PROJECT REPORT

ON

DEEP: DEvnagari Enabled Programming

Submitted By

Ruchita Kolte

Mayur Jagtap

Nikita Babhale

Shubham Halle

in partial fulfilment for the award of the degree of

Bachelor of Engineering

of

Savitribai Phule Pune University

IN

DEPARTMENT OF INFORMATION TECHNOLOGY

MIT COLLEGE OF ENGINEERING

PUNE

2016-17

PROJECT REPORT

ON

DEEP: DEvnagari Enabled Programming

Submitted By

Ruchita Kolte

Mayur Jagtap

Nikita Babhale

Shubham Halle

Guided by

Prof. Aditi Jahagirdar

DEPARTMENT OF INFORMATION TECHNOLOGY

MIT COLLEGE OF ENGINEERING

PUNE

SAVITRIBAI PHULE PUNE UNIVERSITY

2016 – 2017

DEPARTMENT OF INFORMATION TECHNOLOGY

Certificate

This is to certify that,

B120388612: - Ruchita Kolte

B120388584: - Mayur Jagtap

B120388515: - Nikita Babhale

B120388572: - Shubham Halle

have successfully completed this project report entitled "**DEEP: Devanagari Enabled Programming**", under my guidance in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Department of Information Technology of Savitribai Phule Pune University, Pune during the academic year 2016-17.

Date: -

Place: -

Prof. Aditi Jahagirdar

Project Guide

Dr. Anil Hiwale

Head of Department

ACKNOWLEDGEMENT

We take this opportunity to thank our project guide **Prof. Aditi Jahagirdar** and Head of the Department **Dr. Anil Hiwale** for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the **Department of Information Technology of MIT College of Engineering, Pune** for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

Ms. Ruchita Kolte

Mr. Mayur Jagtap

Ms. Nikita Babhale

Mr. Shubham Halle

ABSTRACT

Technology has become the fourth basic need of man's life. So, the need for computer programming is arising every passing day. However, since most of the computer programming languages are English-based, they can act as barriers for people who are not comfortable with English. Mother tongue is essential for learning as a part of intellectual ability. It helps a child in his/her moral, mental and emotional development. Mother tongue has central role in education that demands cognitive development. Studies show that children who come to school with a solid foundation in their mother tongue develop stronger literacy & logical abilities. DEEP is an initiative that will provide the Marathi speaking people, a Non-English based programming system which will enable them to learn and write computer programs in Marathi. It will also provide the Marathi students a platform to learn and practice computer programming at the Elementary school level itself.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 PROJECT IDEA

Developing a system that will enable the users to write their code using Marathi language and then compile the code to get the result or errors if any.

1.2 MOTIVATION OF THE PROJECT

We have been losing the existence of our native languages over the past few decades[6]. Lack of logical ability and research in particular native language is one of the reasons behind it[6]. Also, introduction of programming to Primary schools can be done in better way by facilitating them to use their own mother tongue (Native Language). People will be more comfortable while learning, developing logic as well as implementing it in the computer programs in their native languages which they understand inside out[3]. The Government of India has been encouraging various Digital India initiatives since 2014, to transform India into a digitally empowered society and knowledge economy. This project can be a step taken towards the accomplishment of the initiative. This soft-ware accomplishes to achieve all of the above objectives. This project aims at removing language barrier of English based programming languages, by allowing people from non-English background to take on software development, especially, Devnagari and Marathi background.

1.3 INTRODUCTION

Multiple studies have shown that, learning in Mother Tongue is an essential key for intellectual development of a person[3]. Also, it has been found that, we have been losing the existence of our native languages over the past few decades[6]. Lack of logical ability and research in particular language is one of the reasons behind it[4]. Besides that, introduction of programming at Primary schools can be done in better way by facilitating them to use their own mother tongue (Native Language)[2]. People will be more comfortable while learning, developing logic as well as implementing it in the computer programs in their native languages which they understand inside out[4].

The Government of India has been encouraging various Digital India initiatives since 2014, to transform India into a digitally empowered society and knowledge economy. This

project can be a step taken towards the accomplishment of the initiative. This project accomplishes to achieve all of the above objectives.

1.4 APPLICATION & NEED

DEEP aims at satisfying the following

Application –

1. DEEP can be used to allow masses to program for themselves.
2. Introduce programming at elementary school level.
3. As part of language conservation.

Needs –

1. Removing language barrier for programming.
2. Capability to introduce programming at elementary level in *Marathi* language.
3. Conservation of native linguistic culture.

CHAPTER 2

LITERATURE SURVEY

[1] ChaScript: Breaking Language Barrier using a Bengali Programming System

8th International Conference on Electrical and Computer Engineering 20-22 December, 2014, Dhaka, Bangladesh.

ChaScript is non English based educational programming for the people who are not adequate in English. It helps Bengali people to write programs in Bengali. Social Cognitive Theory states that, learning from other people by means of observing them is an effective way of gaining knowledge and altering behaviour. In order to build the programming system, the grammar of the scripting language ECMAScript was used and the grammar was parsed using a JavaScript parser generator, Jison. It would be useful for Bengali students to use as a platform to learn computer programming.

In this paper they have presented a Bengali educational programming language called ChaScript. It was built in order to create a platform for Bengali speaking people to learn computer programming without the need for English Language. To understand the need of Bengali programming language first we need to know about Social Cognitive Theory. Social Cognitive Theory states that, learning from other people by means of observing them is an effective way of gaining knowledge and altering behaviour. In case of common programming languages it is a tough task. Not only do these programming languages have very complicated structures but their documentations are also a huge barrier for people with limited knowledge of English. This is where ChaScript came in handy. It gave a gateway for new programmers to start coding and getting the hang of it through observation and do it so without any particular need of supervision.

ChaScript, was designed for students to use it easily for solving computational and mathematical problems. In order to build the programming system, the grammar of the scripting language ECMAScript was used and the grammar was parsed using a JavaScript parser generator, Jison. Since ChaScript contains modified parser syntax and it can be considered as a translated version of ECMAScript.

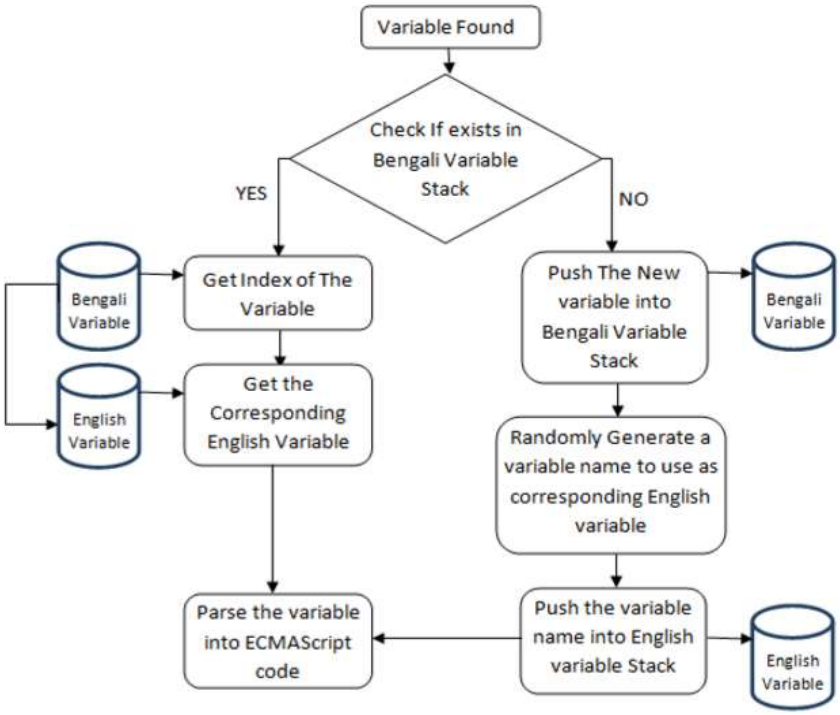


Figure 2.1: Variable Handling

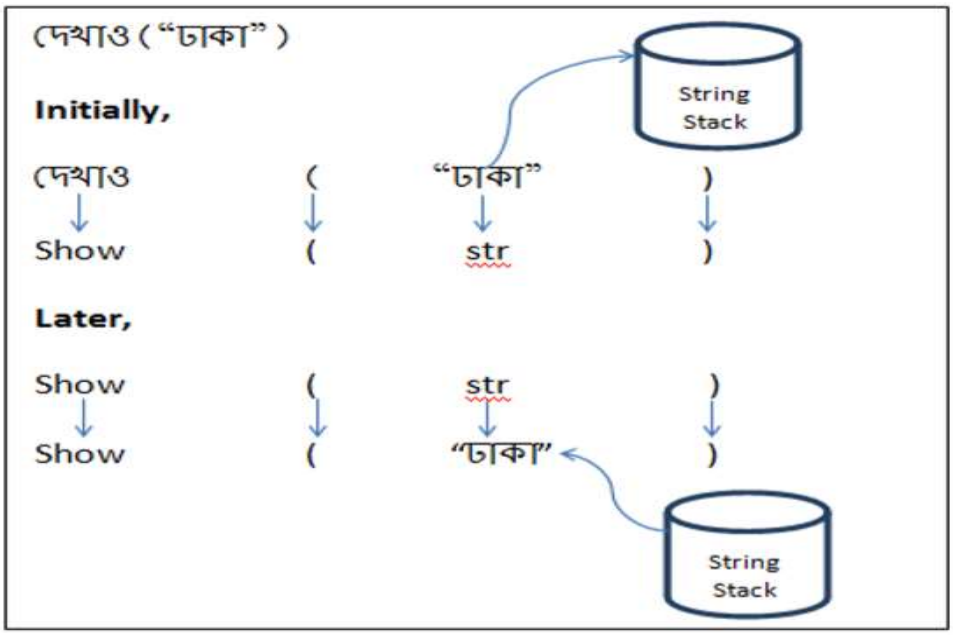


Figure 2.2: String Handling

[2] Dolittle — Experiences in Teaching Programming at K12 Schools

Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing (C5'04).

In this paper, they have first described the outline of Dolittle. Then they have reported some examples of using Dolittle in the lessons at lower and upper secondary schools. Dolittle is Japanese based educational programming language. It uses concept of interepreter, i.e. single line programs, statements. No functions support is provided. It is developed in Java, so that it can be run on variety of computer environments, most often used in schools. Dolittle has been extended to support various programming paradigms such as robot control, graphics and distributed programming.

The design policy of Dolittle is listed as follows:

- Simple syntax with Japanese words
- Incremental programming
- Text-based programming
- Object-oriented
- Open expandability

The practicality of Dolittle was examined through experimental lessons. In the small experimental lessons for high school students, it was confirmed that students understood the principles of software through the programming experience. In the large experimental lessons for junior high school students, the programming lessons could be conducted in classrooms using Dolittle. Then they verified that elementary and junior high school students were able to learn programming effectively using robots controlled by programs. Then the idea of making Dolittle objects network capable was introduced and the lessons learned were discussed using the distributed sharing version of Dolittle.

[3] Noormohamadi : Mother tongue, a necessary step to intellectual development.

2008 Journal of Pan-Pacific Association of Applied Linguistics, 12(2), 25-36.

Language is primary tool from childhood to adult stage. It helps in social interaction, understanding, reasoning etc. Mother tongue is essential for learning as a part of intellectual ability. It helps a child in his or her moral, mental and emotional development. Mother tongue has central role in education that demands cognitive development. Studies show that children who come to school with solid foundation in their mother tongue show stronger intellectual abilities.

Mother tongue (first language or native language) is essential for learning as a part of intellectual ability. Mother tongue is the language human beings acquire from birth. It helps the child in his/her mental, moral, and emotional development. Schick, de Villiers, and Hoffmeister (2002) in their study explain that language delays typically observed in deaf children are causally related to delays in major aspects of cognitive development. They maintain, children who cannot understand complex syntactic forms like complements have difficulty understanding how their own thoughts and beliefs may differ from those around them. In fact, much of a child's future social and intellectual development hinges on the milestone of mother tongue (Plessis, 2008). Mother tongue, therefore, has a central role in education that demands cognitive development.

Education is a potential instrument for encouraging independent thinking among the learners. Students should be allowed and encouraged to come up with their own opinions and interpretations of events around them. The curriculum for primary school / elementary school in mother tongue, emphasizes the importance of the individual's personal and intellectual development. Studies show that children who come to school with a solid foundation in their mother tongue develop stronger literacy abilities. Overall, the research is very clear about the importance of children's mother tongue for their personal and educational development. When parents spend time with their children and tell stories or discuss issues with them in a way that develops their mother tongues' vocabularies and concepts, children come to school well prepared to learn and succeed educationally.

