**Feasibility Report**

**Version 2.0**

**Prepared on**

**Date:23-3-2013**

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

**Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Revision #** | **Date** | **Description** | **Authors** | **Approver** | **Comments** |
| 1 | 19-03-2013 | Feasibility Report-Baseline version. | 1.Ashish Shetty  2.Tanyeem Khonane  3. Deepa Y  4.Sripradha K.Rao | Professor Padmasri M.V. | 1. Ambiguous statements. 2. Aesthetics of statements can be improved. |
| 2 | 23-03-2013 | Feasibility report-Final version.  v2.0 | 1.Sripradha K.Rao  2. Shreya Chippagiri | Professor Padmasri M.V. | Final version to frozen |

1. INTRODUCTION
   1. Overview

# 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. The project is currently in the analysis and planning phase.

* 1. Project Scope
     1. The Problem

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.

* + 1. Scope

The idea of an online compiler is to create a generic portal where codes of any language (considering C and C++ 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. Processes and Data

The code to be compiled is sent to the server and the server responds with the success or error messages.

The user can share the code with other users. The file is retrieved from the server and can be accessed.

The user details which comprises of the data for the application will populate the database. The database entries can be retrieved as well as updated.

* + 1. High level system requirements

The system requirements include:

* Computer.
* Client-based web browser.
* Back-end server.
* MySQL database.

1.2.5 Constraints

* 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.

1.2.6 Team coordinator/Interface for the project

The team coordinator for the project is the team leader, Ms.Shreya Chippagiri.

1.2.7 Sponsor/People Involved with the project (other than the students)

Not Applicable.

* 1. Documentation/Literature Survey Summary

Since we are learning web technology as part of the course, not much of extensive literature survey was required. However, the back-end team had to research on PHP, LAMP. Sites like stackoverflow.com, codeacademy.com, w3schools.com were visited. Online tutorials and blogs were read to learn more about MySQL.

* 1. Reference
     1. 1. 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].*

HTML: *www.w3schools.com/html/default.asp*

CSS: *www.w3schools.com/css/default.asp*

PHP: *www.w3schools.com/php/default.asp*

LAMP: *help.ubuntu.com/community/ApacheMySQLPHP*

* + 1. 2. 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. Project Approach

The project will be implemented as an agile process using iterative incremental development principles.

The front-end, back-end and the database are implemented in parallel. The three will be integrated in the end to obtain the final work product. Additional user requirements will be added as and when the need arises under the given time-constraint.

1. Potential Solutions

The concept of e-Compiler is to provide the users with a platform, so that they can compile their codes without bothering about the compiler being installed on OS or even the version of the compiler.

It further provides additional functionalities such as saving and sharing the compiled code. It provides a user friendly, interactive web based interface for the user to conveniently compile his code.

* 1. Required resources

The standard input device, 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. Web browser is required to be installed on the user’s machine to interact with the application. The libraries needed will be specified during the implementation phase of the project.

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).

3.2 Cost/benefit analysis

## 

Figure 3.1 Functional Requirements Diagram

**Information Domain values**

**1. Number of inputs:**

/\*Each user input that provides distinct application oriented data to the software is counted.\*/

a. Registering - name, password, e-mail id, gender = 1

b. Compiling the submitted code - the code snippet =1

c. Sharing - The e-mail id for sharing = 1 (best case)

Hence the number of inputs: 3

**2. Number of outputs:**

/\*Each user output that provides application oriented information to the user is counted. \*/

a. Authentication when logged in = 1

b. Compiled output- error messages, compilation success output = 1

c. Sharing - The success screen for sharing the code snippet =1

d. Saving - Screen for finding the previously saved files and save the current file =1

e. Registration success screen = 1

Hence the number of outputs: 5

**3. Number of user enquiries:**

/\*An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output.\*/

a. Authentication login - username and password =1

b. Saving – file name =1

Hence the number of user enquiries: 2

**4. Number of files:**

/\*Each logical master file is counted.\*/

a. Backend Database file which contains authentication details and mapping to their respective profiles =1

b. The user saved file = 1 (best case)

Hence the number of files: 2

**5. Number of external interfaces:**

/\*a user identifiable group of logically related data that is used for reference purposes only\*/

1. Compiler setup file : with implementation of 3 languages =1

Hence the number of external interfaces: 1

**Value Adjustment Factors**

1. Reliable backup and recovery=4

2. Specialised Communication=3

3. Distributed processing function=4

4. Critical Performance=2

5. System operational environment=3

6. Online data entry=5

7. Multiple screens=4

8. ILF Updation=4

9. Complexity=3

10. Internal processing=4

11. Reusability=1

12. Installation= NA

13. Multiple Installation=NA

14. Ease of Use=3

|  |  |  |  |
| --- | --- | --- | --- |
| **Information domain value** | **Count** | **Weight** | **FP count** |
| Number of inputs | 3 | 4 | 12 |
| Number of outputs | 5 | 5 | 25 |
| Number of user enquiries | 2 | 4 | 8 |
| Number of files | 2 | 10 | 20 |
| Number of external interfaces | 1 | 7 | 7 |
| **Count total** |  |  | **72** |

