Ain Shams University

Faculty of Computer & Information Sciences

Computer Science Department

WEB-BASED IDE

WEB-BASED IDE

Team Members

1-Ahmed Abdellatif Mhana CS
2-Ahmed Magdy Mohamed Zaki CS
3-Mohmed Nabil Mansour CS
4- Ahmed Alaaeldin Abdelwahab CS

Supervisors

1- Dr. Ahmed A Gawish
2- Dr. Zaki Taha Zaki

Acknowledgment

All praise and thanks to ALLAH, who provided me the ability to complete this Work. I hope to accept this work from me.

I am grateful of my parents and my family who are always providing help and support throughout the whole years of study. I hope I can give that back to them.

I also offer my sincerest gratitude to my supervisors,

Prof. Dr / Zaki Taha Zaki

Dr. Ahmed Gawish

Finally, I would thank my friends and all people who gave me support and encouragement.

Abstract

Web-Based IDE is made to create a strong framework for the developers to enable them to code edit, share and run the code from their browsers anywhere at any time.

The first problem is that many devices now tends to be small and portable like tablets PCs, mobile phones , To make the device as small as possible this effects the memory and CPU sizes and capabilities greatly , With the current versions of IDEs they can’t have installed IDE , even if there is one it’s not as powerful as the one which runs on a PC, second problem is that there is a gap between the IDEs and software that manages versions and share the code (version management)between developers who are working in the same project , Also The Unstably of this services, that gap demanded to be covered by a software that allows developers to share the code and edit it together with minimum effort to configure the software taking in consideration that it must be easy to use, On another point of view every application is now taking it the step of making the cloud version of itself so it’s a natural step forward for the IDE software.

The goal was to solve these problems by creating this web service which allows the developers to code, communicate, run, share, and collaborate together whiteout a lot of configurations.

Taking steps to create this project by learning more about the problems, Progress of other in this area, Which programming langue to target and which developers will use it the most ,learning the current web technology, Knowing how current compilers and IDEs work, How to connect this to web technology in the most useful way.

So the idea was to make this web service solving the problems and adding a new value for the software developing community.

Table of Consents

List of Abbreviations

IDE : Integrated Developing Environment

EJS : Embedded java script

SQL : Structured Query Language

List of Figures

List of Tables

Chapter One:

Introduction

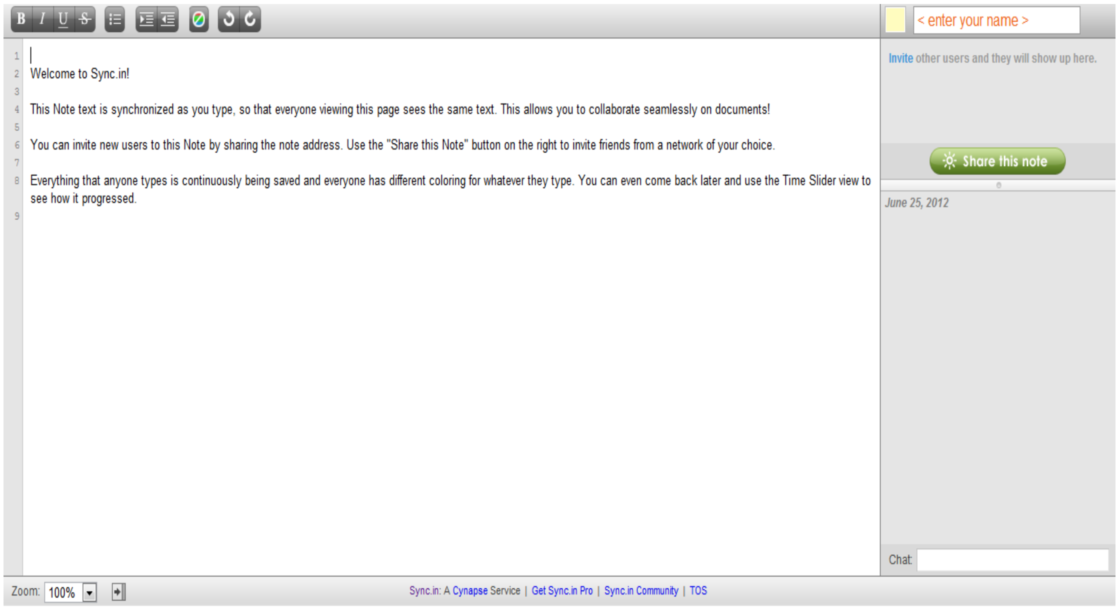
1.1. Overview

The idea of a web-based IDE received a lot of attention in the recent years because of the rapid developing of software and hardware which tends to give the devices smaller sizes limiting the capabilities to run IDEs software this proposes a problem how can developers make software on this devices , how to make use of the new founded features on this devices to make the experience of software developing better, Developers usually needs to share the code together or even work on the same file and that causes a lot of problems especially if the developers are not in the same physical place or can't communicate easily, Some solutions exist but it requires hard and annoying configurations and it's not stable , More important is every software tends to go cloud or online , so it's a natural step forward in the process .

