**Group Members**

1. **BSSE 0502 – Jobayer Ahmed**
2. **BSSE 0517 – Shadiqur Rahaman**
3. **BSSE 0518 – Sauvik Bhowmik Anik**

**Supervisor**

**Dr. Kazi Muheymin-Us-Sakib**

**Associate Professor**

**Institute of Information Technology**

**University of Dhaka**

**ACKNOWLEDGEMENT**

We express our humble gratitude o all the persons who helped us in completion our project in time. We also remember all the persons who helped us with important information and timely co-operation.

We would like to thank from the core of our heart and to put special greatefulness to our honourable supervisor **Dr. Kazi Muheymin-Us-Sakib** for his timely response and direction throughout our project. His valuable and wise suggestions have inspired us greately in time of melancholy.

We would also like to thank **Mainuddin Talukder** from BSSE 4th batch for providing us with necessary documents and tutorials.

**Signature of the supervisor**

**First signature() :**

**Second signature() :**

**Third signature() :**

**Final signature() :**

**Contents**

**SYNTEX CHECKER Documentation Preface**…………………

1.1 Purpose of the documentation………………………………

1.2 Intended Audience………………………………………….

1.3 Overview of Document…………………………………….

**SYNTEX CHECKER Project Proposal**………………………..

2.1 Stateent of problem: the “Why?”………………………......

2.2 Objectives: the “What?”……………………………………

2.3 Technical Approach: the “Why”…………………………..

2.4 Project Management: the “How”………………………….

2.5 Team Qualification: the “Who?”………………………….

**Design and Implementation**……………………………………

* 1. Software requirements…………………………………..
  2. System Architecture…………………………………….
  3. Implementation of Syntex Checker……………………..

**User Manual**…………………………………………………..

4.1 Installing Syntex Checker in Windows OS……………….

4.2Installing Syntex Checker in Linux OS…………………….

**Discussion and Conclusion**…………………………………….

5.1 General Description of the project…………………………

5.2 Achievements through the project…………………………

5.3 Future Scope………………………………………………

5.4 Final Remarks……………………………………………..

**Chapter One**

**Syntex Checker Documentation Preface**

This chapter provides the purpose and overview of the document, gives some information about the project and documentation.

* 1. **Purpose of the documentation**

The purpose of the documentation is to present a detailed description of the Syntex Checker. It will explain what the system will do, the constraints under which it must operate with minimum required enviornment. This document is intended for the C programmers specially who are the apprentice in this field.

* 1. **Intended Audience**

This document is made for the C programmers who are apprentice in this field. We have tried a lot to make it easier for the audiences who have a little bit knowledge about C programming. As such it aims for a high level readability for a non-technical audience, while providing enough specifity to be useful to a C programming beginner.

It is assumed that the reader has a general understanding of the use of Windows and Linux operating system and does not require definition of the common programming components like Terminal, Command for Linux etc.

* 1. **Overview Of Document**

The document gives an overview of the functionality of the product. It describes the informal requirements is used to establish a context for the technical requirements specification in the next chapter.

The sections of the document describe the same software product in its entirety, but are intened for different audiences.

**Chapter two**

**Syntex Checker Project Proposal**

**2.1 Statement of Problem: the “Why?”**

The Syntex Checker is an important project for a number of reasons. C is one of the most widely used programming languages of all time. C has facilities for [structured programming](http://en.wikipedia.org/wiki/Structured_programming) and allows [lexical variable scope](http://en.wikipedia.org/wiki/Lexical_scope) and [recursion](http://en.wikipedia.org/wiki/Recursion_%28computer_science%29), while a [static type system](http://en.wikipedia.org/wiki/Static_type_system) prevents many unintended operations. Many later languages have borrowed directly or indirectly from C, including [D](http://en.wikipedia.org/wiki/D_%28programming_language%29), [Go](http://en.wikipedia.org/wiki/Go_%28programming_language%29), [Rust](http://en.wikipedia.org/wiki/Rust_%28programming_language%29), [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29), [JavaScript](http://en.wikipedia.org/wiki/JavaScript), [Limbo](http://en.wikipedia.org/wiki/Limbo_%28programming_language%29), [LPC](http://en.wikipedia.org/wiki/LPC_%28programming_language%29), [C#](http://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29), [Objective-C](http://en.wikipedia.org/wiki/Objective-C), [Perl](http://en.wikipedia.org/wiki/Perl), [PHP](http://en.wikipedia.org/wiki/PHP), [Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29), [Verilog](http://en.wikipedia.org/wiki/Verilog) (hardware description language),[[4]](http://en.wikipedia.org/wiki/C_%28programming_language%29#cite_note-vinsp-4) and Unix's [C shell](http://en.wikipedia.org/wiki/C_shell). These languages have drawn many of their [control structures](http://en.wikipedia.org/wiki/Control_structures) and other basic features from C.

A student of Computer Science or Software Engineering starts his life learning C language first. Its very important to develop programming logic. So, it has a great importance in th life of a computer programmer.

Though C compilers are available for the majority of available [computer architectures](http://en.wikipedia.org/wiki/Computer_architectures) and [operating systems](http://en.wikipedia.org/wiki/Operating_systems), we want to make an efficient platform only for the beginners that will only check the errors of a C code. To be speacialist in this field, one must have strong basics in C. One has to know the syntex of C properly. So, we decided to work with it and our main target is the beginners.

