SOURCE CODE GENERATOR USING SPEECH

Project Reference No.: 42S_BE_0947

College : The Oxford College of Engineering, Bengaluru
Branch : Department of Information Science and Engineering
Guide : Dr. R. Kanagavalli
Students : Ms. Fanoos Fathima
          Ms. Akshata V Kulkarni
          Ms. Deekshitha R

Keywords:  Speech to text, code generation, mapping the text, Speech Interface.

Introduction:
Researchers have shown that most effort of today’s software development is maintenance and evolution. Developers often use integrated development environments, debuggers, and tools for code search, testing, and program understanding to reduce the tedious tasks. Research has shown that more than 60% of software engineering resources are spent on maintenance. Software maintenance requires code comprehension, as reading and understanding source code is the prerequisite of any modification. Program comprehension is time-consuming and cost most of developers’ time. We have proposed to generate code automatically based on speech. The speech input is processed and based on that the code will be automatically generated and add in the appropriate part of the program.

Objectives:
1. The main objective of our project is to provide a platform where the developers can develop the code in a cost-efficient way to complete the project.
2. The developers with less experience can easily code and our platform should be able to provide syntactical error free code.
3. To generate the code with speech input and use NLP to make the platform convenient to use.
4. To provide a platform where the developer can add members of a class with the speech input.
5. To provide an efficient and easy to access predefined code, so that the user can integrate the codes in their classes and to integrate multiple classes in the same platform.
6. To provide a platform where developers can generate, compile and execute the codes all together and display the error message.

Methodology:
1. Speech Processing
   The user of the application uses Android based App to give the speech input. The speech input is processed using Google API and converts it into text. The generated text is sent to the server and further passed to the NLP server. The communication between the Android app and the server is done using HTTP protocol.

2. NLP Server
   Once NLP server receives the input from the server, process the input to find out the part of speech. NLP Server is hosted on the local host and listens at port number 9000. From the java controller running on tomcat call the REST API of NLP using on local host. NLP Server processes the query and returns the result to the controller running in
tomcat server. Based on the returned result from the NLP, the controller checks the input command and performs the required task.

3. Mapping
Once the mapping process is invoked it tries to map with the predefined instructions. If it finds exactly the matched instruction, it adds the code segment into the project or to a respective class. As we are handling object oriented programming, user has simple and easy to use instructions and the details are found in the app itself. The mapper is capable of handling few features of object oriented programming like creating a class, handling inheritance and polymorphism and many more.

4. Code Management
The code management module is handled by the server and checks whether the required classes are there in the package or not and if any other packages need to be imported will be managed by the server. Once the code generation process is over, it invokes the compilation module and compiles the code. Based on that if any exception or bugs are there, it generates the report and send it to the user. User can view the code in the smartphone or in the web interface and debug the code and send it for recompilation. Once the code is free of bugs, compiles successfully, executes the code and returns the result to the user.

Results And Conclusions:

<table>
<thead>
<tr>
<th>Code Description</th>
<th>Manual Typing</th>
<th>Auto Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Sort</td>
<td>4.36 Min</td>
<td>23 sec</td>
</tr>
<tr>
<td>Create Class</td>
<td>25 sec</td>
<td>11 sec</td>
</tr>
<tr>
<td>Create Method</td>
<td>12 sec</td>
<td>7 sec</td>
</tr>
<tr>
<td>Create class variable</td>
<td>5 sec</td>
<td>4 sec</td>
</tr>
<tr>
<td>Create local variable</td>
<td>9 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>Import Package</td>
<td>8 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>Add JButton</td>
<td>32 sec</td>
<td>12 sec</td>
</tr>
<tr>
<td>Add JTextField</td>
<td>31 sec</td>
<td>11 sec</td>
</tr>
</tbody>
</table>

The proposed system provides 42.5% more efficiency than when the code is typed manually. From our proposed work we can conclude that the proposed platform can provide easy to code by the developers. A platform where developers can complete codes, development task with less span of time and with more accuracy and efficiency. We have tested that the auto generated code are bugs free. There are less chances of Syntax errors. This platform can provide standard way of development codes so that when multiple developers are working on
the same project, there will not be any confusion or difficulties to understand the code written by the other developers.

**Scope and Future Work:**

This platform can provide a standard way of development codes so that when multiple developers are working on the same project, there will not be any confusion or difficulties to understand the code written by the other developers. As we have worked on code generation and easy way to add logic into the program, in future we can work on graphics design, graph generation etc. More features in GUI creation. The way we developed the platform to generate code can be applied on different fields other than generating codes, like robotics, artificial intelligence etc. Presently to develop the code we have used Android as the input and Windows platform to generate compile and execute the program. In future we can integrate everything into mobile app or on different platform. It can enhance technical advancements. As of now, we are working on Java. In future, we can extend this to other programming languages also. Java programming language deals with ambiguity with respect to variables and interfaces. This can be mitigated in the future work.