Web-based IDE allows all of this problems to be solved by taking the whole developing experience to the web , coding , sharing , editing , collaborating and even running, which and that solves the problem of the limited hardware capabilities of the portable devices, Since all you need is a device that can run a web browser, also you whole code is hosted online so you can share it with anyone , and taking an extra mile the Real-Time Collaboration feature was added so two persons can directly edit the same file at the same time seeing the changes while it happens.

1.1.1. History

The idea of the a web-based IDE dates back to 2008 when the idea of the real-time collaboration was found to be extremely useful for developers to share code the first version of the ether pad was launched in November 2008



Welcome to Sync.in!

This Note text is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents!

You can invite new users to this Note by sharing the note address. Use the "Share this Note" button on the right to invite friends from a network of your choice.

Everything that anyone types is continuously being saved and everyone has different coloring for whatever they type. You can even come back later and use the Time Slider view to see how it progressed.

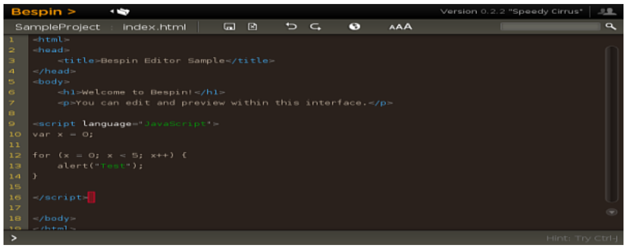
Invite other users and they will show up here.

June 25, 2012

Sync.in: A Cynapse Service | Get Sync.in Pro | Sync.in Community | TOS

Figure 1.1: Ehterpad

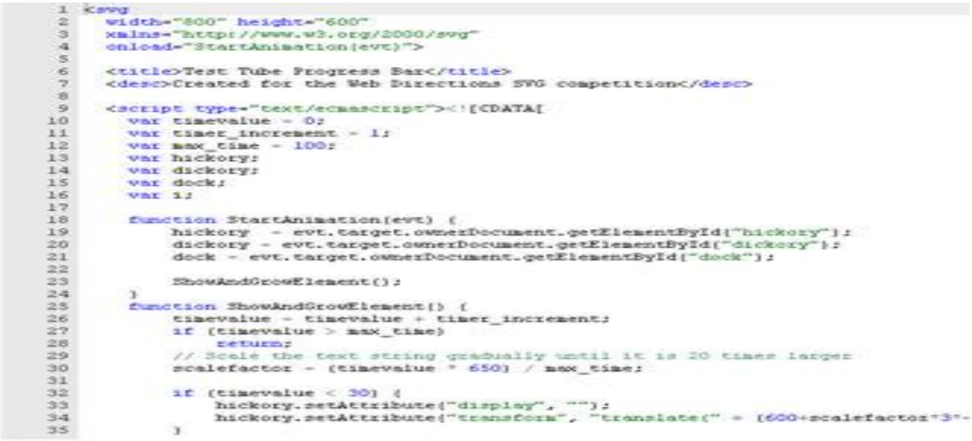
PiratePad, Telecomix Pad, Framapad, Mozilla Pad (MoPad), PrimaryPad, TypeWith.me, Sync.in, TitanPad and iEtherPad.com. Then to take it to the next level the real-time collaboration applications and developers added the syntax highlighting, then Mozilla at 2010 annouced the developing of Bespin which is: open extensible web-based framework for code editing that aims to increase developer productivity, enable compelling user experiences, and promotes the use of open standards.



```
<html>
<head>
    <title>Bespin Editor Sample</title>
</head>
<body>
    <h1>Welcome to Bespin!</h1>
    <p>You can edit and preview within this interface.</p>

<script language="JavaScript">
var x = 0;

for (x = 0; x < 5; x++) {
    alert("Test");
}

</script>

</body>
</html>
```