**Calculations**

FPestimated = count-total [ 0.65 + 0.01 \* (F i) ]

40

FP estimated = 76

Average productivity of the organization = 6FP /pm

Labour rate for one person = 25000 /pm

Man -months=76/6=13 man months

Total estimated cost= 13\*25000= Rs 3, 25,000

Cost per fp= 3, 25,000/76 = Rs 4277

* 1. Expected duration

The team consists of 10 members, 3 of whom are involved in testing. In order to keep the testing team completely unbiased and separated from the development team, the team in effect comprises of 7 members. These 7 people work in groups on this project without any interference or interdependency. Thus the project is can be completed within:

13 person-months/ 7 people = 2 months .Due to other courses, ongoing projects, assignments and events, the project spans over 3 months. Thus the project is time-feasible.

* 1. Major deliverables

The major product which is being delivered is an online compiling application, which will help the user to 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. The major document deliverables include:

* The Software Requirements Specification(SPEC) document.
* The feasibility report.
* The design and architecture document.
* The source code.
* The test report.
  1. Project Plan Deliverables
     1. Feasibility Report

**Technical Feasibility**

* The software is a client-server based web application.
* Apache web server is required to run this application over the internet.
* It is a browser based application, hence OS independent. The client is required to install the browser which could be Mozilla Firefox, Google Chrome or Internet Explorer 7, etc.
* The server will be made platform compliant so that the user can input any code that will be run on the appropriate server platform. Thus, platform independence is achieved from the user point of view.
* The front-end comprising of the GUI will be implemented using HTML, CSS and JS.
* The back-end is implemented using PHP.
* The database will be developed using MySQL.
* All this can be implemented easily using LAMP in Linux and WAMP in Windows.

All of the above mentioned requirements for development are well within that that can be expected out of the technical expertise of the development team. The above mentioned technologies are open-source, user-friendly, easy to install, develop on and deploy and cost-free. They also have built-in debuggers provided. They are highly convenient as they are extensive in usage, easily available and do not require additional training.

**Time Feasibility**

* According to the cost-estimate 13 man-months are required to complete the project.
* Since there are 10 people working in groups on this project without any interference or interdependency, the project is time feasible.
* The delivery date can be met conveniently.

**Cost Feasibility**

* The budget comes upto Rs.4300 approximately. The project is cost feasible.
  1. Use cases for the solution

1. Amit is an engineering student currently in his 3rd semester Information Science and Engineering at PESIT, Bangalore. Since it is his first semester in coding, he still has some ambiguities with reference to code compilation. He isn’t very comfortable with the features provided by the different versions of a compiler and its platform dependency. He finds himself in a dilemma when it comes to compiling his code. He also wishes if there could be a way he could share his programs among his friends and maintain a history of the same.

In this scenario, the purpose of an online compiler is justified. Amit now conveniently uses the e-Compiler application to compile all his programs, save and share them amongst his friends. He also gets the added advantage of securing his programs by maintaining his personal account on the website.

1. Swathi is a software engineer at IBM. Due to work purposes, she is often travelling. Though she has a work laptop, she finds it difficult to continue with her work when her laptop battery gives way. In such situations, she would have to borrow another person's laptop or desktop as an alternative to continue her work. Chances are that, the required version of the compiler or the platform it is compliant on may not be available on that particular system.

In such cases, she conveniently uses the e-Compiler web application to compile her programs. Since she does not see the need to maintain a permanent account on the website, e-Compiler provides her with the option to compile her program without registering and logging-in.

* 1. High level risks and challenges

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

Performance highly depends upon the user traffic.

* 1. Time commitments from Team

Team meetings were conducted regularly after college hours for 30 minutes. Also, group chats on Google and Facebook were held to discuss about the documentation.

* 1. High level schedule with Milestones

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sl..No | Phase | Activity | Start Date | Planned End Date | Who all | Comments |
| 1. | Identification | Ideas were jotted down and zeroed down to one. | Jan 28 | Feb1 | All team members |  |
| 2. | Scope | The scope of the project was written and finalized. | Feb 1 | Feb 3 |  |  |
| 3. | SRS | The SRS was reviewed and validated. | Feb 6 | Feb 9 |  | Validation by team 4 |
| 4. | Design and Documentation | The front-end of the application was decided in the design phase. | Feb 16 | Mar 1 |  |  |
| 5. | Coding | Back-end coding started. | Mar 11 | Apr 24 |  |  |



* 1. Interaction and Communication plan within and outside the project

The team members working on database have to interact with the front-end team members of the group.

3.11 Assumptions

* + - It is assumed that the user will not make use of 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.

3.12 Other considerations

*Other plans like SCM, SQA and Test Plan etc. would follow as we follow the course*