**2.2 Objectives: the “What?”**

Today is the day of computer science. We can’t think our everyday life without computer. What makes the computer so poppuler is software and hardware. Hardware is the physical existence of the computer and software is considered as the life for computer. Now the number of computer users is increasing day by day. The main reason that is working ehind it is man performs their tasks with the help of computer.

An ordinary user just uses a software for some specific tasks. But he may not care what’s working behind it technically. We can’t give command to a computer through our general languages that we use for our communication with each other. There are special languages to communicate with the computer. These languages are implemented through programming. Programming is the craft of transforming requirements into something that computer can execute.

As we are studying software engineering, so we have to know our domain clearly. We develop software for the customer,so we have to think easiness of the customer. A student of software engineering starts his life by studying C language first. He is introduced with computer programming with C language. So it is very important for him to learn this language properly. For this reasons,we have decided to work in this context.

**2.3 Technical Approach: the “How?”**

**2.4 Project Management: the “When?”**

Here are given the estimated workhours and time to complete the project:

* Workhours: 12 hours(weekly)
* Time: 12-14 weeks

**2.5 Team Qualification: the “Who?”**

We may not have vast knowledge on computer programming but we all of us are very much energetic and eager to learn new technologies. To add to this, we can work hard and have strong basics of C programming. All of us are very much helpful and supportive. We also studied a lot in the internet to complete the project properly. We divided the whole work properly between three of us. However, when anyone faced any problem we discussed it together and solved the problem.We also have courses which will be helpful to accomplish this project. They are:

* CSE 101: Structured Programming
* CSE 201: Data Structures and Algorithm

So, we strongly believe that we have a standard team for completing this project successfully and within the time constraints.

**Chapter Three**

**Design and Implementation**

Generally, Design is the creation of a plan or model for the construction of a system. From the view of our project, design represents system requirements that focus on the hardware and software needs, system architecture which describes about software components that are necessary to implement the system. In computer science, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment. Many implementations may exist for a given specification or standard. For example, web browsers contain implementations of World Wide Web Consortium recommended specifications, and software development tools contain implementations of programming languages.

**3.1 Software Requirements**

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These prerequisites are known as (computer) system requirements and are often used as a guideline as opposed to an absolute rule.

To run a software, we need a set of requirement’s because of making suitable environment for the software. There are many requirements needed to all computer operating systems that are designed for a particular computer architecture. Most software applications are limited to particular operating systems running on particular architectures. Software may not be compatible with different versions of same line of operating systems, although some measure of backward compatibility is often maintained.

We have two types of requirements that are :

1. **Hardware Requirements**

To fulfill our requirements we need one PC with atleast 512MB of RAM and 2GHz processor.

1. **Software Requirements**

We have used GCC to compile and run our software, Codeblocks-13.12 to edit the code of the program. C and C++ technology have been used to create the Syntex Checker**.**

**3.2 System Architecture**

**3.3 Implementation of Syntex Checker**

A Syntex Checker is a program that is able to find out the basic syntex errors made by a user. As the project is intended to develop a Syntex Checker for C language, it can be implemented both on Windows and Linux operating system.

The basic concept of Syntex Checker is that it checks the basic syntex errors of a C code. When a C file is loaded by the user it reads every line of that code and checks errors. As soon as an error is found, it prints a relevant error message.

The implementation of Syntex Checker can be divided into several subtasks as given below:

**Chapter Four**

**User Manual**

**Chapter Five**

**Discussion and Conclusion**

**5.1 General Description of the Project**

Syntex Checker is developed for finding syntex errors of a C code. It is specially uesful for the C language beginners. It checks only basic syntex errors. This is more user friendly than other softwares that are avilable to find errors. It is easier to use. An apprentice faces no difficulty to use it.

**5.2 Achivements through the project**

Though we had strong basics in C language, we didn’t have enough idea on C++. Now we have learnt more about this language because the project is totally based on C and C++. It gave us the joy of learning new thing. We think that it will be very helpful for us in future.

Team work is very important in Software engineering which we have tried our best in our project. From this project, we have learnt how to communicate with the team members and how to co-ordinate the whole team work. It also gave us the full flavor of group working. We also learnt the art of module programming through this project.

We have practiced punctuality throughout our project. We have to submit our Project Proposal, Requirement Analysis, Project Mid-Work, ProjectFinal Work and Project Documentation in time. So, we have developed the good practice of punctuality in our character.

Communication skill is always important in any type of field such as Software Engineering, Marketing etc. We have developed our communication skill throughout the project. We have to give three presentations and meet our Supervisor several times. Thus, the project has helped us to build communication skill to a greater extent.

**5.3 Future Works:**

Now this software is able to check the basic syntex errors. It can’t find any error in some fields of a C code like array,pointer etc. So,we want to work in these fields in future.We also want to add some interesting features like autocorrection etc. We also want to develop the software in such a way that it can check all type of errors. Finally we want to build an editor of its own where a user can write a C code, compile it and run it there. And we are very hopeful that we will be able to reach our goal.

**5.4 Final Remark:**

The software we developed is intended to the C programmer. The success of this project may come to help the C programmers, especially those who are the beginners in this field.