Figure 1.2: Bespin

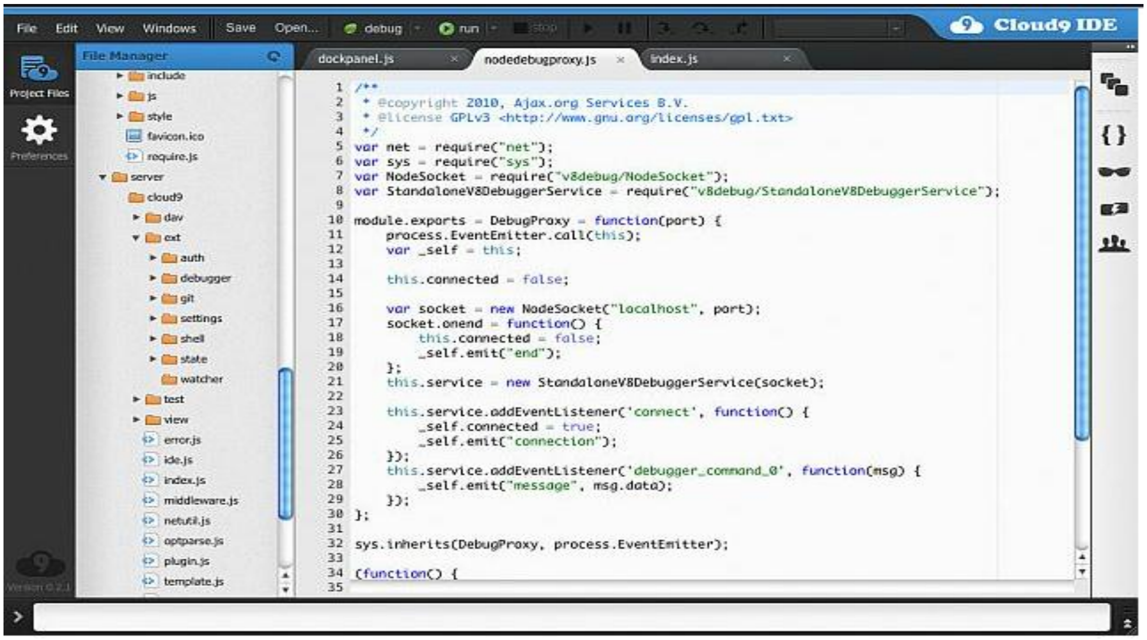
Then It developed a newer version which was called sky writer



```
<svg
  width="800" height="600"
  xmlns="http://www.w3.org/2000/svg"
  onload="StartAnimation(evt)">

  <title>Test Tube Progress Bar</title>
  <desc>Created for the Web Directions SVG competition</desc>

  <script type="text/ecmascript"><![CDATA[
    var timevalue = 0;
    var timer_increment = 1;
    var max_time = 100;
    var hickory;
    var dickory;
    var dock;
    var i;

    function StartAnimation(evt) {
        hickory  = evt.target.ownerDocument.getElementById("hickory");
        dickory = evt.target.ownerDocument.getElementById("dickory");
        dock = evt.target.ownerDocument.getElementById("dock");

        ShowAndGrowElement();
    }
    function ShowAndGrowElement() {
        timevalue = timevalue + timer_increment;
        if (timevalue > max_time)
            return;
        // Scale the text string gradually until it is 20 times larger
        scalefactor = (timevalue * 650) / max_time;

        if (timevalue < 30) {
            hickory.setAttribute("display", "");
            hickory.setAttribute("transform", "translate(" + (600+scalefactor*3*-
        }
```

Figure 1.3: Skywriter

Then It developed a newer version which was called sky writer, In January 2011 the skywriter was merged into ace.org and the cloud9IDE , the first web-based IDE which supported only web languages ,Our idea is to make a web-based IDE for desktop programming with the ability to run the software online.



```
/**
 * @copyright 2010, Ajax.org Services B.V.
 * @license GPLv3 <http://www.gnu.org/licenses/gpl.txt>
 */
var net = require("net");
var sys = require("sys");
var NodeSocket = require("v8debug/NodeSocket");
var StandaloneV8DebuggerService = require("v8debug/StandaloneV8DebuggerService");

module.exports = DebugProxy = function(port) {
    process.EventEmitter.call(this);
    var _self = this;

    this.connected = false;

    var socket = new NodeSocket("localhost", port);
    socket.onend = function() {
        this.connected = false;
        _self.emit("end");
    };
    this.service = new StandaloneV8DebuggerService(socket);

    this.service.addEventListener('connect', function() {
        _self.connected = true;
        _self.emit("connection");
    });
    this.service.addEventListener('debugger_command_0', function(msg) {
        _self.emit("message", msg.data);
    });
};

sys.inherits(DebugProxy, process.EventEmitter);

(function() {
```

Figure 1.4: Cloud9IDE

1.1.2 **Applications**

There are a huge number of applications to this project for example:

1- Programming Learning.
2- Cloud-Based Coding.
3- Running, Testing, Debugging Online.
4- Real-Time Collaboration.
5- Sharing Projects.
6- Agile Methods Improvement.
7- Extreme Programming Improvement.
8- Unit Testing Concept Improvement.
9- Code Review Concept Improvement.

And so on a lot of applications and concept improvements can came along with a project like this one because it opens the door to a whole new world of software developing concepts , also it’s useful to focus the developer’s mind and power to the software production process instead of filling it with configurations and other non-critical Processes ,

1.2. **Problem Definition**

The Problems that motived us to work in this project are divided in to 4 parts:

1-The rapid developing of smaller devices, more portable ones that uses small hardware which is to this point is not powerful enough to hold IDE software for the developers,
Even if the device can install an IDE on it, it will not have the full processing power to run the developed software needed.

2-Thelimitions of the version management software which allows the users working in the same project requires a lot of configurations and it's unstable.

3- The growing need of more computational and processing power which can only be found within the internet server that provides a lot of power and space to the internet though not all of it is being used to the most usefulness of it.

4-The need of a way to take the software development process to the cloud so that it can be easier and more stable.