CHAPTER 3

PROJECT DEFINITION & SCOPE

3.1 WHAT IS TO BE DEVELOPED ?

DEEP is a way to programme in Marathi. It is actually a compiler and NOT a new programming language. This project aims at removing language barrier of English based programming languages, by allowing people from non-English background to take on software development, especially, Devanagari and Marathi background. The reason we call it Devanagari Enabled programming is that although it works for Marathi keywords as of now, the project has the capability to incorporate different language keywords using their respective language keywords.

3.2 PROBLEM STATEMENT

To implement DEEP - DEvnagari Enabled Programming, a Non-English based programming system which will enable learning and writing computer programs in Marathi, by translating given *Marathi* code into equivalent *English* code and feeding it to the classical compiler/interpreter.

3.3 TECHNOLOGY USED

Python

Python is cross platform interpreted programming language with wide support from global open source community. Python 2.7 has enormous plugins & libraries support for almost all aspects of programming. Beside that python also provides us way for functional, procedural as well as object oriented programming.

Advantages specific to our project for using python are:

1. Support for Unicode handling
2. OOP features
3. Operating System Library
4. Other features listed here,

PyCharm IDE

DEEP has been developed in PyCharm using Python 2.7. PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python

language. It is developed by the Czech company JetBrains. PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License and there is also Professional Edition released under a proprietary license.

PyQT5

DEEP interface is developed using PyQT5. Qt is a set of C++ libraries and development tools that includes platform independent abstractions for graphical user interfaces, networking, threads, regular expressions, SQL databases, SVG, OpenGL, XML, user and application settings, positioning and location services, short range communications (NFC and Bluetooth) and access to the cloud. PyQt5 implements over 1000 of these classes as a set of Python modules.

PyQt5 supports the Windows, Linux, UNIX, Android, OS X and iOS platforms. PyQt does not include a copy of Qt. You must obtain a correctly licensed copy of Qt yourself. However, binary wheels of the GPL version of PyQt5 are provided and these include a copy of the appropriate parts of the LGPL version of Qt. PyQt5 comprises a number of different components. First of all there are a number of Python extension modules. These are all installed in the PyQt5 Python package.

PyParsing Library

DEEP includes grammar for various syntaxes. This grammar is developed using PyParsing library. The purpose of the pyparsing module is to give programmers using the Python programming language a tool for extracting information from structured textual data. The way that the pyparsing module works is to match patterns in the input text using a recursive descent parser: we write syntax productions and pyparsing provides a machine that matches the input text against those productions.

The pyparsing module works best when you can describe the exact syntactic structure of the text you are analyzing. A common application of pyparsing is the analysis of log files. Log file entries generally have a predictable structure including such fields as dates, IP addresses, and such. Possible applications of the module to natural language work are not addressed here.

3.4 METHODOLOGY

ACCEPTING MARATHI INPUT

The interface provided by DEEP has a reserved text space dedicated for entering the Marathi code. This code is has to be written using the specific keywords and should follow the syntax required for the code to work. This code is then compiled and executed. If the code is correct, the result of the code gets displayed on the output text space on the interface. Similarly, if the code has some errors, the errors are pointed out and displayed in the same output address space on the interface.

TOKENIZATION

The code entered is parsed and the tokens are formed from the code. Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining. Tokenization is used in computer science, where it plays a large part in the process of lexical analysis.

INTERMEDIATE CODE FORMATION AND EXECUTION

The tokens formed are then matched against the list of Marathi keywords. If a match found, the corresponding English keyword is selected and printed into the Intermediate code file. If match is not found, error is displayed to the user that the syntax is not valid. Once match is found for all the entered keywords, the intermediate code formed is fed to the Python compiler which executes the code.

The result of this execution is once again parsed and then goes through the above mentioned steps. This English output is also converted back into the Marathi output and then displayed to the user.

CHAPTER 4

SOFTWARE REQUIREMENT & SPECIFICATION

4.1 GATHERING & ANALYZING USING UML:

Product Overview:

DEEP consists of a python interpreter and a compiler integrated with DEEP GUI. DEEP GUI is developed using Pyqt5 library. The purpose of DEEP is to encourage users who are not comfortable using English language to learn to develop and implement code in Marathi. DEEP compiler is responsible for tokenization and parsing of the code statements entered by the user. Intermediate code gets generated after tokenization and parsing. This intermediate code is nothing but an English equivalent of the typed Marathi code. Python Compiler executes the intermediate code to generate the output in English. This English output is then processed through the compiler to obtain the final output in Marathi.

Overview of responsibilities of Developer:

Responsibilities of Developer:

- Building code for tokenization of the statements.
- Building code for parsing the tokens.
- Storing and maintaining the keywords.
- Handling numbers and variables separately.
- Building code for GUI using Pyqt5 library.
- Documentation.

USAGE SCENARIO:

User Profiles:

Table 4.1: User Profiles

Sr.	Actor	Description
1	End User	Start GUI and write code to execute.
2	DEEP compiler	1. Perform tokenization, parsing and statement formation on *Marathi* code. 2. Generate intermediate code & feed to python interpreter. 3. Process output from python interpreter and print it in the input language.
3	Python Interpreter	Execute python statements and generate output in a file.

Use cases:

Table 4.2: Usecases

Sr.	Use case	Description	Actors	Assumptions
1	Start DEEP	DEEP GUI is started by an end user.	End User	Started using python 2.7 & other dependencies, by

				executing GUITest.py
2	Write Marathi code	Code to be executed is written	End User	User knows all syntaxes for writing the code.
3	Execute	The written code is executed.	End User	Developed logic is correct.
4	Tokenization	Code/output is tokenized and parsed by an interpreter.	DEEP Compiler	User finished typing of the code.
5	Generate Intermediate code for Python Interpreter	Code is written on the window of GUI.	DEEP Compiler	Tokens of the code are parsed successfully.
6	Process output from Python Interpreter	The English output is converted into corresponding Marathi output.	DEEP Compiler	Code is executed or if errors present, they are entered in a file.
7	Execute Intermediate code	Intermediate code is executed by python interpreter.	Python Interpreter	Intermediate python code is correct.
8	Generate output to file	1. Output of intermediate code is printed in a file. 2.Output is printed on GUI in Marathi language	1.Python Interpreter 2.DEEP Compiler	1. No errors found. 2. Output is processed through the interpreter successfully.

Use case view:

Figure 4.1 shows the Use Case diagram:

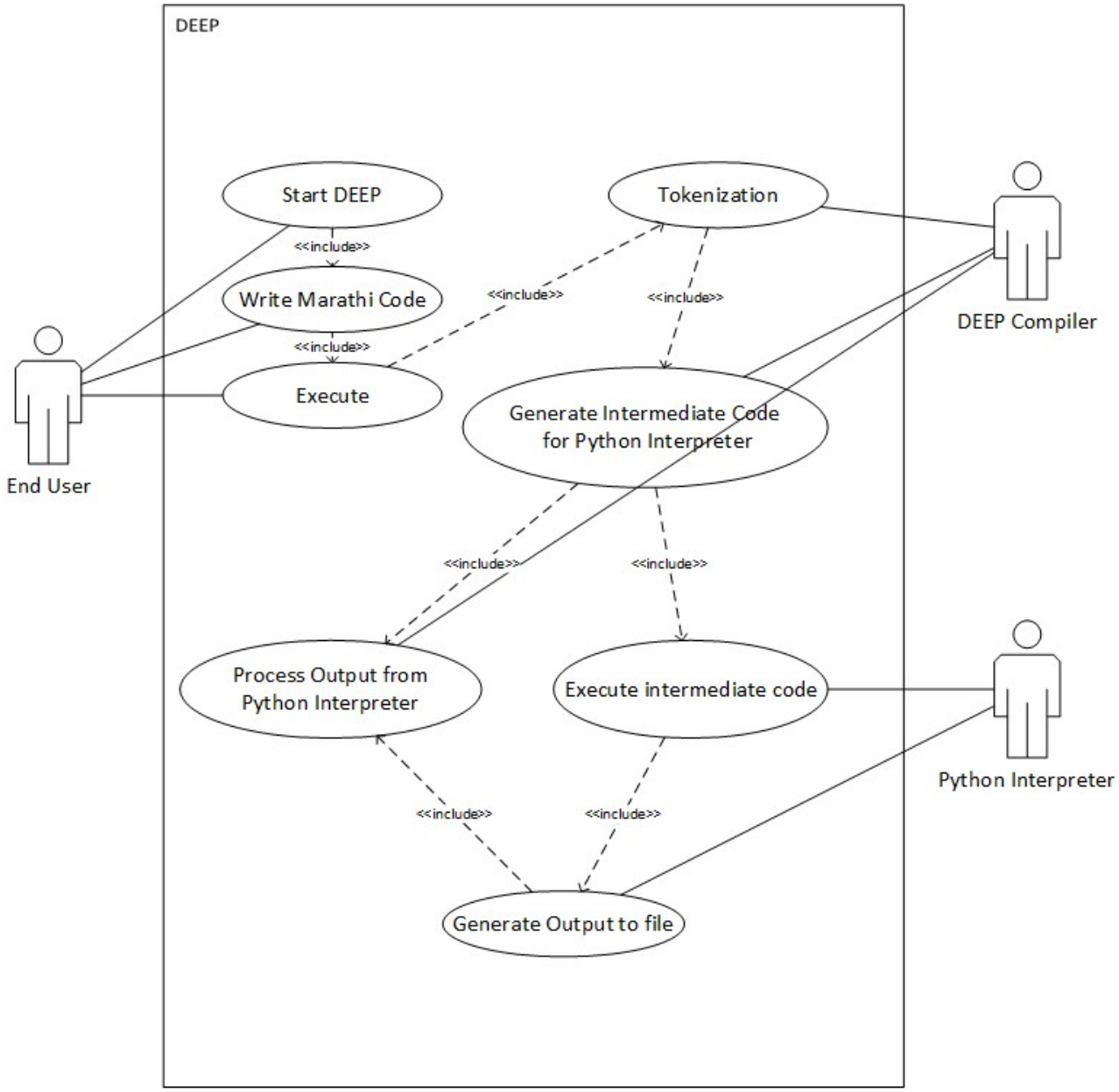


Figure 4.1: Use Case View

Activity Diagram:

Figure 4.2 shows the Activity diagram:

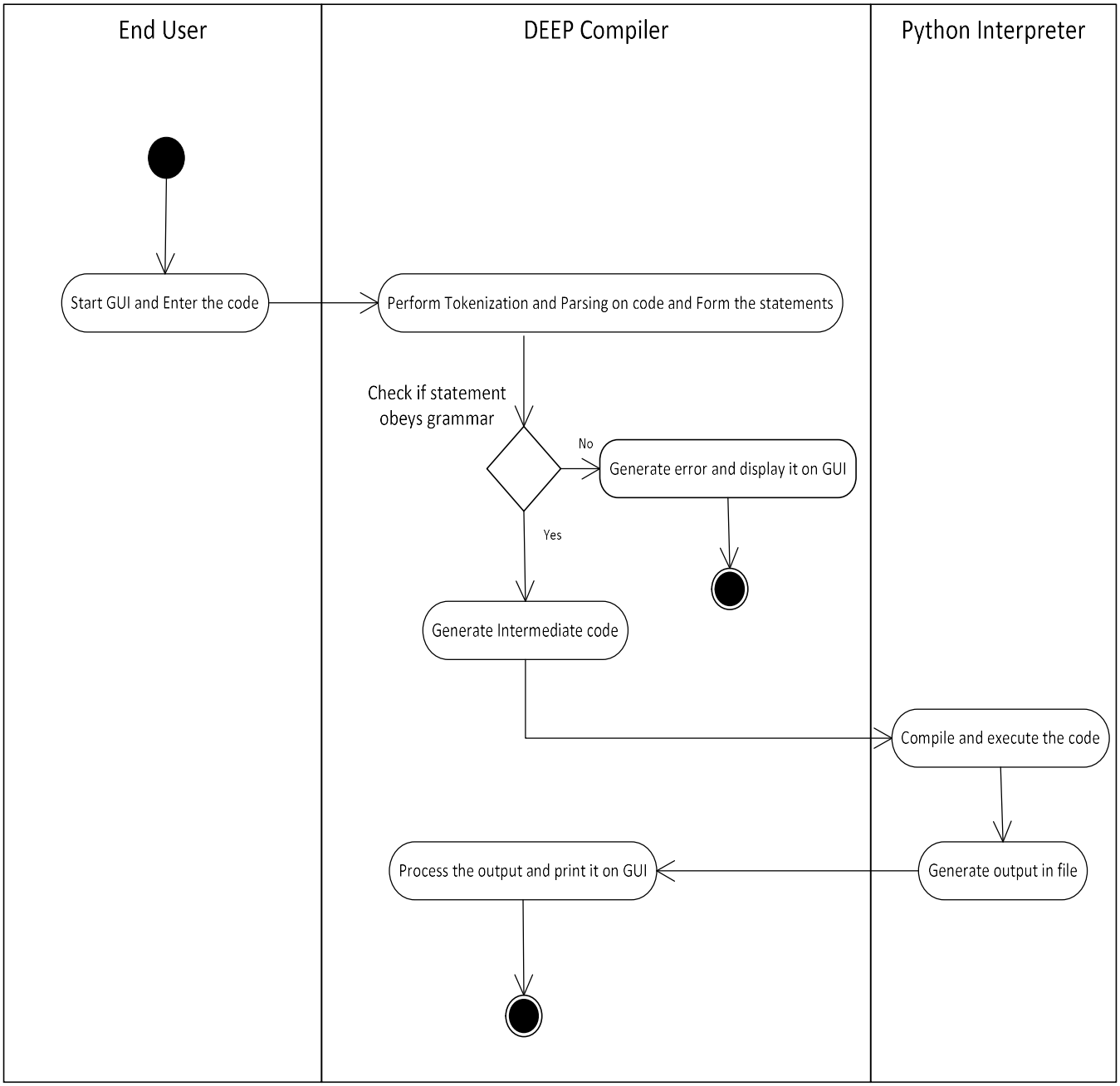


Figure 4.2: Activity Diagram

4.2 SPECIFIC REQUIREMENTS:

External Interface Requirements:

- User interface: Pyqt5 library for GUI, JetBrains PyCharm Community Edition.
- Hardware interface: PC/Desktop with python libraries and IBUS installed, Marathi Keyboard.
- Software interface: Ubuntu OS, Windows OS, PyCharm IDE.

Software Product Features:

- Product Perspective: DEEP enables user to code in Marathi language.
- Description of product features and functionalities:
    - Simple and understandable GUI.
    - Intermediate code is available.
    - Efficient error handling.
    - IBUS makes Marathi typing easy.
- Constraints:
    - DEEP supports few syntaxes.
    - Python 2.7 and PyQT5 must be installed on the system.
    - There must be a virtual keyboard available or IBUS installed to enable typing in Marathi.
- Assumptions and Dependencies:
    - Assumptions:
        1. People from Marathi background are unable to code in English and can type in Marathi efficiently.
        2. Logic developed by the user is correct.
        3. User have an idea about python syntaxes.
    - Dependencies:
        1. Successful output of the code depends on how well the logic is developed and how correct the program is typed.
        2. Working of DEEP GUI depends on its collaboration with the python interpreter & PyQt5 library.

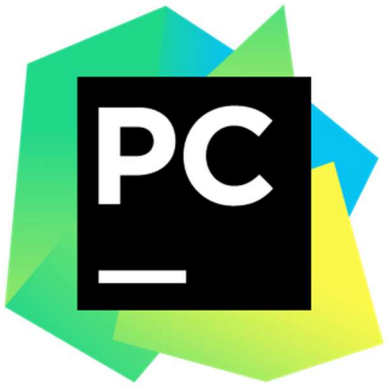
CHAPTER 5

PROJECT IMPLEMENTATION

5.1 TOOLS AND TECHNOLOGIES:

PyCharm IDE

DEEP has been developed in PyCharm using Python 2.7. PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License and there is also Professional Edition released under a proprietary license.



PyQT5

DEEP interface is developed using PyQT5. Qt is a set of C++ libraries and development tools that includes platform independent abstractions for graphical user interfaces, networking, threads, regular expressions, SQL databases, SVG, OpenGL, XML, user and application settings, positioning and location services, short range communications (NFC and Bluetooth) and access to the cloud. PyQt5 implements over 1000 of these classes as a set of Python modules.

PyQt5 supports the Windows, Linux, UNIX, Android, OS X and iOS platforms. PyQt does not include a copy of Qt. You must obtain a correctly licensed copy of Qt yourself. However, binary wheels of the GPL version of PyQt5 are provided and these include a copy of the appropriate parts of the LGPL version of Qt. PyQt5 comprises many different components.

First of all, there are a number of Python extension modules. These are all installed in the PyQt5 Python package.

PyParsing Library

DEEP includes grammar for various syntaxes. This grammar is developed using PyParsing library. The purpose of the pyparsing module is to give programmers using the Python programming language a tool for extracting information from structured textual data. The way that the pyparsing module works is to match patterns in the input text using a recursive descent parser: we write syntax productions and pyparsing provides a machine that matches the input text against those productions.

The pyparsing module works best when you can describe the exact syntactic structure of the text you are analyzing. A common application of pyparsing is the analysis of log files. Log file entries generally have a predictable structure including such fields as dates, IP addresses, and such. Possible applications of the module to natural language work are not addressed here.

IBus For Devanagari Typing

User can provide Marathi input to DEEP by installing ibus in the system. The intelligent Input Bus(IBus) is an input method framework for multilingual input in Unix-like operating systems. IBus is used to type in your own language in most GUI applications. Using IBus, user can type in Marathi language very easily by using an English keyboard. User should select Marathi – Phonetic (m17n) as input method to enable typing in Marathi using English keyboard.

5.2 IMPLEMENTATIONS:

This section of the report focuses on the methodology used in the implementation of the system. System works in six stages.

1. Input from the user

2. Tokenization

3. Keyword Handling

4. Execution of the intermediate code

5. Output processing

6. Error handling

Detailed description of the mentioned stages is given below:

Input from the user

DEEP first accepts the Marathi code input from the user. The user needs to take care of the syntaxes, indentation and all the rules required to write the

proper code. This input can be provided from a file. Devanagari characters in the form of Unicode are accepted as the input.

Tokenization

The input is split into several strings based on line breaks. The input statements are parsed using the grammar which is written with the help of PyParsing library. Then each of these strings is again broken down into tokens. Each token is then analyzed and handled accordingly Tokens are formed based on the grammar which is written with help of PyParsing library. These tokens are required for the conversion of the Marathi code into its English equivalent code. For example, if the input string is “c = a + b”, the tokens are formed from this string and stored as {c, =, a, +, b}.

Keyword Handling

The tokens are then compared with the predefined list of keywords which contains the English-Marathi keywords' pairs. If match found, the corresponding English keyword is printed into the intermediate code. If no match found, the token is searched into the variables' list. The keywords are defined to check whether the entered syntaxes follow the exact grammar required for the code to execute. The Marathi keywords used have almost the same meaning as the corresponding English keyword.

Execution of the Intermediate Code

The Intermediate Code generated is fed to the Python Compiler and the output generated is processed further. This Intermediate code can be provided to

the user if required. The Intermediate Code is in the form of the standard English Python Code.

Output Processing

The output generated by the Python Compiler is converted into corresponding Marathi language. This conversion is done with the help of the grammar written with the help of the PyParsing library. Thus, the output displayed to the user is also in Marathi language format which is understandable to him.

Error Handling

Syntax errors in the program are also displayed to the user using Marathi language representation. Thus, the user can understand the errors and correct them accordingly. Syntax errors are also in Marathi language form, thus enabling the user to understand the errors and carry out the required corrections.

CHAPTER 6

SCOPE OF PROJECT

PROJECT SCOPE

DEEP Currently support only selected syntaxes and selected errors handling. This scope is defined here. Supported functionalities & features of DEEP:

Syntaxes supported:

1. Mathematical Expressions
2. Assignment Expressions
3. Loops – For, While, Do-While
4. Conditional Expressions – If..Then..Else
5. Functions – Raw_input, Print

These syntaxes are in reference to Python 2.7 like programming language.

CHAPTER 7

SOFTWARE TESTING

7.1 INTRODUCTION

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the **process of validating & verifying** that an application works as expected

Black Box testing

Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is NOT known to the tester

White Box Testing

White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

7.2 TESTING & RESULTS:

Further pages consists of part of test cases & results.

Test Case ID	Test Case Title	Prerequisite	Steps	Input File (Expects input codes or file name of the codes attached)	Expected Output	Actual Output	Status
TC-01	Start DEEP	1.DEEP application is installed in a computer. 2.Python 2.7	1.Run DEEP application		DEEP application is running successfully.	DEEP application is running successfully.	Pass
TC-02	Writing Marathi program	IBus/virtual Marathi keyboard is installed.	1.Write program in Marathi language/provide file input		Marathi program is written successfully.	Marathi program is written successfully.	Pass
2.1	**Test Case for "Expressions"**			अ	चुकीची मांडणी + ओळ क्रमांक	चुकीची मांडणी	Partially Pass
2.2				अ =	चुकीची मांडणी + ओळ क्रमांक	चुकीची मांडणी	Partially Pass
2.3				अ = ब	Trace back (most recent call last): File "./out.py", line 2, in <module> a = b NameError: name 'b' is not defined	ब ची किंमत माहिती नाही. ओळ क्रमांक	Partially Pass

2.4				अ = ब +	Traceback (most recent call last): File "./out.py", line 2, in <module> a = b NameError: name 'b' is not defined	चुकीची मांडणी + ओळ क्रमांक	**FAIL**
2.5				अ = ब + क	Traceback (most recent call last): File "./out.py", line 2, in <module> a = b NameError: name 'b' is not defined	चुकीची मांडणी	Partially Pass
2.6				ब = १० क=१० अ=ब+क छापा अ	२०	२०	**Pass**
2.7				ब = १० क=१० अ = ब + क +* छापा अ	२०	चुकीची मांडणी	**FAIL**
2.8				ब = १० क=१० अ == ब छापा अ	चुकीची मांडणी	खरं	**FAIL**

2.8				You can also give here file name i.e. TC2_8.txt	चुकीची मांडणी	खरं	**FAIL**
2.9	**Test Case for "Print" function**			छापा	चुकीची मांडणी + ओळ क्रमांक	चुकीची मांडणी	Partially Pass
2.1				छापा अ	Traceback (most recent call last): File "./out.py", line 2, in <module> print a NameError: name 'a' is not defined	अ ची किंमत माहिती नाही. ओळ क्रमांक	Partially Pass
2.11				छापा १	१' displayed in output window	'१' displayed in output window	Pass
2.12				छापा १+२	३' displayed in output window	'३' displayed in output window	Pass
				छापा अ+२	Traceback (most recent call last): File "./out.py", line 2, in <module> print a + 2 NameError:	अ ची किंमत माहिती नाही. ओळ क्रमांक	Partially Pass

					name 'a' is not defined		
2.13				छापा अ+ब	Traceback (most recent call last): File "./out.py", line 2, in <module> print a + b NameError: name 'a' is not defined	अ ची किंमत माहिती नाही. ओळ क्रमांक	Partially Pass
2.14				छापा :	चुकीची मांडणी	चुकीची मांडणी	Pass
2.15				छापा "अ"	"अ" is displayed in output window	अ is displayed in output window	Pass
2.16				छापा ":"	":" is displayed in output window	: is displayed in output window	Pass
2.17	**Test Case for "IF" loop**			जर	चुकीची मांडणी	चुकीची मांडणी	Pass
2.18				जर अ	चुकीची मांडणी	चुकीची मांडणी	Pass
2.19				जर अ>ब	चुकीची मांडणी	चुकीची मांडणी	Pass

2.2				अ=10 ब=20 जर अ>ब	चुकीची मांडणी	चुकीची मांडणी	Pass
2.21				अ=10 ब=20 जर अ>ब छापा अ	चुकीची मांडणी	चुकीची मांडणी	Pass
2.22				अ=१० ब=२० जर अ>ब छापा अ नाहीतर छापा ब	चुकीची मांडणी	चुकीची मांडणी	Pass
2.23				जर अ;	चुकीची मांडणी	चुकीची मांडणी	Pass
2.24				अ=१० ब=१० जर अ==ब छापा अ	चुकीची मांडणी	चुकीची मांडणी	Pass
				अ=१० ब=३० जर (अ>ब): छापा अ नाहीतर: छापा ब	३०	३०	Pass

