Software Requirements Specification

Version 2.0

**Prepared on**

**Date: 11-2-2013**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | * Aravindan M * Rajvardhan Singh Dhakrey * Shah Ronak Udaykumar * Sripradha K * Vipul B Marlecha * Tanyeem Khonane * Ashish Shetty * Shreya Chippagiri * Deepa Y * Shivaraj | NA | 08/02/2013 |
| 2.0 | * Aravindan M * Rajvardhan Singh Dhakrey * Shah Ronak Udaykumar * Sripradha K * Vipul B Marlecha * Tanyeem Khonane * Ashish Shetty * Shreya Chippagiri * Deepa Y * Shivaraj | Amendments made according to the validation checklist. | 11/02/2013 |

# 

# Introduction

# The advancement of Internet towards Web 2.0 conveys the potential it has in a wide range of scopes. The ongoing progress of the Web technology and its availability in teaching and learning, as user’s proﬁle has given potential to the development of myriad web applications.

# e-Compiler - is an application which greatly lightens the user’s workload at the initial stage of programming. During this initial period the users will neither have to deal with the complexities of the installation and the conﬁguration of compiler tools, nor with the understanding of multiple options which the software presents.

# With an online compiler, the user can store the code online, compile and execute it, allowing him the freedom to program from any computer (or device) with a connection to the internet.

**1.1 Document Purpose**

The purpose of this document is to present an overview of the e-compiler, version 1.0. It will explain the purpose and features of the web application, the interfaces, what the application will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

## 1.2 Product Scope

Often while compiling a code, we observe that the required compiler for the language doesn’t exist on the system. Downloading and installing the compiler can be quite a cumbersome task. Moreover, certain compilers and their versions may not be compatible with system configurations. Portability across operating systems can also be an issue.

The idea of an online compiler is to create a generic portal where codes of any language (considering few in our project) can be compiled online conveniently. The compiler will be set up on the server and given a code by the client, it will be compiled and executed on the server and the output thus generated is displayed on the client browser. The error messages after compilation and the runtime error, if any, will be displayed to the client too. Also, the user has the facility for maintaining an account on the website, giving the user the privileges of privacy, saving the codes and sharing them with the intended other users.

## 1.3 Intended Audience and Document Overview

This document is intended to be read by

• The Client

• The Developer.

• The Professor.

The reader is assumed to have basic knowledge of the basic theory and operation of Compilers, and be aware of technical terminologies used in the document. Also, knowledge in reading use case diagrams and activity diagrams is required.

## 1.4 Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Client | A computer or computer program that initiates contact with a server in order to make use of a resource. |
| Compiler | A computer program (or set of programs) that transforms source code written in a programming language (the *source language*) into another computer language (the *target language*, often having a binary form known as *object code*). |
| GUI | **Graphical User Interface**  A type of user interface that allows users to interact with electronic devices using images rather than text commands. |
| HTTP | **Hypertext Transfer Protocol 1.1** |
| IDE | **Integrated Development Environment**.  A software application that provides comprehensive facilities to computer programmers for software development |
| Operating System | A collection of software that manages computer hardware resources and provides  common services for computer programs. |
| RDBMS | **Relational Database Management System** |
| Server | A computer system that selectively shares its resources |
| SMTP | **Simple Mail Transfer Protocol** |
| Web Application | A computer software application that is accessed by users over a network such as the Internet or an intranet. |
| Web Browser | A software application for retrieving, presenting and traversing information resources on the World Wide Web. |

## 1.5 Document Conventions

Formatting conventions:

* The SRS has been written with the text in font Arial and size 11.
* The main and the subsection headings have been given in bold with text size 14 and 12 respectively.

## 

## 1.6 References and Acknowledgments

**World Wide Web**

“SRSExample-Webapp.doc”.Internet:[www.cse.msu.edu/~chengb/RE.../SRSExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE.../SRSExample-webapp.doc) [Feb 5, 2013].