1.3. **Objectives**

The goal is to create an online frame work that allows the software development process to take place online allowing the developers to easily communicate , share , develop , code , host ,edit ,run all in one place, thus eliminating the need to configure a lot of software to do each task independently , and go through a lot of trouble shooting and configuration and it may fail sometimes making the whole process to stop for no good reason . providing the ability to access your projects anytime anywhere from any device regarding the platform of the device that you use.

1.4. **Time plan**

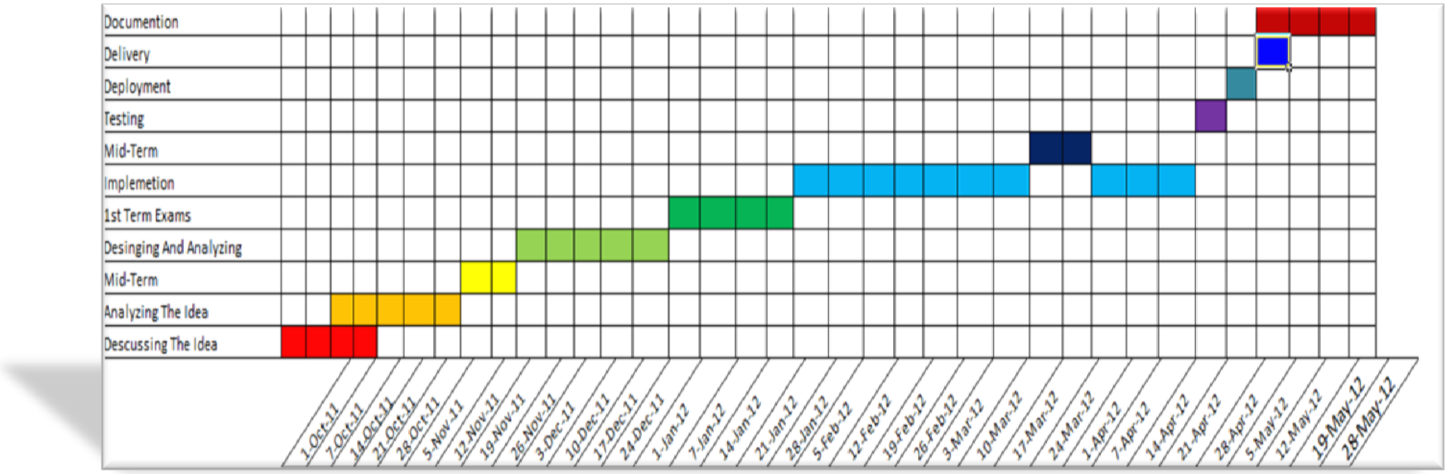


Figure 1.5: Time plan

This time plan shows the work done by the working team to do this project.

1.5. **Documentation Outline**

Chapter 2: Background

This chapter represents the background that should be known for anyone who reads this documentation and the algorithm that have been used to create this project including some information about it.

Chapter 3: System Design and Analysis

This chapter concludes the system design and analysis including the classes used to make this project and how every class connects to the other.

Chapter 4: System Implementation

This chapter shows the main components of the system and how it has been made.

Chapter 5: System Testing and Deployment

This chapter shows how the system been tested and deployed to the server and other computers and how to user it.

Chapter 6: Conclusion and Future Work

This chapter shows the future of this project and how to make use of it in other parts and what is it expected to be and if someone will be developing this project what point should be started with

Chapter Two:

Scientific Background

2.1 **History & Problems**

In this chapter we will focus on the history and problems in this field and a solution for each.

2.1.1 **Portability**

Can access files and work anywhere.
Solution: Online Storages.

2.1.2 **Work Anywhere**

Work without being chained to specific platform that has your programs and tools installed, just with internet connection can work anywhere.
Solution: appeared simple solution for some languages, some in the IDE's for these languages and other as a web service, some websites provide the ability to code a lot of languages with a syntax highlighting and coloring for reserved words, and then for some languages it can the code in the server side and send back a runnable file like (exe) in windows.

2.1.3 **Organizing Versions of a large Project**

While working in a multi-developer project, it needs frequent updates form and to all developers with the latest updates, logging all changes.
Solution: Using SVN (Software Version Number) Servers.

2.1.4 **Working Simultaneously**

This is the feature is a main target in our project, easing the work for developers.
Solutions: The RTC (Real Time Collaboration), which give the ability for multi-developers to change simultaneously in a project even in the same page, which can be used in developing or learning as well.

Each problem had a solution for it; our project is to gather all solution in easy to use service.

2.2 **Previous Technologies**

2.2.1 **XMPP Standard Foundation**

The Extensible Messaging and Presence Protocol (XMPP) is an open technology for real-time communication, which powers a wide range of applications including instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data. The technology pages provide more information about the various XMPP “building blocks”.

The core technology behind XMPP was invented by Jeremie Miller in 1998, refined in the Jabber open-source community in 1999 and 2000, and formalized by the IETF in 2002 and 2003, resulting in publication of the XMPP RFCs in 2004 (see the history page for more details).