				अ=१० ब=३० क=१३ जर (अ>ब) आणि (अ>क): छापा अ किंवाजर (ब>अ) आणि (ब>क): छापा ब नाहीतर: छापा क	३०	चुकीची मांडणी	FAIL
2.25	**Test Case for "FOR" loop**		साठी	चुकीची मांडणी	चुकीची मांडणी	Pass	
2.26				साठी अ	चुकीची मांडणी	चुकीची मांडणी	Pass
				क = ० छापा क साठी फ़ आत मध्ये (०,५) : क = क + १ छापा क	० १ २ ३ ४ ५	० १ २ ३ ४ ५	Pass

				अ = ० ब = १ क = ० छापा अ छापा ब साठी फ़ आत मध्ये (०,५) : क = अ+ब अ=ब ब=क छापा क	० १ १ २ ३ ५ ८ ०	० १ १ २ ३ ५ ८ ०	Pass
2.27	**Test Case for "WHILE" loop**			जोपर्यंत अ	चुकीची मांडणी	Nothing is displayed in output window	FAIL
2.28				अ = ० जोपर्यंत अ< ५: छापा(अ) अ += १	Numbers from " ०" to "४" are displayed in output window	चुकीची मांडणी	FAIL
2.29	**Test Case for "RAW_INPUT" function**			छापा "तुझे नाव काय आहे ?" नाव = माहिती() छापा "नमस्कार %s!" % नाव	Input provided by user should be displayed in output window	तुझे नाव काय आहे ? नमस्कार %s!	FAIL

2.3				छापा "तुझे नाव काय आहे ?" नाव = माहिती() छापा नाव	Name provided by user should be displayed	तुझे नाव काय आहे ? None	FAIL
				अ=माहिती("पहिले नाव ?") ब = माहिती("आडनावं ?") क = अ+ब छापा क	पहिले नाव ? आडनावं ? शुभमहाळ्ळे	पहिले नाव ? आडनावं ? शुभमहाळ्ळे	Pass
2.31	**Test Case for "INT RAW_INPUT" function**			छापा "अंक?" नाव =माहितीक() छापा अंक	Number entered by user should be dispkayed in output window	अंक? None	FAIL

CHAPTER 8

WORKING RESULTS

This chapter consists of working results of the project.

Screen shots

1. Figure 8.1 shows screen shot of the GUI for programming in Marathi.

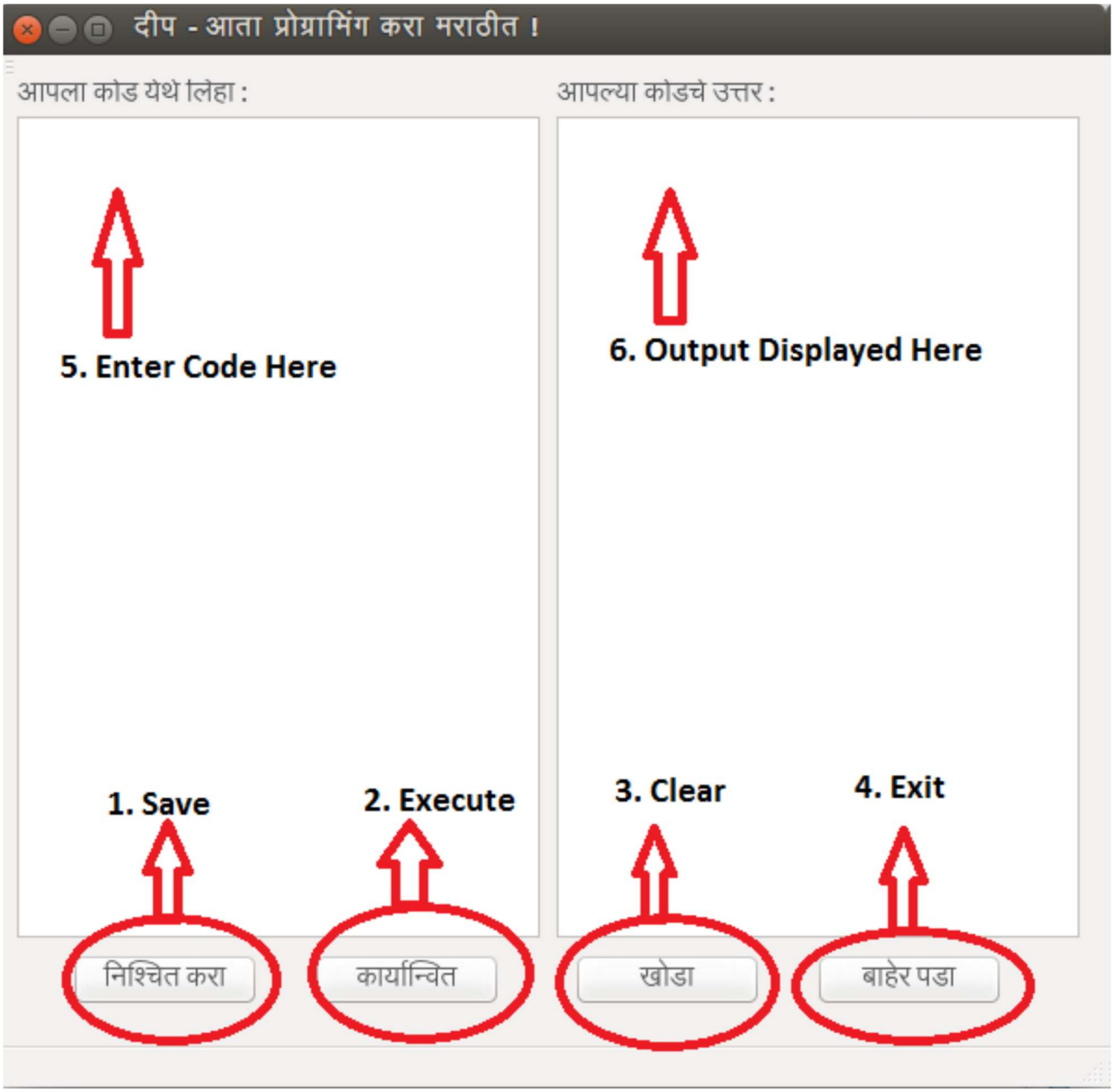


Figure 8.1: GUI for DEEP

2. Figure 8.2 shows screen shot of the working model of DEEP.

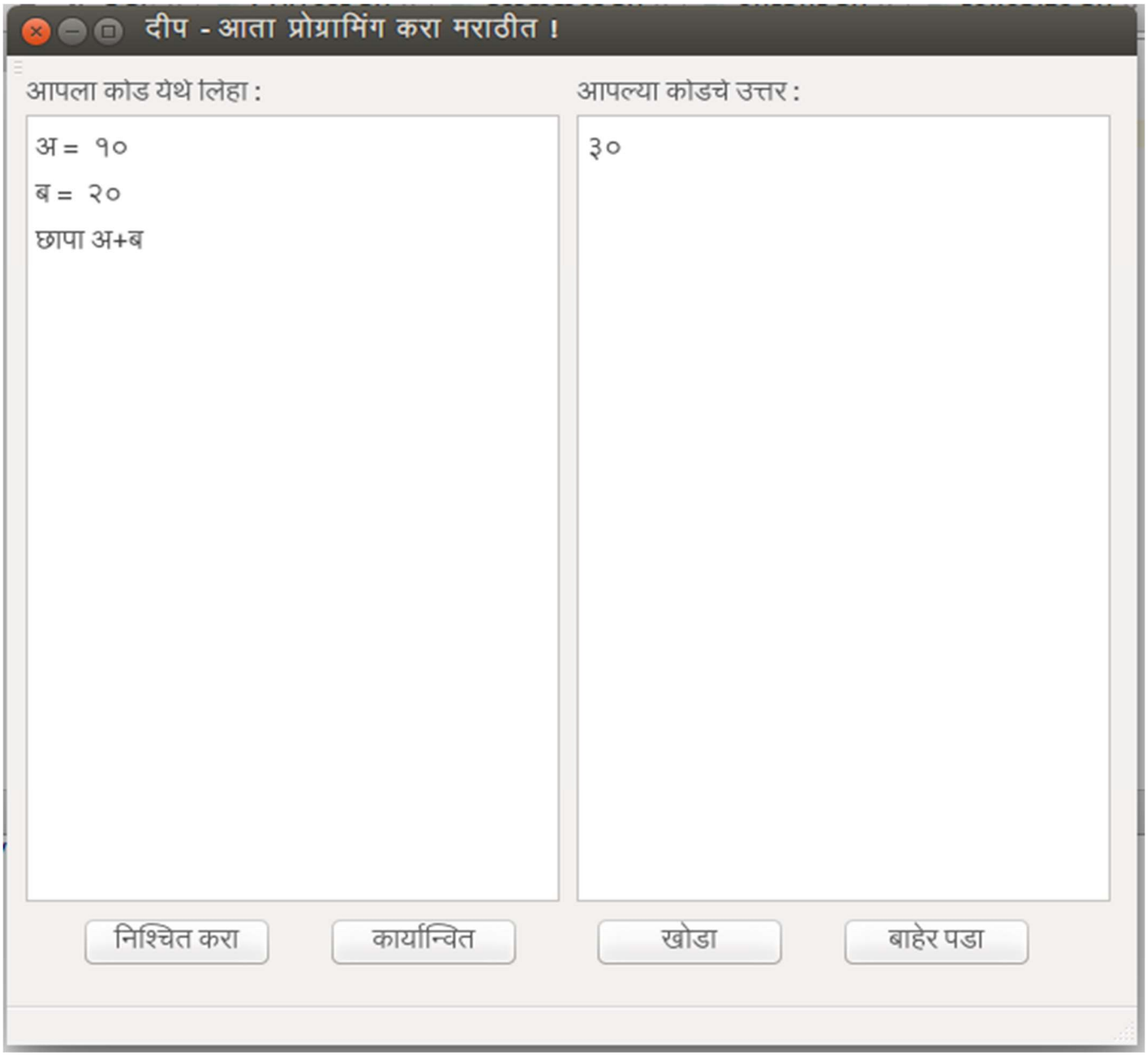


Figure 8.2: working model of DEEP

3. Figure 8.3 shows screenshot of the Error detection in DEEP.

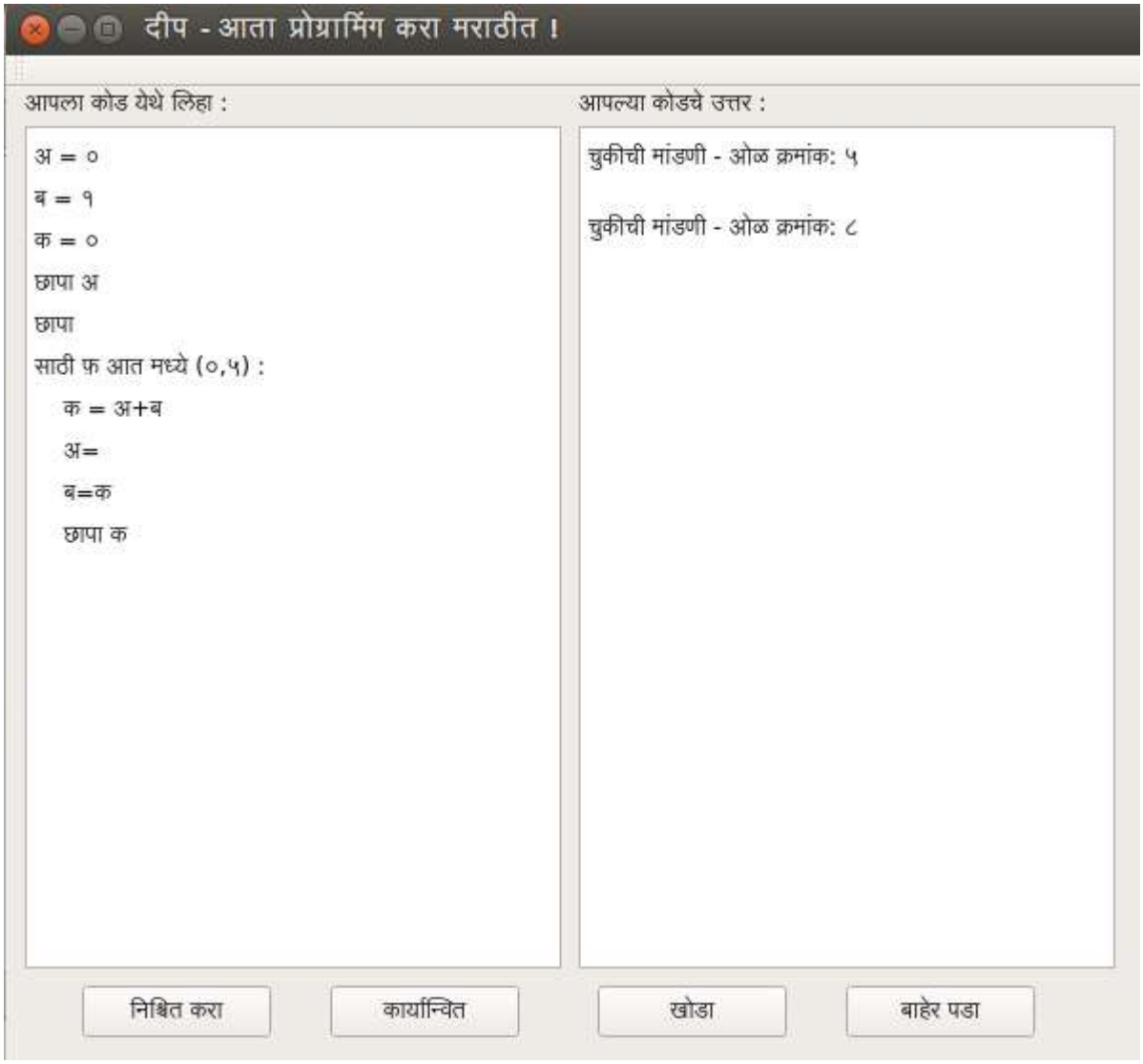


Figure 8.3: Error detection in DEEP.