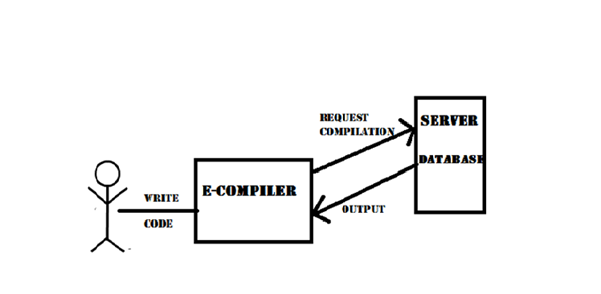
**Books**

*Ali Bahrami, “Object-Oriented Analysis Process: Identifying Use Cases” in an overview of object-oriented systems development, Irwin/TataMcGrawHill, 1999, ch.6, sec.6.6,pp 129-133*

**2.** **Overall Description**

## 2.1 Product Perspective

The product, e-Compiler, is a web application where users can compile and execute their code online using a web browser. The features of the application include a user account for the user, wherein he/she can login and save his/her codes. The product also facilitates the user to share the code with other users. The user can choose from a given set of programming languages and compile and execute the code. The user’s data and preferences are stored in a database on the sever. The product is a complete package and is not a subsystem of any other larger system.



**Figure 2.1 Product Perspective**

## 2.2 Product Functionality

* User Registration and Login
* Editor for the user to write code
* Code Compilation and Execution
* Code Repository
* Code Sharing

## 2.3 Users and Characteristics

**New Users (Unregistered users)**

Unregistered users can just compile and execute their program online.

**Registered Users**

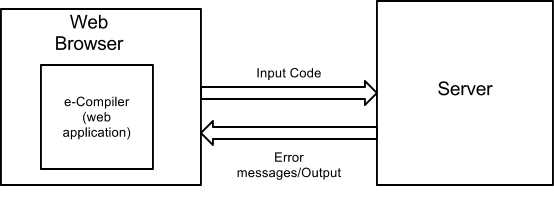
In addition to code compilation, registered users can also save and share their code.

**Beginners**

The product would be more useful to beginners who have just started programming. They could be registered or normal users. Beginners are usually tentative installing the required software to start coding. The product will help these users to practice and learn programming online.

## 2.4 Operating Environment

The product being a web application runs on the web browser. The user needs to have a web browser and an internet connection to use the product. The web browser should support images.



**Figure 2.2 Operating Environment**

## 2.5 Design and Implementation Constraints

* The numbers of users that can access the product simultaneously is dependent on the server capacity.
* Usage of system calls is prohibited to ensure security of the server.
* A user with a slow internet connection chokes the server thereby reducing the number of concurrent users.
* The languages the user can input his code for is limited to C, C++ .
* Limited Database Storage capacity may put a cap on the storage space of each user as the number of users increases.

## 2.6 User Documentation

The product being a web application runs on the web browser. The user should know how to use a web browser. The user should know how to access a web page and use the GUI associated with it. The online code editor will be similar to a text editor. The user should be familiar with the usage of a text editor.

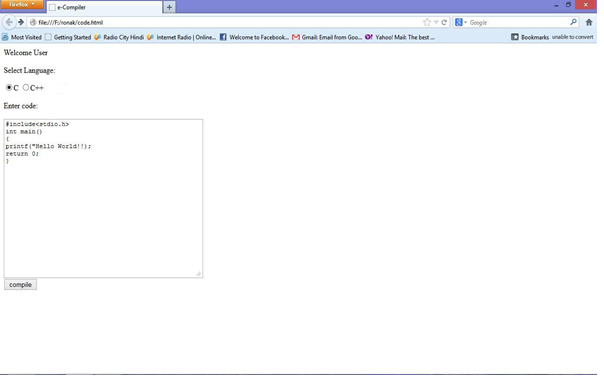
## 2.7 Assumptions and Dependencies

* It is assumed that the user will not make use of any system calls in the program which may alter the functioning of the server.
* It is assumed that the user will only make use of the libraries and API’s that are present on the server for that language.