Although the core technology is stable, the XMPP community continues to define various XMPP extensions through an open standards process run by the XMPP Standards Foundation. There is also an active community of open-source and commercial developers, who produce a wide variety of XMPP-based software.

2.2.2 **Ignite Real Time**

Open source software, open, collaborative minds are developing so that they would improve the software landscape.

The perspective is that base functionality – the core messaging aspects of RTC should be free and open source.

The community: Igniterealtime is a professionally-led (from Jive Software) Open Source community composed of end-users and developers around the world who are interested in applying innovative, open-standards-based Real Time Collaboration to their businesses. We're aimed at disrupting proprietary, non-open standards-based systems and invite you to participate in what's already one of the biggest and most active Open Source communities anywhere.

2.1.3 Eclipse ECF (Eclipse communication Framework)

ECF is a framework for building distributed servers, applications, and tools. It provides a modular implementation of the **OSGi 4.2 Remote Services** standard, along with support for **REST-based** and **SOAP-based** remote services, and **asynchronous messaging for remote services**.

ECF/DocShare Plugin:

ECF version 2.0.0 and up includes a plugin called DocShare (**org.eclipse.ecf.docshare**) that implements real-time shared editing.

Real-Time Shared Editing over XMPP:

The ECF project has been adding some real-time collaboration functions for dev team support...for example, sharing selected text in Eclipse editors, as well as real-time collaborative editing...all over the ECF datashare API...which means you can use your favorite public IM accounts like google talk, or skype rather than be forced to have yet another account. We've also got some new sharing of Mylyn tasks as per bug 195737.

RT Shared Editing:

Motivation for the RT Shared Editing, which is dubbed "**Cola**" (**Col**l**a**bode) for now, is supposed to be a tool enabling developers to reap the benefits of pair programming within the Eclipse IDE.

The term *pair programming* describes an activity in which two developers simultaneously work on a single development machine.

2.2.4 **Google Wave**

Google Wave is a real-time collaboration system based upon operational transformation approach to replicated state synchronization. With the Cola System (DocShare), ECF has been using operational transformations, for some time now. I will implement a provider to allow an equinox+ecf based web server to inter-operate with Google Wave.

With an announcement on 08/04/2010 Google discontinued the development of Google Wave as a standalone product. Anyway, an open source implementation of the Google Wave server is available and the protocol is also open. The Google Wave (aka Federation One) server comes with an implementation of the operational transformation algorithm, which allows you to build real time shared editing applications from the comfort of your eclipse RT / ECF environment using this provider.

Implementation of a Google Wave ECF provider that people can use to build their own wave applications on top of ECF. The provider will handle the basic wave-protocol operations like managing waves, contacts and documents. Also the provider will provide an API which allows users to add listeners to wave changes and implement real time shared editing applications.

2.2.5 **Etherpad**

Etherpad is a hosted web service that allows really real-time document collaboration for group of users.

All editing of the document is instantly visible on the screens of all participating users, enabling new and productive ways to collaborate on text documents.

Etherpad is useful for meeting notes, drafting session, education, team programming, and more.

EtherpadLiteDotNet:

An implementation of the Etherpad Lite API for .NET written in C#

It implements the Etherpad Lite API, more infomation on the API can be found on the Etherpad Lite wiki.
It matches the API in structure as closely as it can.
It parses the returned JSON into strongly typed objects to allow the use of intellisense and reduce the use of magic strings.
The library is written in C# and uses .Net Framework 4.

2.3 RTC Applications

2.3.1 CodeRun

A free, Cross-Platform browser-based IDE

CodeRun Studio is a cross-platform Integrated Development Environment (IDE), designed for the cloud. It enables you to easily develop, debug and deploy web applications using your browser.

CodeRun Studio can be used instead or alongside your existing desktop IDE. You can upload existing code in order to test it in the cloud or for sharing with your peers.

2.3.2 IDEOne

Ideone is something more than a pastebin; it's an online compiler and debugging tool which allows
to compile and run code online in more than 40 programming languages.

ideone.com is designed mostly for programmers (but, of course, common plain text can also be uploaded). You can use it to:

- share your code (that's obvious - it is a pastebin, isn't it? :)) in a neat way,
- **run your code** on server side **in more than 40 programming languages** (number still growing)
- and do it all **with your own input data!**
- ideone.com also provides free Ideone API which is availabe as a webservice. Its functionality allows you to build your own ideone-like service!
- for logged in users Ideone offers possibility to manage their codes, publish multiple submissions at one go, and more.

All codes can be accessed through convenient hash links. Source code pages provide information about the code and its execution: memory usage, execution time, language and compiler version, code itself, input uploaded by the user, output generated by the program and error messages from compilers and interpreters.

2.3.3 **CodePlex**

CodePlex is Microsoft's free open source project hosting site. You can create projects to share with the world, collaborate with others on their projects, and download open source software.

2.3.4 **ECCO**

is an online programming development environment with file manager, console interface for compiling and execution of code and an editor with syntax highlighting, tabs and support for many languages which is accessed using a browser.