CHAPTER 9

DEPLOYMENT & MAINTENANCE

9.1 DEPLOYMENT

Deployment task of DEEP is somewhat tedious process for absolute beginner. We need to have person comfortable with working on installations on Linux systems to work for deployment of DEEP. We will have to take up on this issue & make DEEP for distributable to masses.

However, the deployment procedure for DEEP follows following steps in overall.

1. Installation of Python 2.7
2. Installation of PyParsing library
3. Installation of Python Development package
4. Installation of PyQt5 libraries (along with Qt & sip libraries)
5. Distributable sources of DEEP

1. Installation of Python 2.7

Python being open source language, is available for all. Detailed installation guide for Python 2.7 can be found here: https://www.python.org/downloads/

2. Installation of PyParsing library

This is available for download at: https://pypi.python.org/pypi/pyparsing

3. Installation of Python Development package

To install Python Development package, follow following commands for terminal:

```
sudo apt-get install python-dev  # for python2.x installs
sudo apt-get install python3-dev  # for python3.x installs
```

4. Installation of PyQt5 libraries (along with Qt & sip libraries)

PyQt5 needs to be installed for GUI usage, before installing PyQt, one must install Qt on the computer.

Download Qt: https://www.qt.io/

Download PyQt5: https://riverbankcomputing.com/software/pyqt/intro

5. Distributable sources of DEEP

These sources can be downloaded (forked) from our Git Repository:

https://github.com/halleshubham/DEEP

9.2 MAINTENANCE

No special maintenance is required, but to download most recent versions of DEEP from our Git Repo is regular update task.

9.3 USER MANUAL

Simple user manual attached with report at **Annexure x** is also provided to user, in both English & Marathi Languages, primary focus of which is to able to use & program using DEEP.