1. **Specific Requirements**

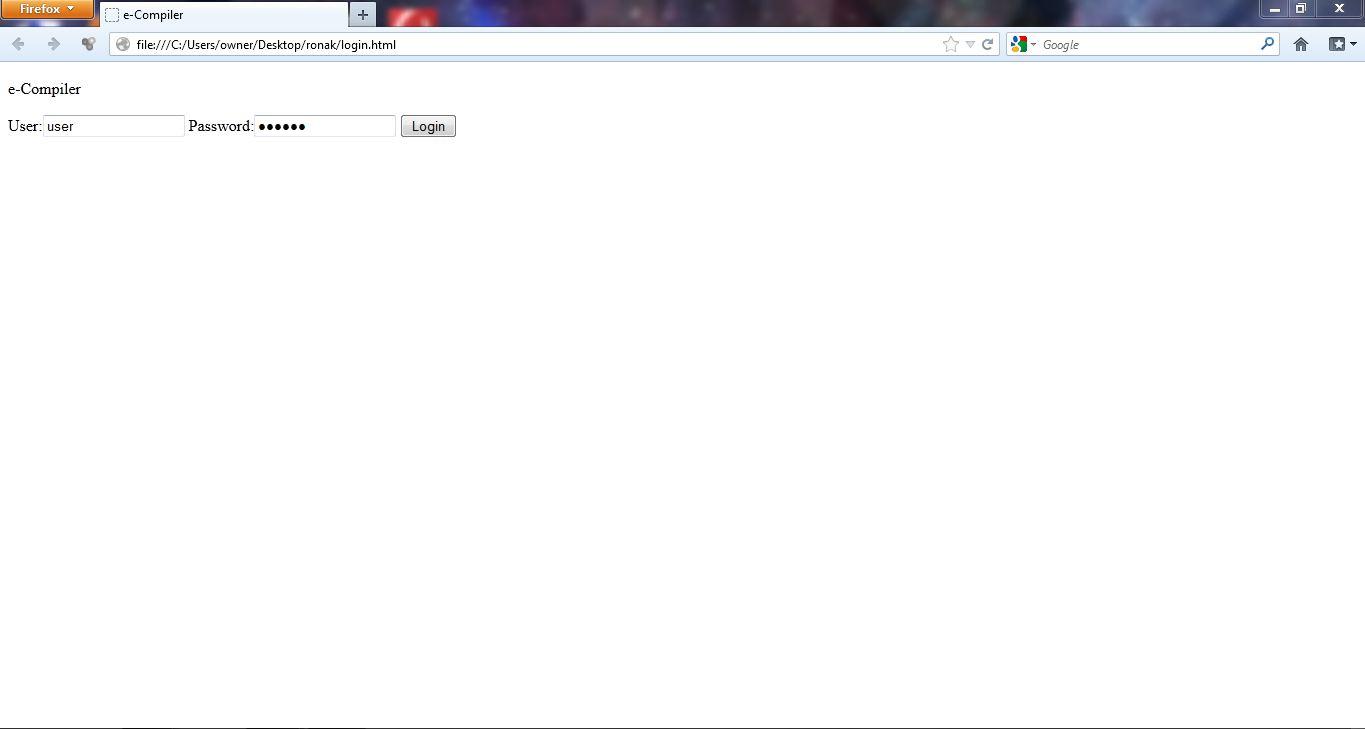
## 3.1 External Interface Requirements

### 3.1.1 User Interfaces



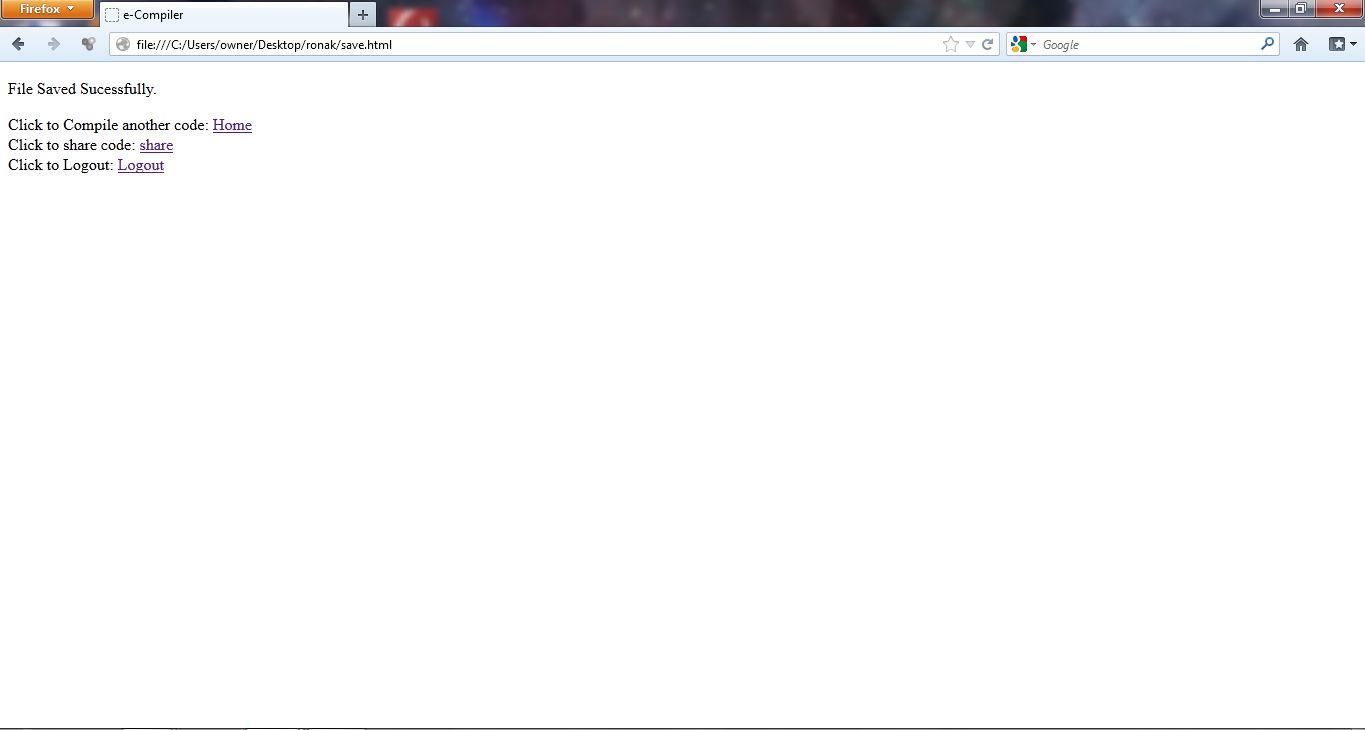
**Figure 3.1 Home Screen**

The above attached image is the welcome screen of the application. This is the space where the user can