Main goals

- E-learning + interactive tutorials

  Students can easily learn new languages through examples or interactive tutorials;

  Students can learn basic linux commands;

  Teachers can track students progress;

- Instant development and code

  Begin to code without spending time with configurations and environments;

  Test your code instantly;

- Access your code from anywhere

  As an online service, it can be accessed from anywhere;

- Must be easy to use

2.3.5 **CollabEdit**

Collabedit is an online code editor that lets people collaborate in real-time.

It works in your web browser so no installation is needed.

Features:

- text editor
- chat
- document history
- syntax highlighting for programming languages

Collaborative Coding:

Email and instant messaging do not work well for sharing code. They don't preserve whitespace, the fonts aren't monospaced, the spell check gets in the way, etc.

Paste bins are better, but they are too static if you're looking for collaboration. Also they don't have a good editor, they're only good for copy and paste.

Collabedit has what you need - a good editor, syntax highlighting, real-time collaboration, chat and versioning.

Teach Programming:

Teaching demands a fine balance between letting your students do, and showing them how.Collabedit allows student and teacher to share a common workspace without the distractions that other tools impose.You don't need the hassle of exchanging emails with bad formatting. Whiteboards do not work very good for code, neither does Power Point. The pain of saying "closing parenthesis" out loud is no longer necessary.

Now you can teach more effectively with collabedit.

2.3.6 **ShiftEdit**

ShiftEdit is an online IDE for developing PHP, Ruby, Python, Perl, HTML, CSS and JavaScript via FTP, SFTP and Dropbox

"The online IDE is one of the final frontiers of apps ported to the web. I would like to be able to develop from any computer or operating system and have the same experience without having to install software or create site definitions." - Adam Jimenez (founder)

2.3.7 **Gobby**

Gobby is a free collaborative editor supporting multiple documents in one session and a multi-user chat. It runs on Microsoft Windows, Mac OS X, Linux and other Unix-like platforms.It uses GTK+ 2.6 as its windowing toolkit and thus integrates nicely into the GNOME desktop environment.

2.3.8 **Sync.in**

Sync.in is a web based word processor for people to collaborate in real-time.

When multiple people edit the same document simultaneously, any changes are instantly reflected on everyone's screen. The result is a new and productive way to collaborate with text documents, useful for meeting notes, brainstorming, project planning, training, and more.

And more (ex: CakePHP, CodeAcademy).

2.4 Collaborative Editing with semantics

Collaborative editing of source code files presents the particular problem that program errors introduced by one collaborator may disrupt the progress of their colleagues. Source files must conform to a particular programming language syntax and semantics, but it is impossible to expect users to maintain semantic validity throughout the coding process: almost any partially written expression, statement, block, method, class, or program is likely to contain errors according to the parser or compiler. Structured editors [17] address this problem only at the significant cost of requiring convoluted steps to make certain simple changes [28], and most programmers do not use them.
To see how errors introduced by one user can make it more difficult for another user to continue, consider the following simple program.

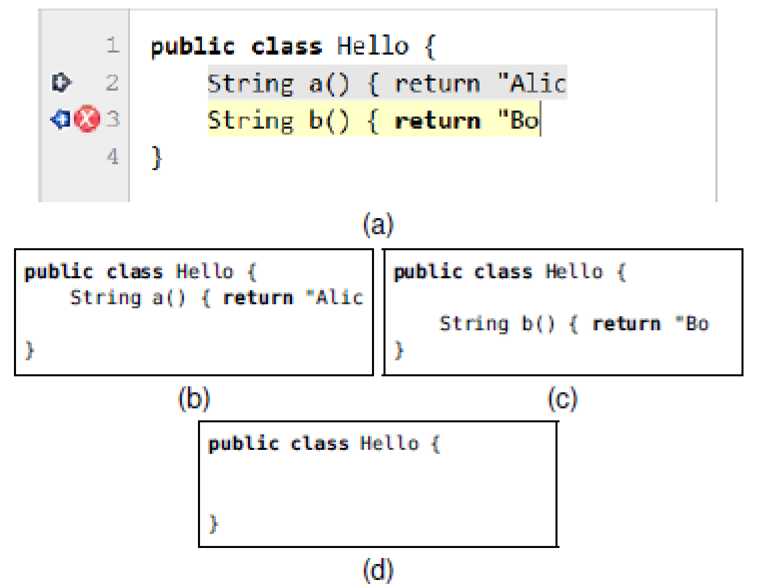
```
class Hello {
int x;
void a() { x = true; }
}
```

This contains an error, since the integer field x cannot be assigned a boolean value. However, suppose another user begins defining a new method:

```
class Hello {
void b() {
int x;
void a() { x = true; }
}
```

