Minor Project Report

On

Developer Assistant

Submitted by:

Name	Roll No	Branch
Sarthak Jain	R171218088	CSE-DevOps
Siddharth Chopra	R178218045	CSE-MAD
Manvi Kulshrestha	R178218022	CSE-MAD

Under the guidance of

Dr. Hitesh Kumar Sharma

Associate Professor

Department of Cybernetics

School of Computer Science and Engineering

University of Petroleum & Energy Studies

Dehradun - 248001

2021

School of Computer Science and Engineering

University of Petroleum & Energy Studies

Dehradun - 248001

2021

Project Proposal Approval Form (2021)

Minor II

ABSTRACT

As technology continues to grow and expand, programmers or developers need to keep up. Frameworks, tools, libraries, and commands become outdated pretty quickly. Usually between two to three years, updated versions come along.

In a sense, updated versions are good, because they are more efficient and make jobs easier. But you also need to get used to them fast, something you may struggle with as a new developer.[1]

Veteran developers know, CLI commands and using quality tools can make them more productive and can also open doors to various approaches to automation that are far more practical with textual interfaces than with GUI apps.

Performing the same operations manually introduces both an added cognitive load and the increased possibility of human error. As usual, we rely on computers to handle tasks humans may find boring, repetitive, or overwhelming.[2]

And such a task is learning commands and implementing them repeatedly which is boring, so here we came with an assistant just like a google assistant that we've named Developer Assistant. It will perform all tasks like configuring, updating or launching services or applications in one go through text or voice inputs, thus helping developers.

Keywords: learning commands, Google Assistant, configure, update, launch, text or voice input.

INDEX

1. INTRODUCTION

In almost all cases, the command line is much faster and easier to use for an expert, but not for a beginner. For beginners, this is synonymous with learning a new language to speak, and if you do not use it you lose it. This is because of its learning curve to remember the commands in which accuracy has the utmost importance. Like even if there is a spelling error, a command will fail. Also, if an instruction is mistyped, people often need to start from scratch. Typing the wrong commands, could accidentally delete files or close the program before you saved your work. This is considered to be the main disadvantage of using a command line that is strictly text-based.[4] Also, some things are hard or annoying to do on the command line, like things that require a lot of typing, such as long file names, that too without minimal cut/paste support.

So, working on the command line is generally harder for humans to become proficient in. A well-designed GUI can provide a more intuitive experience by making tasks easy, especially for someone new to the technology or someone who only uses it infrequently. It is absolutely better for interactive desktop tasks like launching services, uploading, or configuring something.**[5]**

Therefore, we are developing an assistant that will take the input in a human language either in text or voice, that helps in completing a number of tasks. We're designing it to give people conversational interactions, making it easy for beginners and as well as for developers.

2. PROBLEM STATEMENT

For beginners, it is hard to memorize the CLI commands as there are a large number of commands which need to be learned for various applications. Also, if there is a spelling mistake in the command then it will not respond or fail.

If a user can mis-type an instruction, it is often necessary to start from scratch again.

Also, if we need to work simultaneously with different services like AWS, Docker, etc. We need to run different applications which are time consuming.

3. LITERATURE REVIEW

Every developer hates writing commands again and again, whether they are for launching any service or pushing code to GitHub (upload anything) as they are time consuming. The situation becomes worse when the developer is not able to learn all the commands, and most importantly remember the correct commands, as obviously they cannot remember everything. The immediate next step of the developer is memorizing or searching for correct commands or either getting annoyed by typing long file names repeatedly. So, for a better developer experience, developers’ need to be assisted in such a way that they can get things done like they are speaking to an average person, and it understands them perfectly.

Till date most things in the world have been done using voice communication between people. Now it is time to do it again with one human and Google Assistant. **Google Assistant is the next big thing on the market.** In the next year, using voice will become a rapidly growing trend, creating new markets, devices and sectors we can’t even imagine now.**[3]**