start using the application interactively with the environment for writing the program, compiling and executing it.



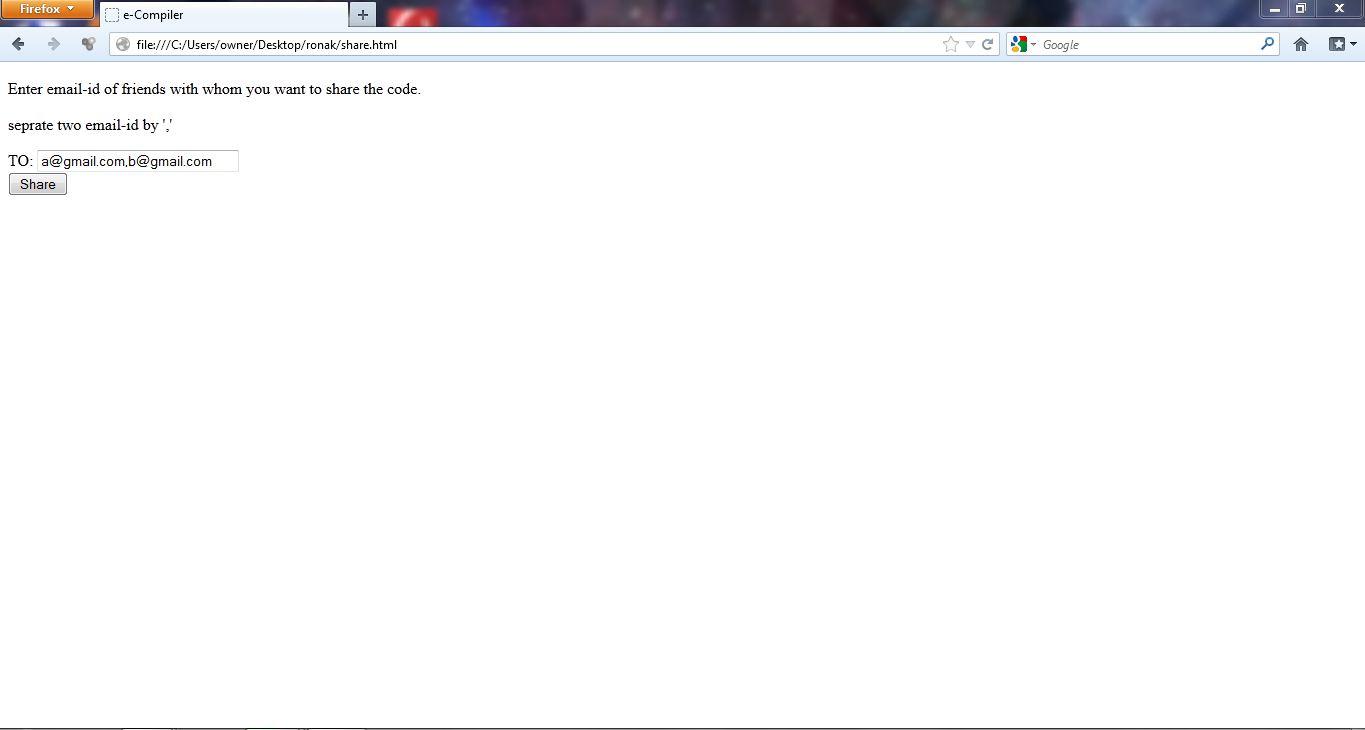
**Figure 3.2 Login Screen**

The above is a login page through which the user can enter a more privileged interface of sharing and saving information.