At this point the compiler will report instead that “x cannot be resolved to a variable,” masking the true error because x now appears to be a local variable of method b. The particular set of failure cases and their likelihood and severity will depend on the compiler. Compiler attempts to prevent common problems such as unbalanced braces from generating opaque errors elsewhere, but it cannot always succeed. The problem is much worse in the case of running the program: although Compiler will allow the programmer to launch an error-containing program, any attempt to execute an errorcontaining line will result in failure. This has the potential to prevent collaborating programmers from writing and testing their code incrementally, and we have observed this problem during pilot studies with both Python and Java programmers.
Rather than constrain collaborating programmers to agree on and reach points at which the program is sufficiently errorfree for them to run it and test code in

progress, we instead account for the semantic content automatically in our synchronizing behavior.



```
public class Hello {
    String a() { return "Alic
    String b() { return "Bo
}
```

(a)

```
public class Hello {
    String a() { return "Alic

}
```

(b)

```
public class Hello {

    String b() { return "Bo
}
```

(c)

```
public class Hello {


}
```

(d)

Figure 2.1:Unit-Real-Time Collaboration ,Alice and Bob are working concurrently to define methods a and b. (a) The union of their changes is displayed in the editor, shown here from Bob's perspective: his unintegrated change is in yellow, with an outgoing arrow; Alice's is in gray, with an incoming arrow. (b) Alice's working copy and (c) Bob's working copy contain only their own changes. (d) The disk does not reflect their workinprogress, since both of their changes still contain errors.

Collabode addresses this issue by first giving each programmer a separate, persistent working copy of the program, and then maintaining two versions that integrate programmers' changes from their working copies: a disk version and a union version. The union version is the text that users see and manipulate, and it contains all edits applied by all users, with highlighting and icons to indicate provenance. So if Alice has begun defining method a and Bob is writing method b, the union version contains both in-progress methods, and both methods appear in the user interface (Figure 2a). As long as their methods contain compilation errors, the working copies of Alice and Bob will each contain only their own method (2b-c) and the disk will contain neither (2d). Once their methods compile, their edits will be shared both with their collaborator's working copy, and with the disk version, which corresponds to the content on disk. It is this disk version that is run when either programmer elects to run the program. This version is always free of compilation errors.

In cryptography and computer science hash trees or Merkle trees are a type of data structure that contains a tree of summary information about a larger piece of data – for instance a file – used to verify its contents. The concept is named after Ralph Merkle.

Concurrent Hash Map

Hash trees are a combination of hash lists and hash chaining, which in turn are extensions of hashing. Hash trees in which the underlying hash function is Tiger are often called Tiger trees or Tiger tree hashes.

- Memory overhead of a ConcurrentHashMap (compared to a "classical" HashMap)

```
public static void main(String... args) throws NoSuchMethodException,
IllegalAccessException {
    for (int i = 0; i < 4; i++) {
        long used1 = usedMemory();
        populate(new HashMap());
        long used2 = usedMemory();
        populate(new ConcurrentHashMap());
        long used3 = usedMemory();
        System.out.println("The ratio of used memory is " + (double) (used3 -
used2) / (used2 - used1));
        System.out.println("For an extra " + ((used3 - used2) - (used2 - used1))
/ 1000000 + " bytes per entry was used.");
    }
}

private static void populate(Map map) {
    for (Integer i = 0; i < 1000000; i++)
        map.put(i, i);
}

private static long usedMemory() {
    return Runtime.getRuntime().totalMemory() -
Runtime.getRuntime().freeMemory();
}
```

Chapter Three:

Analysis and Design

3.1 **Overview**

Our System has three modules (See Fig. 3.1):

- Client

  Web Based-IDE is a cross-platform Integrated Development Environment, Designed for the cloud.

  Website management and handle all user interface with Server.

- Client-Server

  This module can managements all connections between client and server and Encrypt/Decrypt all data that interchange between them.

- Server

  Other hand , Server will be able to run ,debug and deploy web applications when user send his data after encryption will recompile it and send result again to Client-Server to show it for user at Client module.