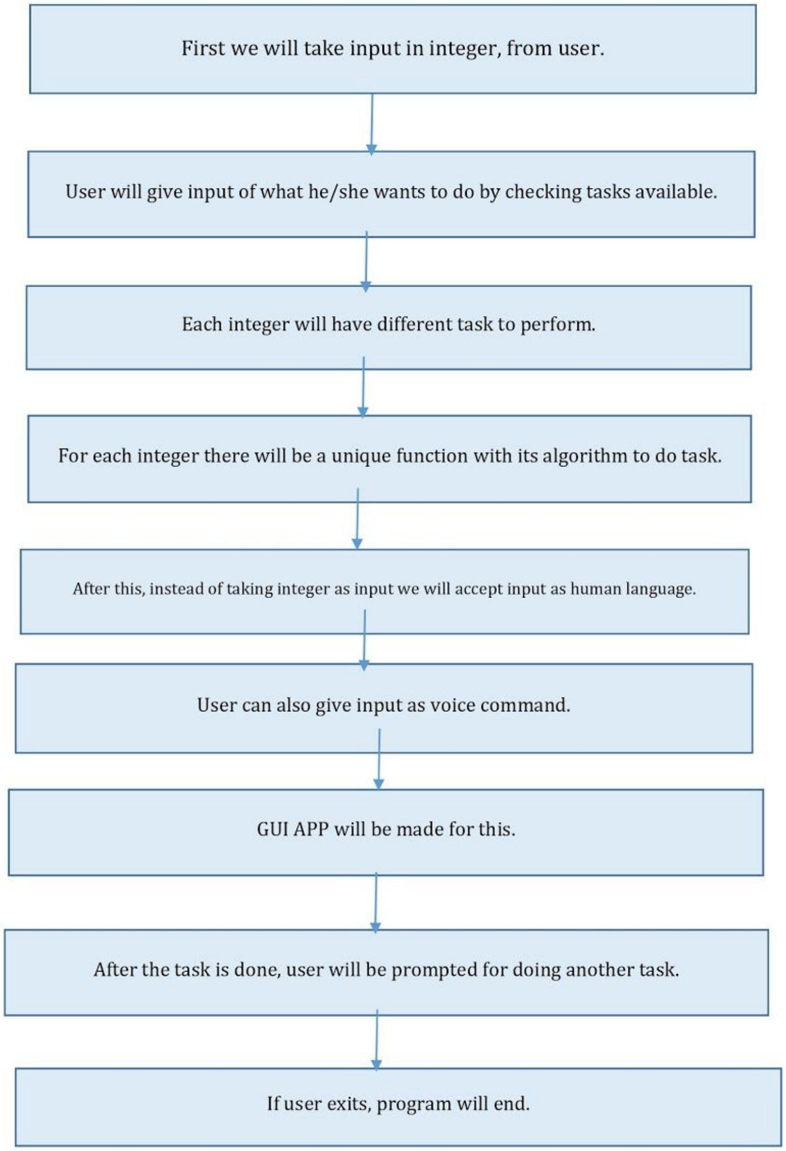
So, catching this wave and surfing the future we decided to create a virtual assistant named Developer Assistant that supports both text or voice entry that will follow the conversation in whichever entry method developer is using.

The voice entry method enhances the developers experience by allowing them to have hands-free interactions and multi-task easily.

4. OBJECTIVES

Our aim is to develop an assistant that works upon voice or text based commands which helps the developer to upload/configure or launch the services like AWS, docker, github, sql and other applications in one go.

5. METHODOLOGY



First we will take input in integer, from user.

User will give input of what he/she wants to do by checking tasks available.

Each integer will have different task to perform.

For each integer there will be a unique function with its algorithm to do task.

After this, instead of taking integer as input we will accept input as human language.

User can also give input as voice command.

GUI APP will be made for this.

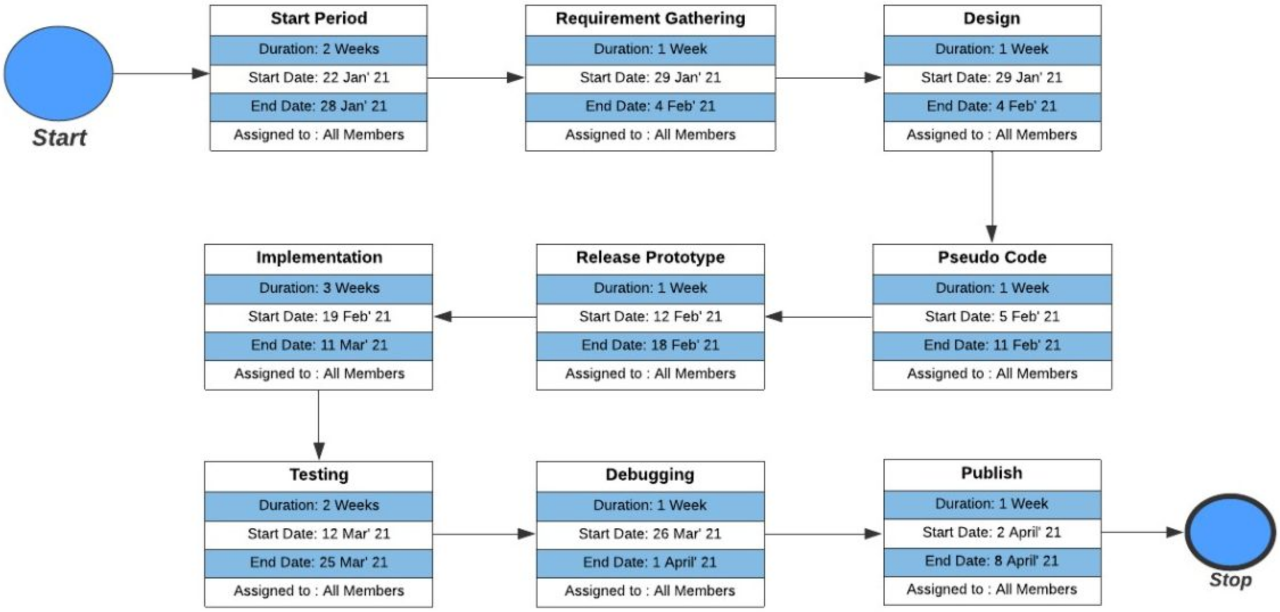
After the task is done, user will be prompted for doing another task.