**Figure 3.3 Success Message**

This screen shows the confirmation of a saved file and provides provision to get back to the home page.



**Figure 3.4 Sharing Screen**

The share screen provides the facility to share the successfully compiled code with anyone using their email ids and the sent screen is the acknowledgement for the same. A logout link is provided for logging out.

### 3.1.2 Hardware Interfaces

Keyboard will be used as a means of input from the user for interaction and also as a means of entering the code to be submitted for compiling, to the application.

The libraries needed will be enlisted during the implementation phase of the development.

### 3.1.3 Software Interfaces

The system calls (Unix, Linux) made by the various compilers installed on the server will be the only interaction of the server with operating system. The system shall also make use of operating system calls to the file management system to store and retrieve files containing the saved and compiled code.

MySQL or Microsoft Excel RDBMS will be used to save the user information (username and password).

Application will be run in a web browser.

### 3.1.4 Communications Interfaces

The application shall be using HTTP protocol for the interaction between the server and the browser.

Since sharing of code includes e-mails, SMTP protocol also comes into the picture.

The application also includes encryption for the login facility

## 3.2 Functional Requirements

**Login**

**Register**

**Compile**

**Save**

**Share**

**Logout**

**Figure 3.7 Functional Requirements**

**The following are the functional requirements of the application:**

* **User registration and Login**

The facility provides users with a privilege to log into the application for sharing and saving the code to the repository.

* **Editor for the user to write code**

Editor is a space where the users can write their code and compile it immediately to produce the output, compile and debug it.

* **Code compilation and execution**

The code submitted by the editor can be compiled and executed online providing output.

* **Code repository**

Any registered users can save their compiled codes online on the repository that can be retrieved for editing or future compilation.

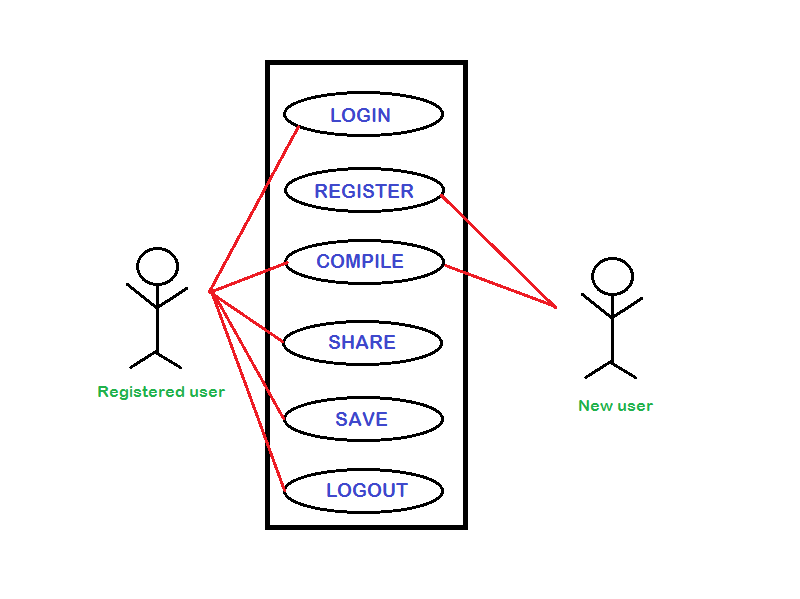
* **Code sharing**

The application provides a unique feature of sharing the compiled code through e- mails.