CHAPTER 10

DESIGN

10.1 Data Flow Diagrams (Level 0 & Level 1):

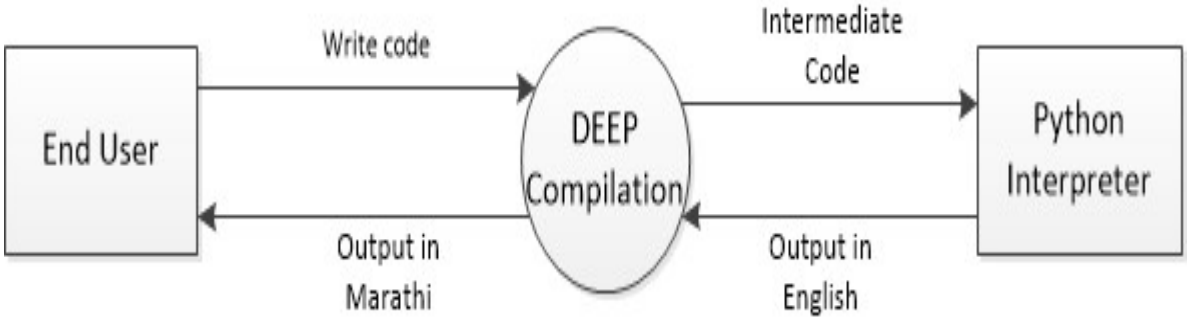


Figure 10.1.1: Data Flow Diagram : Level 0

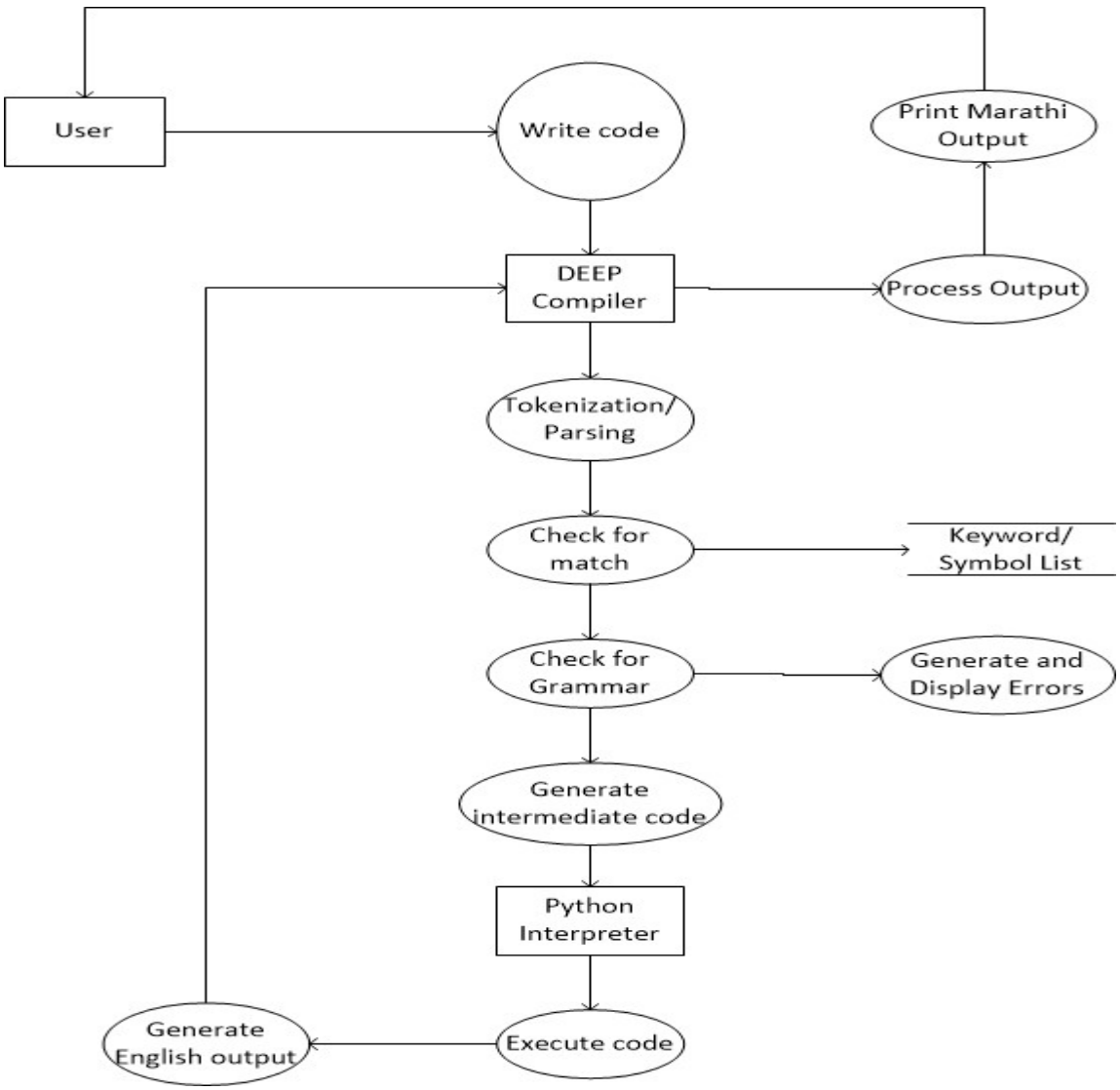


Figure 10.1.2: Data Flow Diagram : Level 1

10.2 UML DIAGRAMS

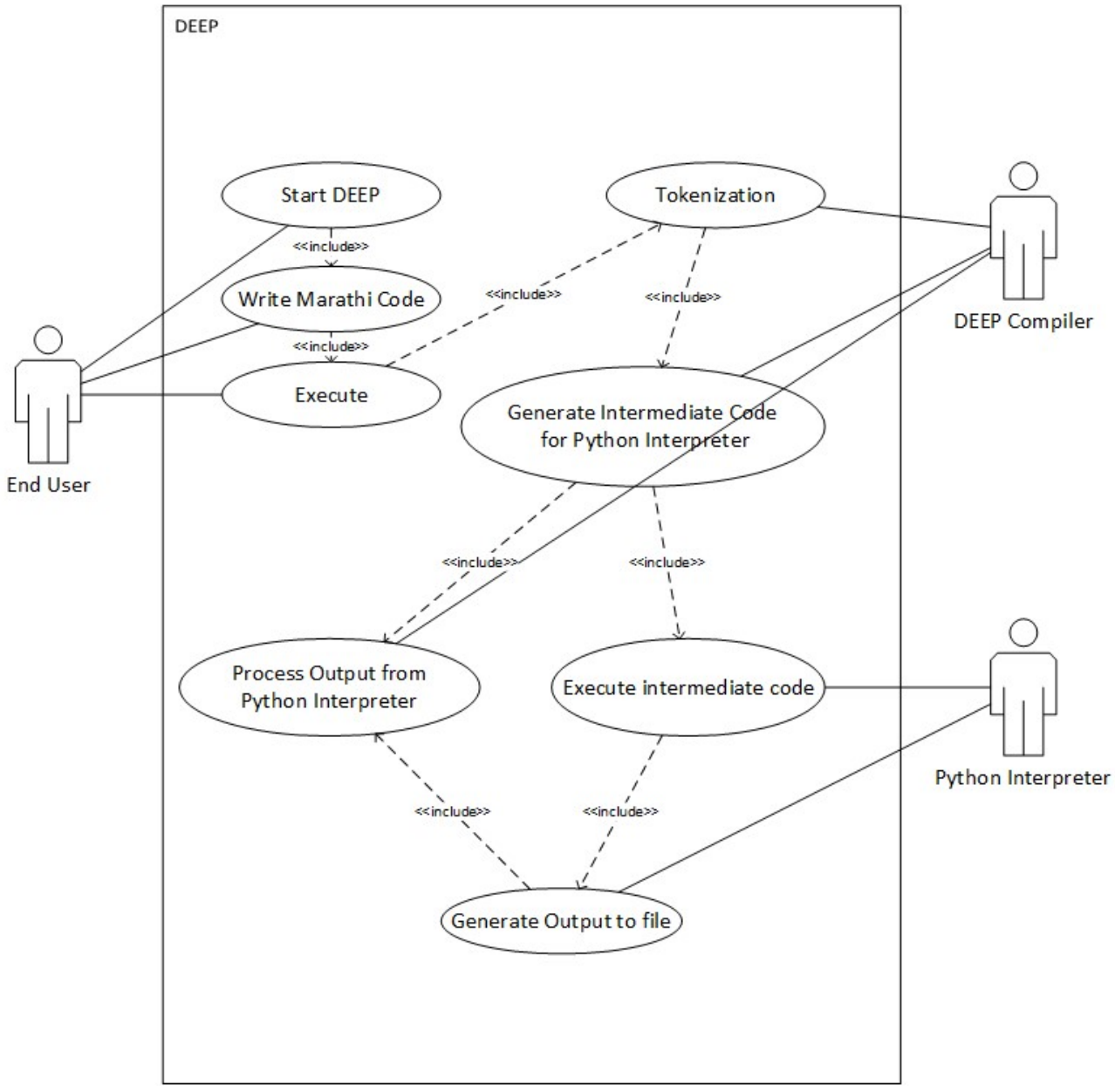


Figure 10.2.1: Use Case View

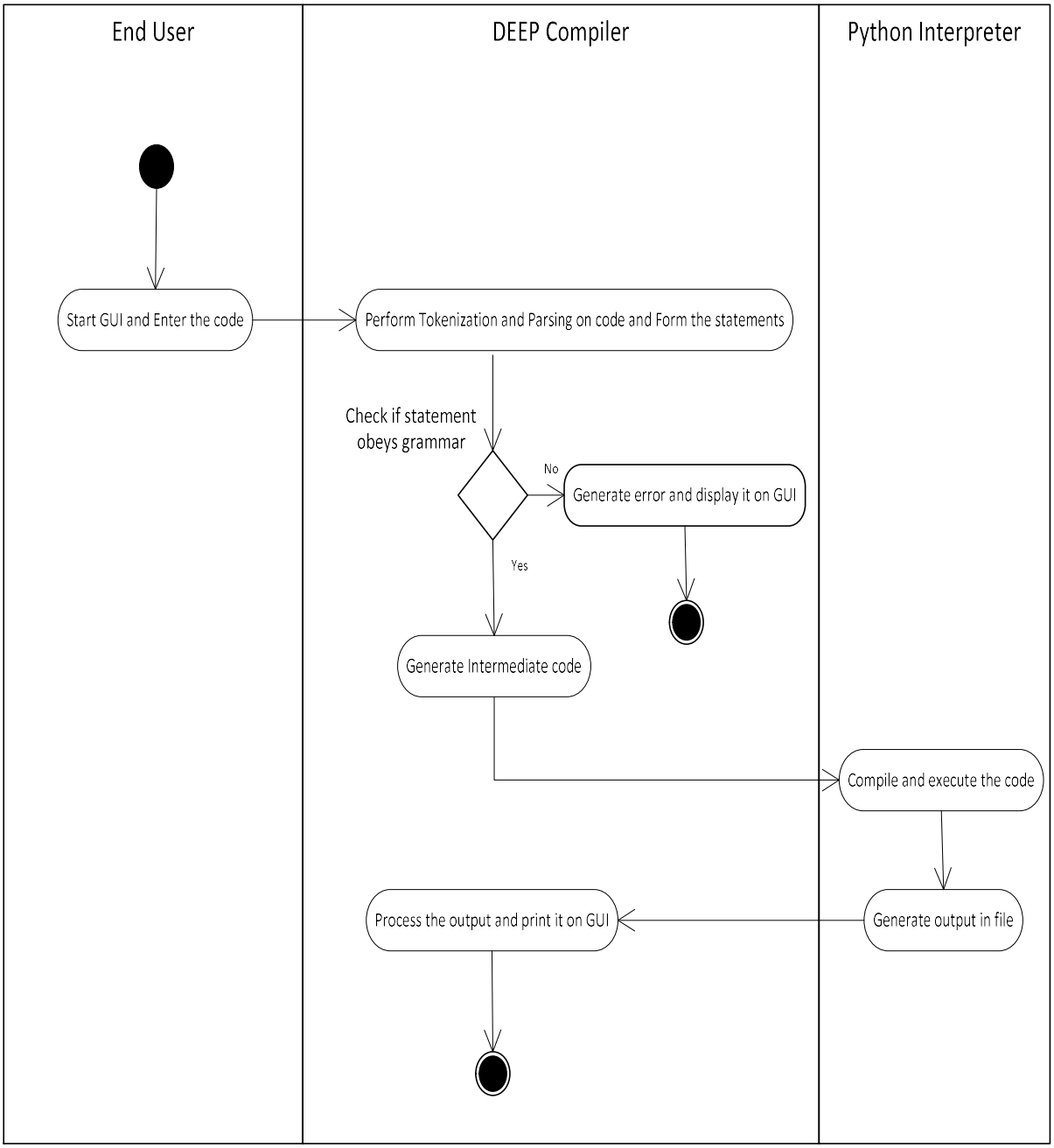


Figure 10.2.2: Activity Diagram

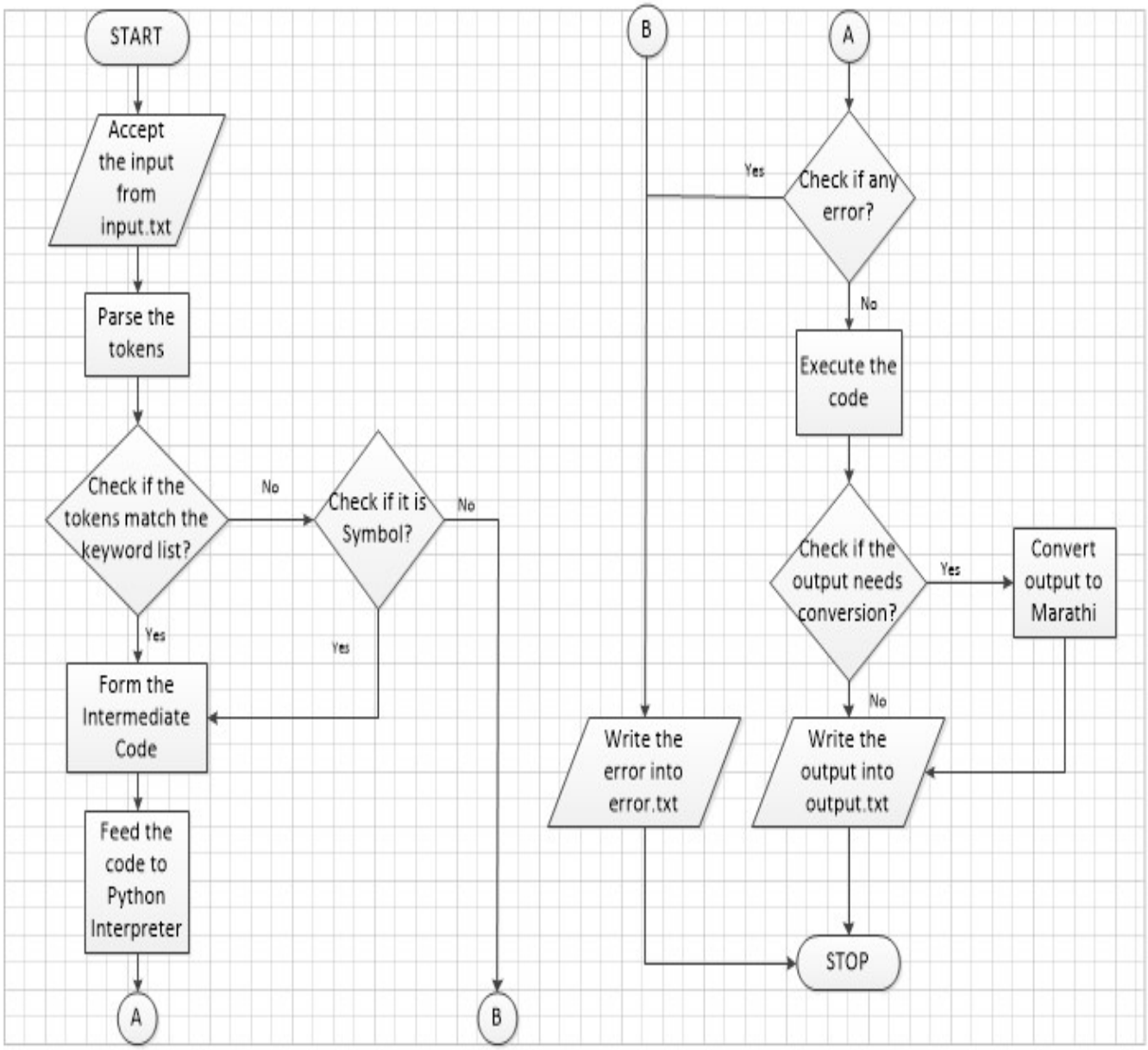


Figure 10.2.3: Flow of the System

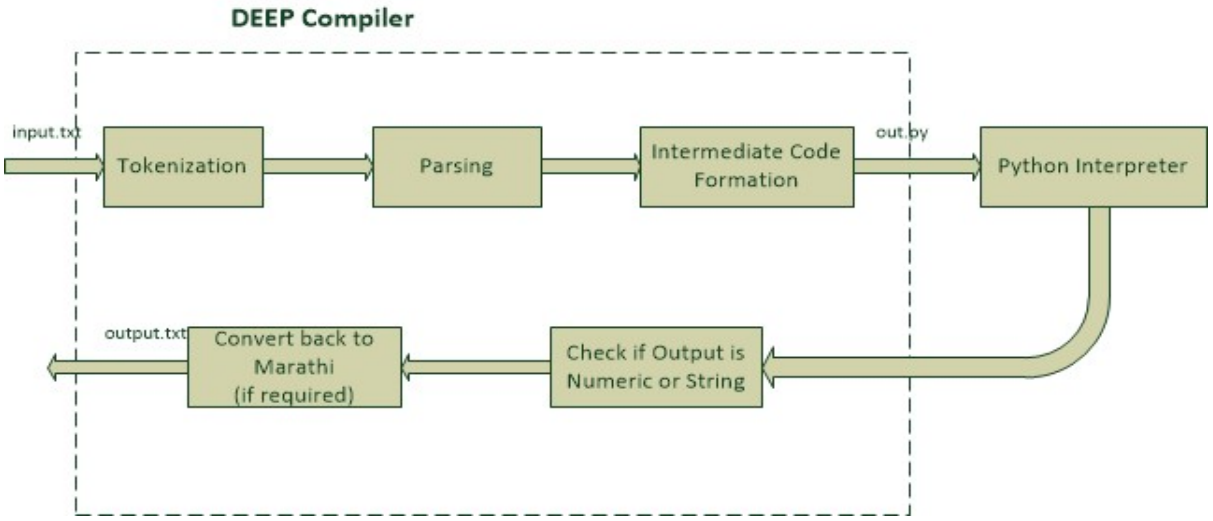


Figure 10.2.4: Architecture Diagram

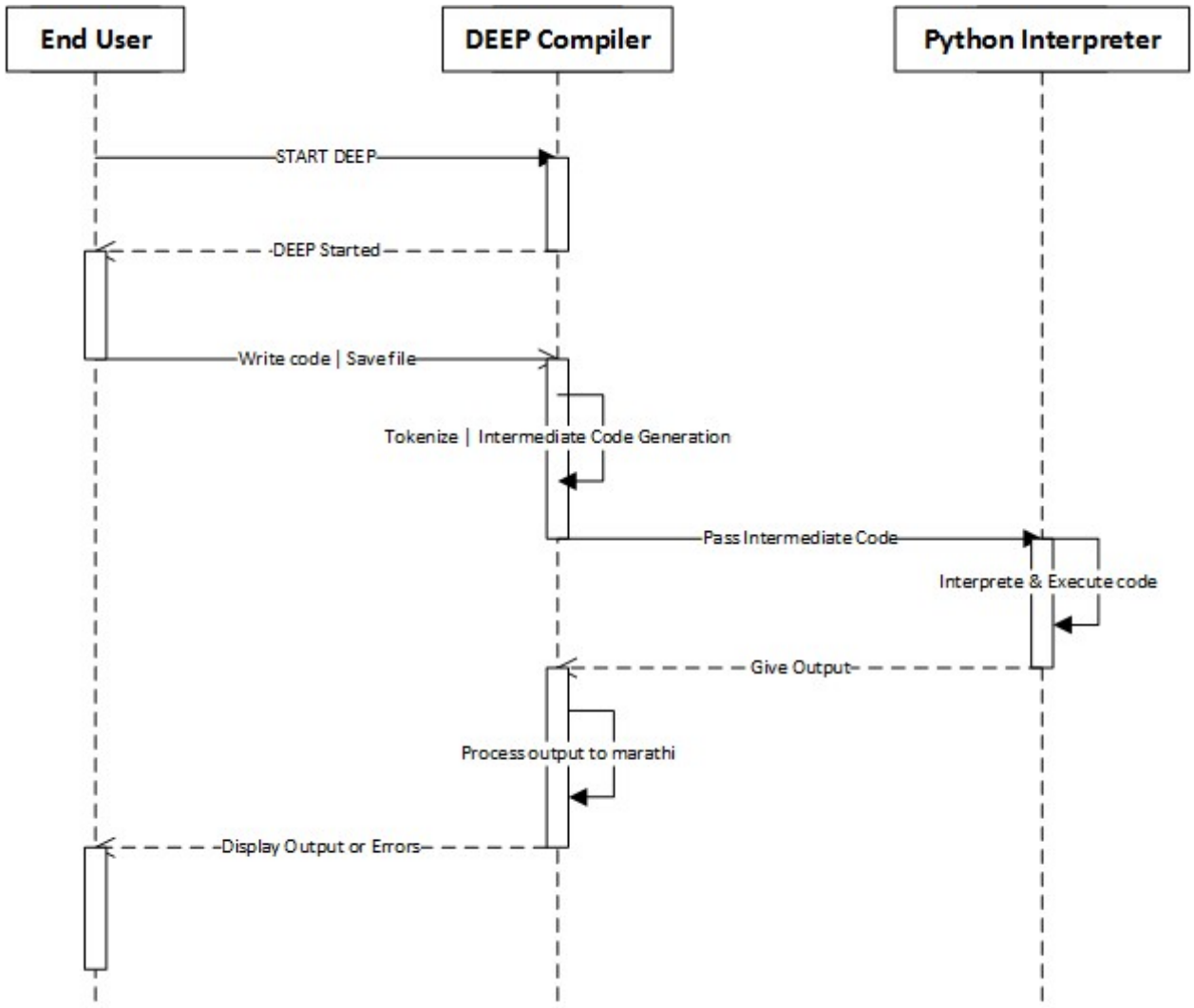


Figure 10.2.5: Sequence Diagram

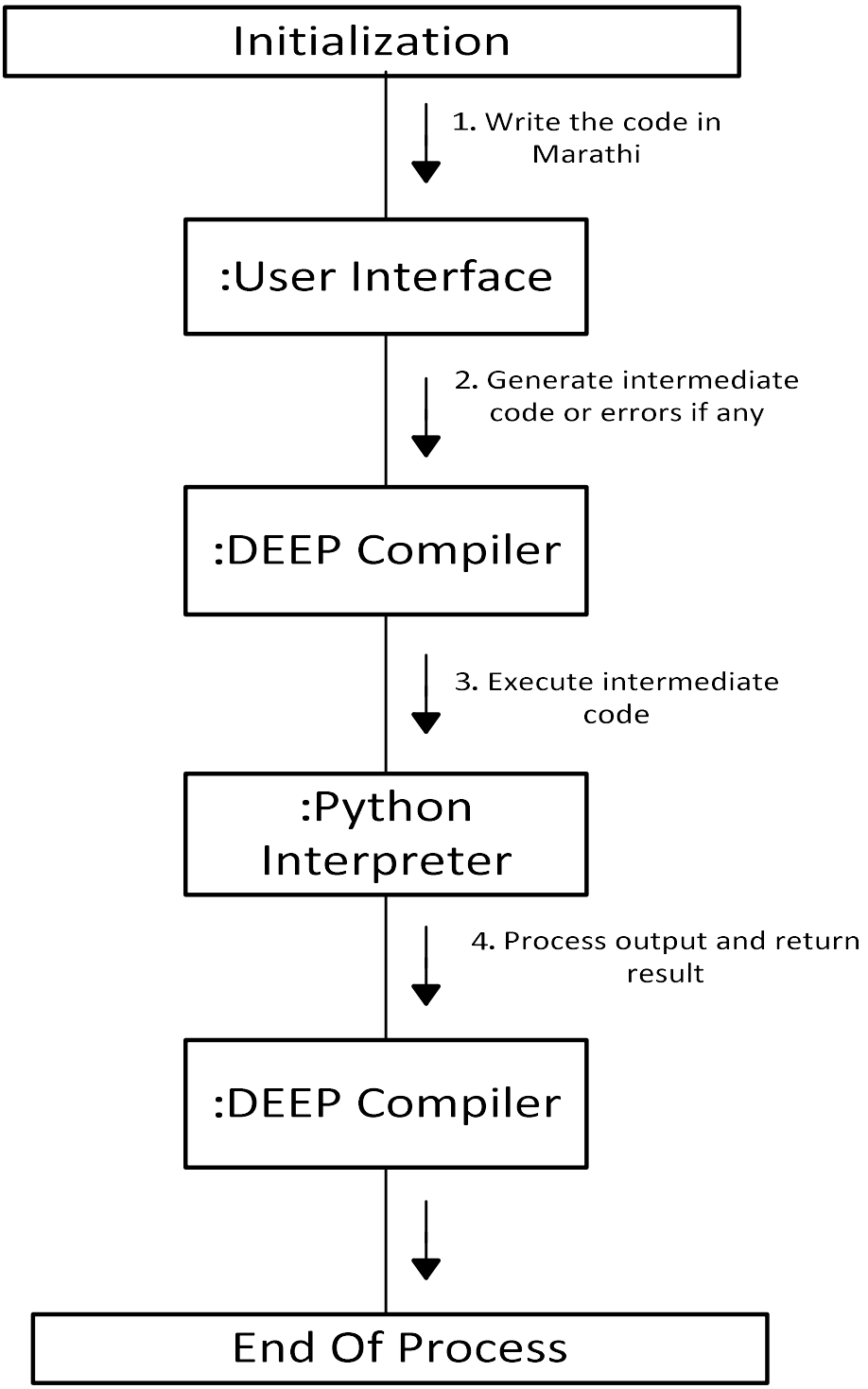


Figure 10.2.6: Collaboration Diagram

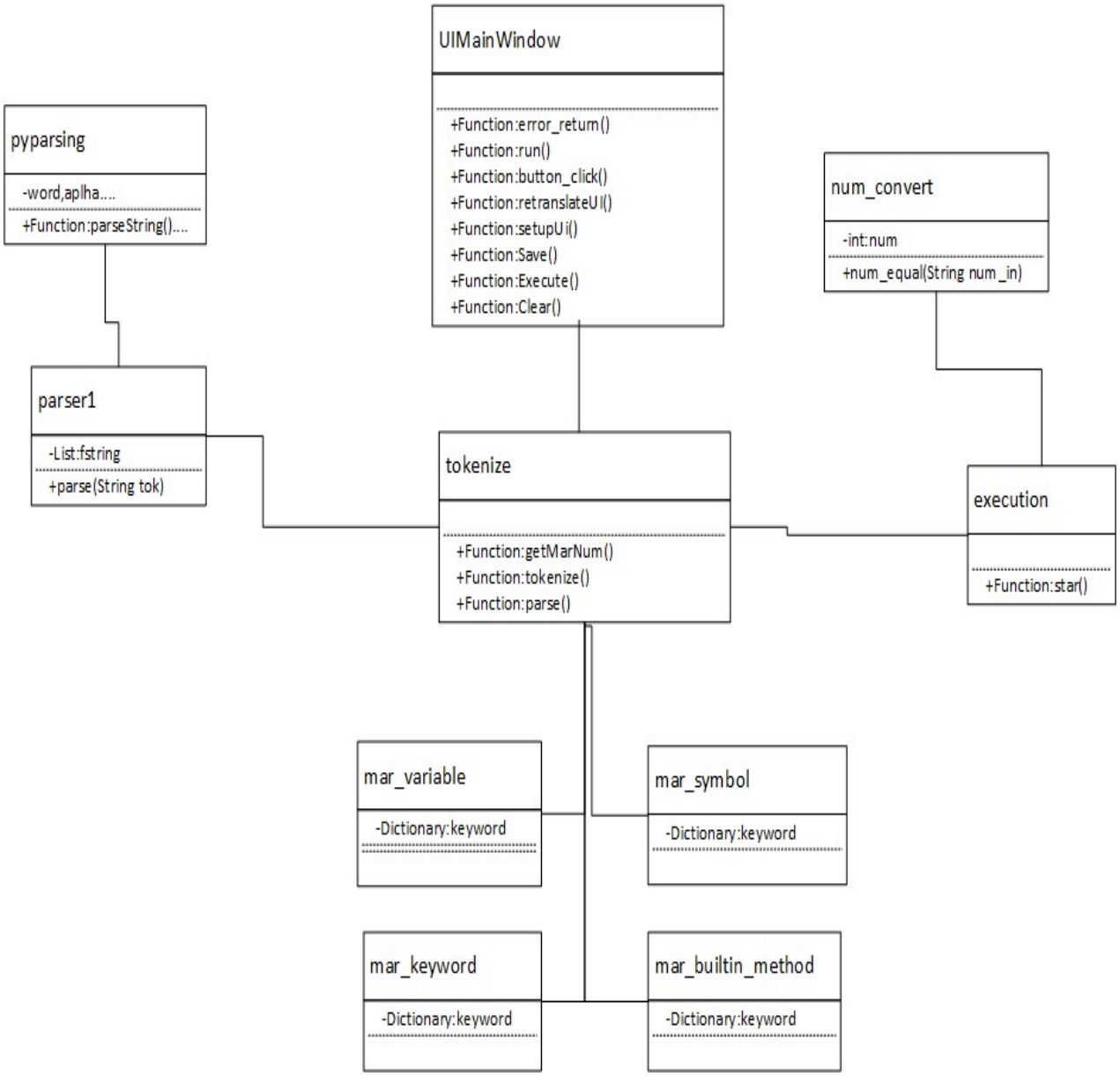


Figure 10.2.7: Class Diagram

CHAPTER 11

PLANNING & SCHEDULING

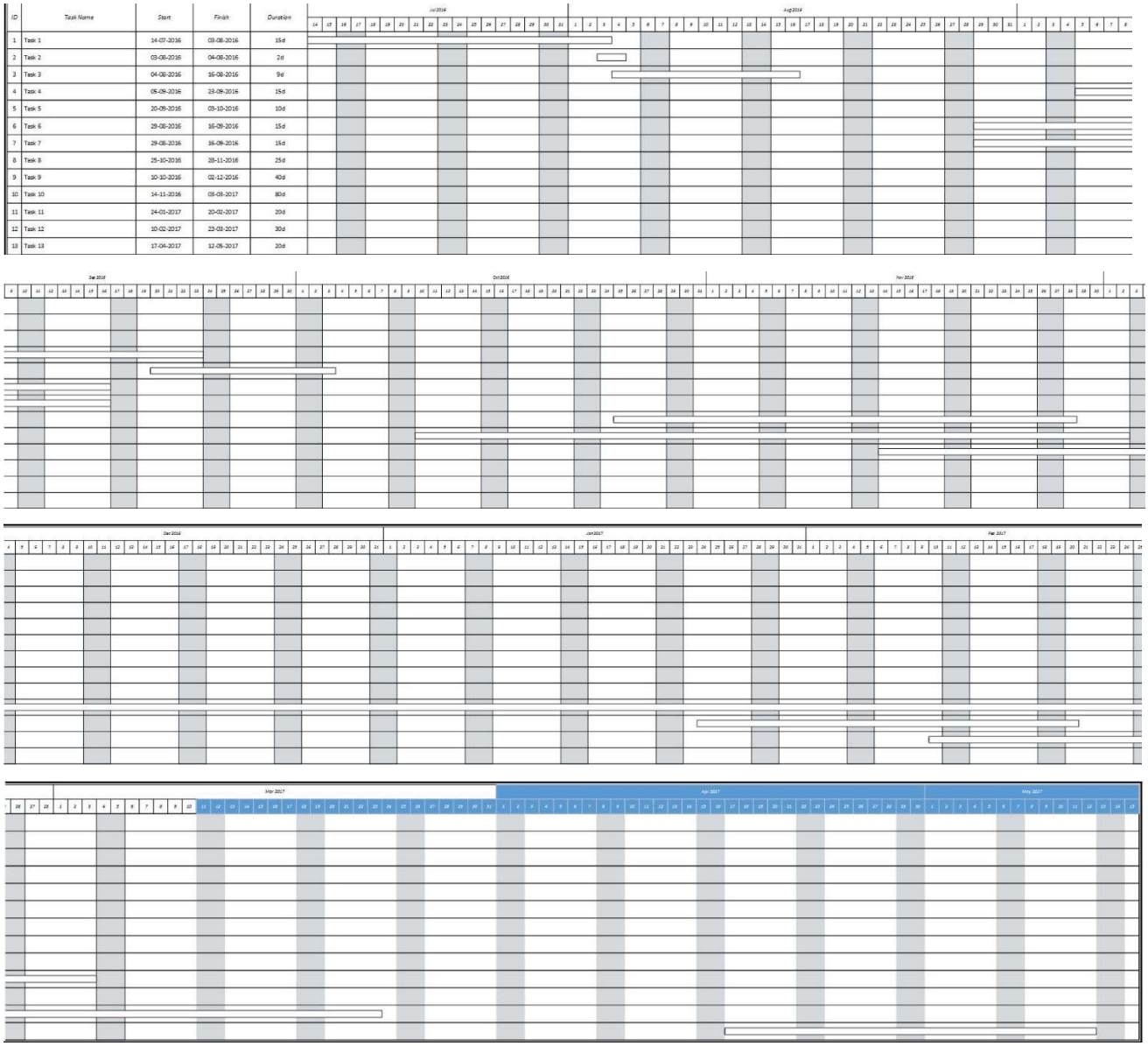
Project Planning & Scheduling

11.1 Project plan

Table 11.1. Project plan table

Sr. No	Task	Description	Duration (Days)
1	Domain selection	Deciding the domain as Interpreter for the project	15
2	Searching key terms	Searching key terms related to the domain	2
3	Deciding topic for project	Selecting DEEP as project topic	12
4	Literature survey	Studied Papers related to non-English based programming languages.	20
5	Problem statement	Making the problem statement	10
6	Synopsis formation	Forming the synopsis	15
7	Studying existing similar projects	Methodologies and drawbacks of existing projects was studied	15
8	System Modelling	For the system model of project	25
9	Study of Python	Study of Python and its existing libraries	40
10	Proposed system	Implementation of DEEP	80
11	Testing	Going through various types of testing	20
12	Data Analysis	Experiment of DEEP with intended audience	30
13	Report generation	Report preparation	20