If user exits, program will end.

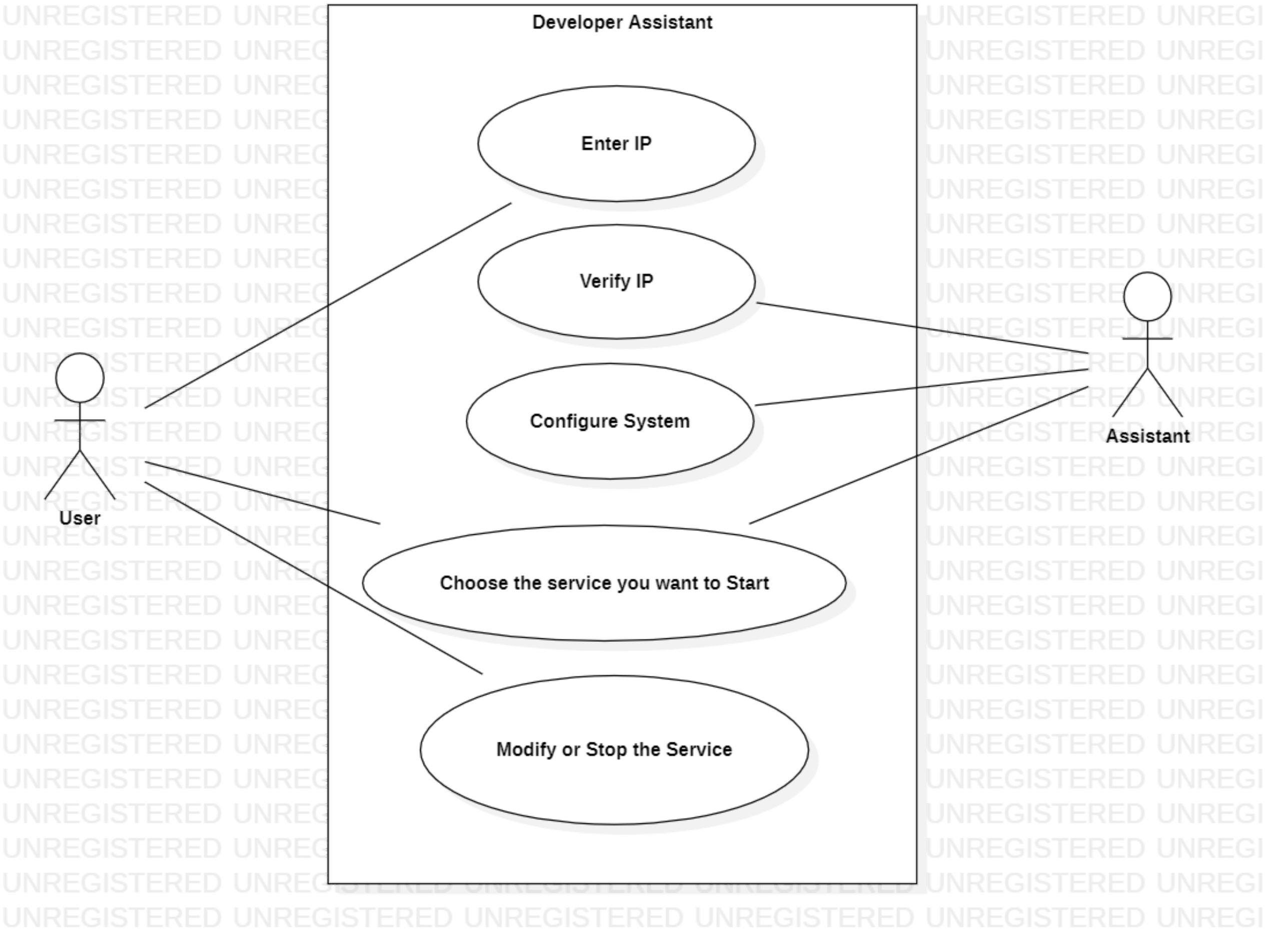
6. SYSTEM REQUIREMENTS

Hardware	**Software**
• Computer system • Minimum 4GB RAM • Microphone	• Linux OS • Python3 • Internet Access

7. 7. PERT CHART



8. **USE CASE DIAGRAM**



9. REFERENCES

[1]https://simpleprogrammer.com/9-common-problems-new-programmers-face/

[2] https://www.toptal.com/software/best-command-line-tools

[3] https://bit.ly/36fW0EH

[4] https://blog.iron.io/pros-and-cons-of-a-command-line-interface/

[5]https://www.quora.com/What-are-the-advantages-and-disadvantages-of-Command-Line-Interface

Approved By

(Name & Sign)	**(Name & Sign)**
Project Guide	**Head of Department**