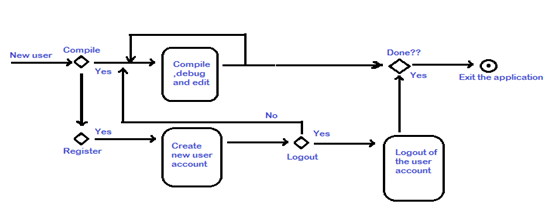
## 

## 3.3 Behavior Requirements

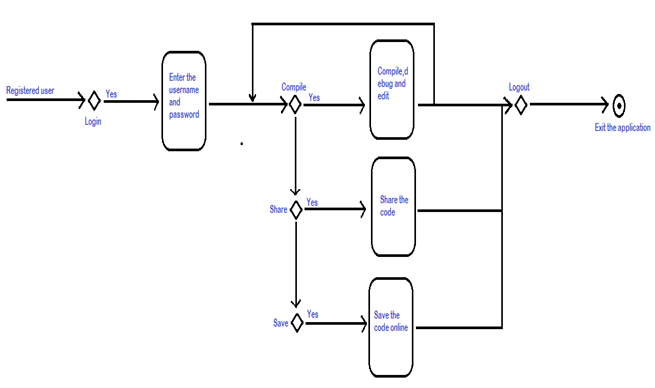
### 3.3.1 Use Case View



**Figure 3.8 Use Case Diagram**



**Figure 3.9 Activity Diagram for New User**



**Figure 3.10 Activity Diagram for Registered Users**

**4. Other Non-functional Requirements**

## 4.1 Performance Requirements

Since this is a web application, the performance would depend entirely on the user traffic. In cases where the number of concurrent users is very large, the server may be down. Otherwise, in all ideal cases of optimum user traffic, any transaction such as logging in, saving and sharing should not exceed 20 seconds. Compiling and execution of the code on the server depends upon the speed of the server.

**4.2 Safety and Security Requirements**

* The application has been provided with a register, login feature wherein the user is required to provide a password which would be encrypted using encryption standards. This is a layer of security being provided by the application.
* The user has the privilege and privacy of sharing the code with the intended users.
* In cases where the user has forgotten the password, an alternate way to access the account will be provided.
* The user owns the copyright reserves of the code input. The product will not make use, modify or plagiarize the code provided in any which way.

## 4.3 Software Quality Attributes

**4.3.1 Availability**

Availability is concerned with server down/failure and its associated consequences.

In all ideal cases, the server is always powered on and working and the application is available.

Also, the availability depends on the presence of a net connection. If there's no net connection, then the user cannot access the application as it works on a browser.

**4.3.2 Performance**

As our application is Web-based system, it gets input from the users which must be compiled in the language selected and produces an appropriate output in a short time, not exceeding 30 seconds. Satisfying the request requires resources to be consumed. While this is happening the system may be simultaneously servicing other requests. Hence, performance highly depends upon the user traffic.

**4.3.3 Security**

The application requires the user to provide username and password to access his account. The password is encrypted using encryption standards. Hence, security is achieved by the system.

**4.3.4** **Usability and reusability**

The application is highly user friendly. The user is expected to be comfortable working with a web browser and text editor only. Since the GUI of the application will be highly interactive, the user will find it easy to navigate through the web pages to accomplish the desired task. The application is targeted at a software coder, computer science student/professor. The user can use the web application any number of times he wishes to or the need arises.

**4.3.5 Reliability**

The output provided by the application is reliable as the application makes use of the standard compilers for the language specified. The application is also reliable in the sense, the server is always powered on. A back-up server may be maintained if need be.

**4.3.6 Portability and Interoperability**

The application provides a generic portal for the user to compile his code. Since it is a web application and runs on the browser, it is portable on all machines installed with a browser. Further,the code is compiled and executed irrespective of the IDE it is developed in.

**4.3.7 Maintainability**

The application makes use of the latest compilers. As and when new releases of a compiler are made, the application incorporates it and is thus maintainable.

**5. Other Requirements**

NA

**Appendix A – Data Dictionary**

|  |  |
| --- | --- |
| Client | A computer or computer program that initiates contact with a server in order to make use of a resource. |
| Compiler | A computer program (or set of programs) that transforms source code written in a programming language (the *source language*) into another computer language (the *target language*, often having a binary form known as *object code*). |
| GUI | **Graphical User Interface**  A type of user interface that allows users to interact with electronic devices using images rather than text commands. |
| HTTP | **Hypertext Transfer Protocol 1.1** |
| IDE | **Integrated Development Environment**.  A software application that provides comprehensive facilities to computer programmers for software development |
| Operating System | A collection of software that manages computer hardware resources and provides  common services for computer programs. |
| RDBMS | **Relational Database Management System** |
| Server | A computer system that selectively shares its resources |
| SMTP | **Simple Mail Transfer Protocol** |
| Web Application | A computer software application that is accessed by users over a network such as the Internet or an intranet. |
| Web Browser | A software application for retrieving, presenting and traversing information resources on the World Wide Web. |

**Appendix B**

Further upgrades to the SRS may be done as when there may be increments and modifications to the requirements. Additional functionalities may arise during the design and implementation phase.