11.1.1 Project Schedule



ID	Task Name	Start	Finish	Duration
1	Task 1	14-07-2016	03-08-2016	15d
2	Task 2	03-08-2016	04-08-2016	2d
3	Task 3	04-08-2016	16-08-2016	9d
4	Task 4	05-08-2016	23-09-2016	15d
5	Task 5	20-09-2016	03-10-2016	10d
6	Task 6	29-08-2016	16-09-2016	15d
7	Task 7	29-08-2016	16-09-2016	15d
8	Task 8	25-10-2016	28-11-2016	25d
9	Task 9	10-10-2016	02-12-2016	40d
10	Task 10	14-11-2016	08-03-2017	80d
11	Task 11	24-01-2017	20-02-2017	20d
12	Task 12	10-02-2017	23-03-2017	30d
13	Task 13	17-04-2017	12-06-2017	20d

Fig 11.1.1 Project Schedule

A work plan is a complete accounting of how a person or grouping proposes going about accomplishing a specific task, approaching a project. Proposed work plan generally includes an introduction or overview of a project or job, a breakdown of how individual project-related tasks will be accomplished, a timeline for completion and cost projections for implementation. The above figure 4.6 shows Gantt chart of proposed plan work this chart explain the plan of work there are total eight task. Task 1 is study IEEE paper in 30 days, task 2 signify design of

algorithm, task 3 represent design of GUI, task 4 represent pre-processing, task 5 signify the coding, task 6 represent Implementation it require 60 days to complete a work, task 7 correspond to Integrating modules and testing and task 8 is documentation.

REFERENCES

[1] Arman Kamal, Md. Nuruddin Mon-sur, Syed Tanveer Jishan and Nova Ahmed, 20- 22 December, 2014, "*ChaScript: Breaking Language Barrier using a Bengali Programming System*", Dhaka, Bangladesh W.-K. Chen, *Linear Networks and Systems*. Belmont, Calif.: Wadsworth, pp. 123-135.

[2] Susumu Kanemune, Takako Nakatani, Rie Mitarai, Shingo Fukui, Yasushi Kuno, July 2004, "*Dolittle — Experiences in Teaching Programming at K12 Schools*", Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing (C5'04).

[3] Rezvan Noormohamadi, 2008, "*Mother Tongue, a Necessary Step to Intellectual Development*", Journal of Pan-Pacific Association of Applied Linguistics, 12(2), 25-36.

[4] Dann G. Mallet, 2011, "*Walking a mile in their shoes: Non-native English speakers' difficulties in English language mathematics classrooms*", Journal of Learning Design, Vol. 4 No. 3, 28-34.

[5] TECHWELKIN, 2011, "*Hindawi: Write Computer Programs in Indic Languages*", Available at:
<http://techwelkin.com>, [Accessed 15 January 2017].

[6] UN News Centre, 2015, "*On Mother Language Day, UN spotlights role of native tongue in education*", Available at: <http://www.un.org/apps/news/printnewsAr.asp?nid=50147>, [Accessed 5 January 2017].

[7] Ruslan's Blog, *Let's Build A Simple Interpreter. Part 11.* https://ruslanspivak.com/

CHAPTER 13

ANNEXURES

13.A GROUP MEMBER's PROFILES

1. Name: Mayur Jagtap

2. Date of Birth: 15/03/1995

3. Gender: Male

4. Permanent Address:

c/o S.B.Jagtap, Rajlaxmi Residency,

Chikhli Road, Deulgaon Raja - 443204

5. E-Mail: jagtap.mayur1503@gmail.com

6. Mobile/Contact No. : 7841915239

7. Placement Details: Accenture

---

1. Name: Ruchita Kolte

2. Date of Birth: 07/07/1995

3. Gender: Female

4. Permanent Address:

28, Mahabal Road, Sambhaji Nagar, Jalgaon - 425001

5. E-Mail: rucha.kolte77@gmail.com

6. Mobile/Contact No. : 8983798360

7. Placement Details: Cognizant Technology Solutions

1. Name: Nikita Babhale

2. Date of Birth: 18/04/1995

3. Gender: Female

4. Permanent Address:

1856, Mahendra Nagar, Karmala Pin 413203

5. E-Mail: nikitababhale@gmail.com

6. Mobile/Contact No. : 8275448049

7. Placement Details: Infosys

---

1. Name: Shubham Halle

2. Date of Birth: 28/10/1995

3. Gender: Male

4. Permanent Address:

25/B-21, Akurli Ratnadeep, MHADA, Kandivali East,

Mumbai 400101

5. E-Mail: halleshubham@gmail.com

6. Mobile/Contact No. : 9967375492

7. Placement Details: Cognizant Technology Solutions

13. B USER MANUAL – ENGLISH

DEEP – DEvnagari Enabled Programming

(USER MANUAL)

A Project By:

Shubham Ashok Halle

Mayur Sandip Jagtap

Ruchita Sanjiv Kolte

Nikita Sudhir Babhale

Guided By:

Prof. Aditi S. Jahagirdar

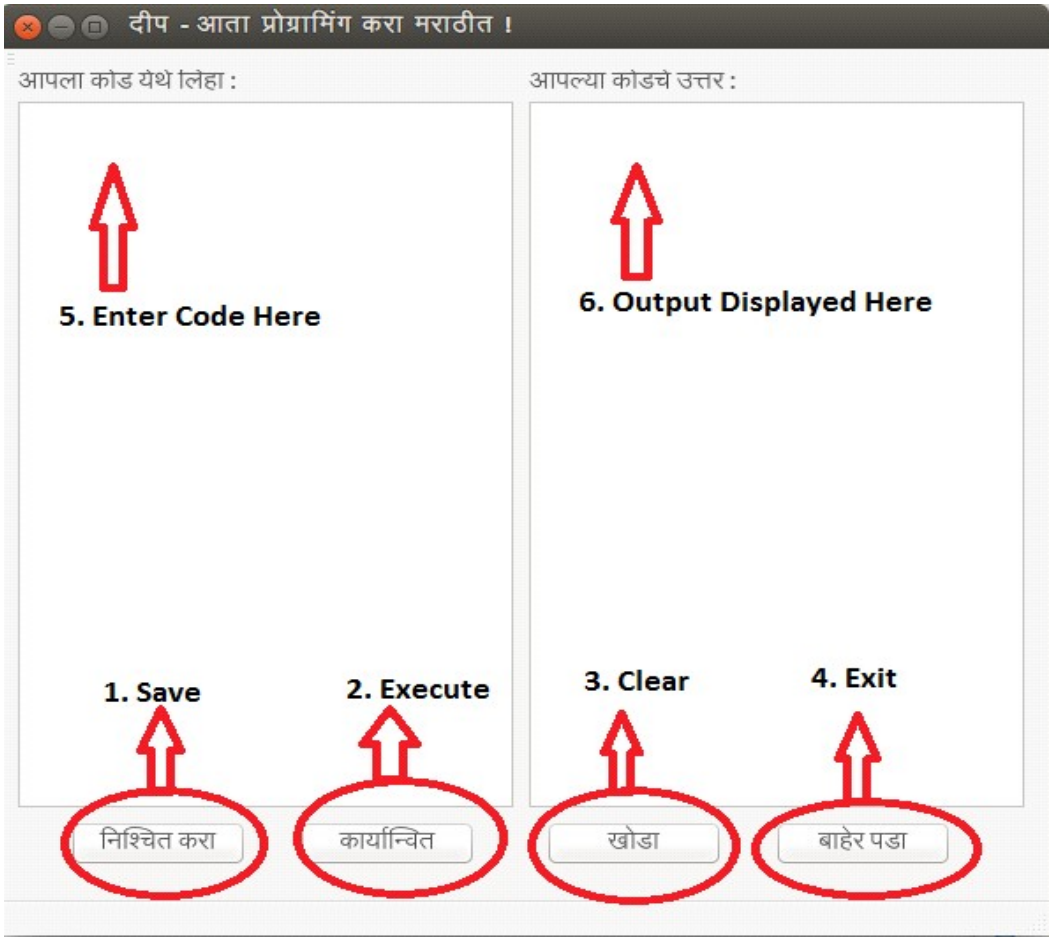
Table Of Contents

1. Introduction to DEEP

Mother tongue has central role in education that demands cognitive development. DEEP is for those people who are very comfortable with Marathi but find it difficult to handle English. DEEP is not a new Programming Language but a Software which will allow the users to write the Python program in Marathi. DEEP works on top of the Python 2.7.

It has a GUI which is quite user-friendly and has all the instructions in Marathi. Thus, it enables the user to learn and implement coding easily.

2. Understanding the DEEP GUI



1. Save Button

Once the user is done writing the program, he has to click this button to tell DEEP that he is done typing and now wants to save the program for further execution.

2. Execute Button

This button has to be pressed after saving the program. It executes the program and displays the result. Any errors if present in the program also get displayed after pressing this button.

3. Clear Button

Pressing this button will clear the contents of both the input and output area.

4. Exit Button

Pressing this button closes the GUI.

5. Enter the code here

The Marathi code is entered here and then the Save button is pressed.

6. Output displayed here

The output of the code is displayed here. If errors are present, they also get displayed in this area.

3. The Control Flow Statements

3.1 *for* Statements

Python's *for* statement iterates over the items of any sequence (a list or a string), in the order that they appear in the sequence.

Syntax: साठी <variable name> आत मध्ये (range):

Example: साठी अ आत मध्ये (०, १०):

3.2 *while* Loops

Python's *while* statement iterates over all the statements inside the loop as long as the condition specified is true. Syntax: जोपर्यंत(condition):

Example: जोपर्यंत (अ<१०):

3.3 *if* Statements

An *elif, else* statement can be combined with an *if* statement. An *else* statement contains the block of code that executes if the conditional expression in the *if* , *elif* statement resolves to 0 or a FALSE value.

Syntax: जर (condition):

किंवाजर (condition):

नाहीतर:

Example: जर (अ<१००):

किंवाजर (अ>१०००):

नाहीतर:

4. Functions Supported

4.1 *raw_input* Function:

It presents a prompt to the user, gets input from the user and returns the data input by the user in a string. In DEEP, the input values to be given to the program need to be entered in a file at first then the program gets executed.

A. For giving integer input

Syntax: संख्या (माहिती ("message for the user"))

Example: संख्या (माहिती ("कृपया अ ची किंमत टाका"))

B. For giving string input

Syntax: ओळ (माहिती ("message for the user"))

Example: ओळ (माहिती ("कृपया अ ची किंमत टाका"))

4.2 *print* Function:

It is used when we want to display something to the user.

Syntax: छापा <variable name>/<any message>

Example: छापा ("नमस्कार")

5. Errors And Exceptions

5.1 Syntax Error:

When there is some syntactical error in the code, the syntax error gets displayed.

Error: चुकीची मांडणी

5.2 Unexpected Error:

When there is any other error other than the syntax error, this error gets displayed to the user.

Error: अनपेक्षित चुक

------ END OF USER MANUAL – ENGLISH ------