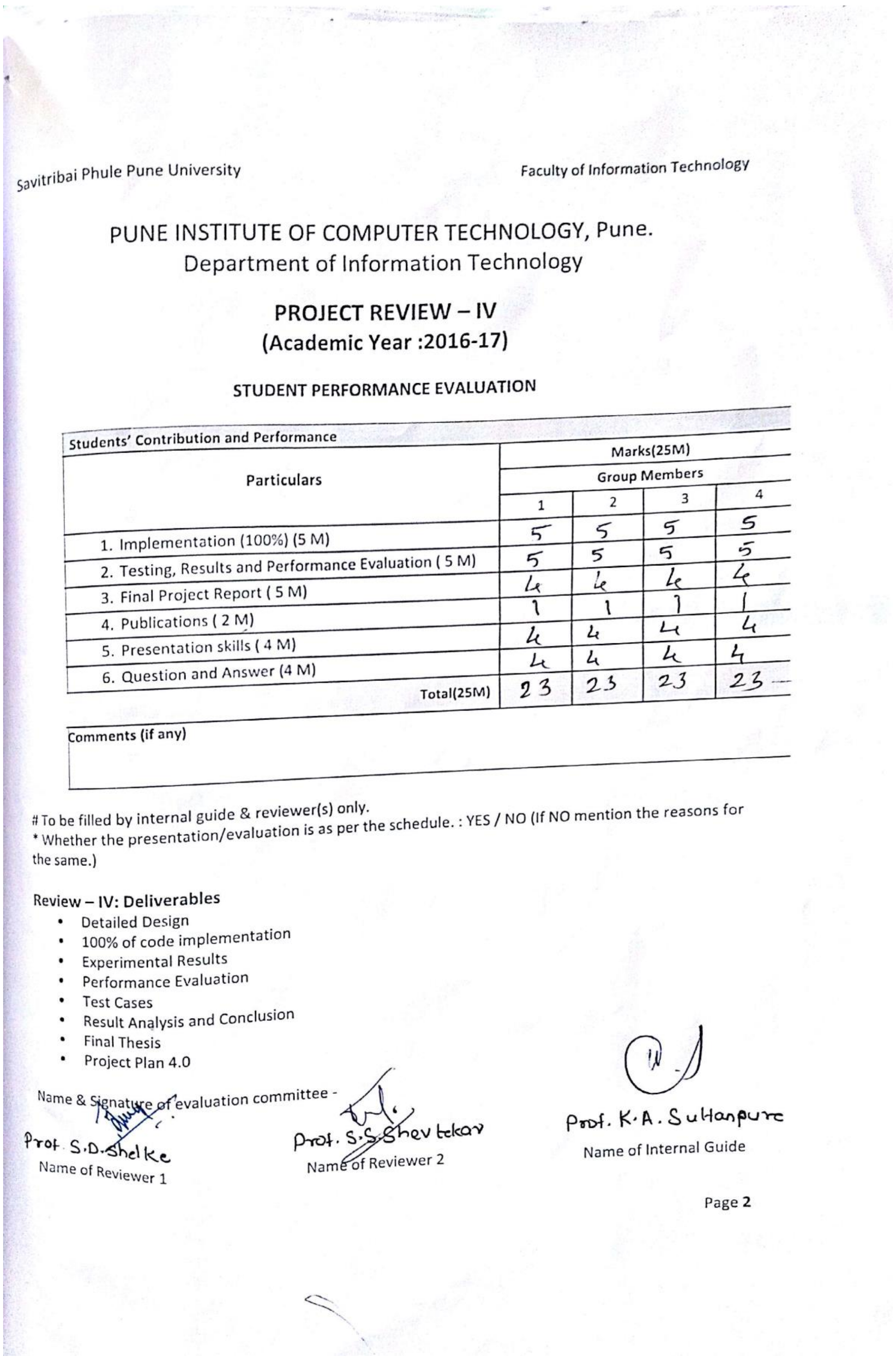
[illegible] [illegible] [illegible]

Name of Reviewer 1
Mr. S.S. Shovtekar

Name of Reviewer 2
Mr. S. Shelke

Name of Internal Guide
Ms. K. A. Sultanpure

Project Evaluation Guidelines



Savitribai Phule Pune University | Faculty of Information Technology

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, Pune.
Department of Information Technology

PROJECT REVIEW – IV
(Academic Year :2016-17)

STUDENT PERFORMANCE EVALUATION

Students' Contribution and Performance

Particulars	**Marks(25M)**			
	Group Members			
	1	2	3	4
1. Implementation (100%) (5 M)	5	5	5	5
2. Testing, Results and Performance Evaluation (5 M)	5	5	5	5
3. Final Project Report (5 M)	4	4	4	4
4. Publications (2 M)	1	1	1	1
5. Presentation skills (4 M)	4	4	4	4
6. Question and Answer (4 M)	4	4	4	4
Total(25M)	23	23	23	23

Comments (if any)

To be filled by internal guide & reviewer(s) only.
* Whether the presentation/evaluation is as per the schedule. : YES / NO (If NO mention the reasons for the same.)

Review – IV: Deliverables

- Detailed Design
- 100% of code implementation
- Experimental Results
- Performance Evaluation
- Test Cases
- Result Analysis and Conclusion
- Final Thesis
- Project Plan 4.0

Name & Signature of evaluation committee -

[illegible] [illegible] [illegible]

Prof. S.D. Shelke
Name of Reviewer 1

Prof. S.S. Shevtekar
Name of Reviewer 2

Prof. K.A. Sultanpure
Name of Internal Guide

Page **2**