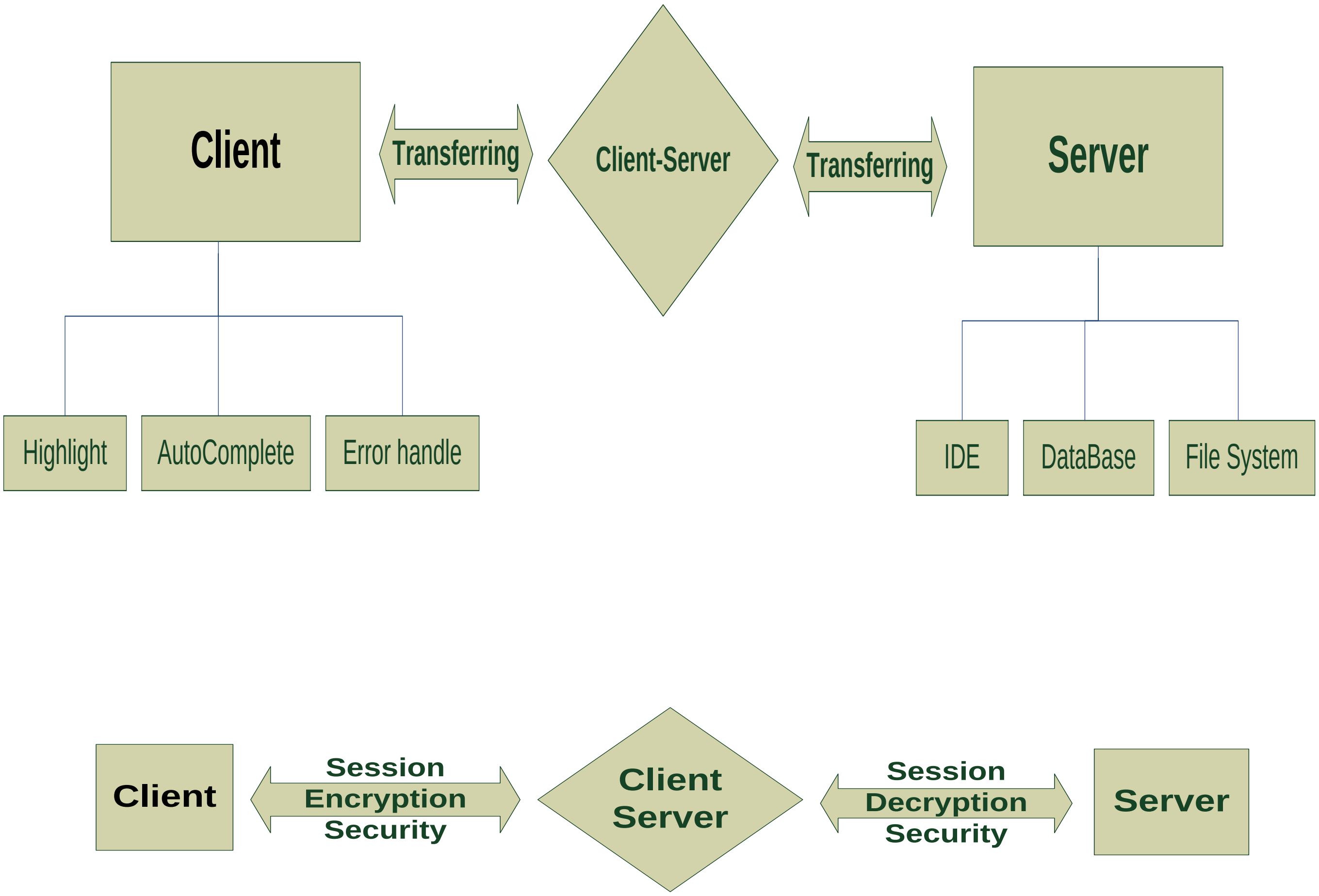


Figure 3.1: Overall System Architecture

3.2 **Use Cases**

Name: Register
Identifier: UC01
Actor: Visitor
Description: Register into the website to have a member account.
Pre conditions: The user must not already have an account.
The account username should be unique.
Post conditions: The user becomes a member and is treated as a member by the system.
Basic Course of Action: The user inputs his/her username, desired password, real name, age, country.
An account is created for the user.
Alternative Course A:
1. The user already registered
2. The user is informed that he is already registered or the username s / he has chosen is already taken.
3. The user can refill the registration form once again.

Name: Login
Identifier: UC02
Actor: Visitor
Description: Login to the website.
Pre-conditions: The visitor has to be a registered member.
Post conditions:
Basic Course o f Action:
1. The user presses on the Login button.
2. The user enters his authentication information.
3. The system checks the validity of this information.
4. If the information is valid, the visitor become logged in and is treated as a member by the system.
Alternative Course A: Authentication information not valid
1. The system rejects the request to login.
2. The user can input his/her information once again.
Alternative Course B: The user forgotten his/her password
1. The user presses the Forgot your Password button.
2. The user inputs his e-mail.
3. The system checks if there’s an account corresponding to this e-mail.
4. The system sends the username and password of the corresponding account to the inputted e-mail.

Name: Logout
Identifier: UC03
Actor: Member
Description: Logout o f the website.
Pre conditions: The user has to be logged in.
Post conditions: The user is treated by the system as a visitor not a member.
Basic Course o f Act ion:
1. The user presses on Logout button.
2. The user becomes no longer logged on and is not treated as a member anymore.

Name: Delete Account
Identifier: UC04
Actor: Member
Description: The user deletes his/he account
Pre conditions: The user must be logged in.
Post conditions: The user account is deleted.
The users Files and their transcripts are removed from the index.
Basic Course o f Action:
1. The user his/her account page.
2. The user presses the Delete button.
The system removes the users Files and their transcripts from the index.

Name: Create Project
Identifier: UC05
Actor: Member
Description: The user create new project
Pre conditions: The user must be logged in.
Post conditions:
Basic Course o f Action:
1. The user presses on the New Project button.
2. The user enters his Project information.
3. The system checks the validity of this information.
4. If the information is valid, The System will Create New Project

Name: Run Project
Identifier: UC06
Actor: Member
Description: The user debug current project
Pre conditions: The user must be logged in.
The Editor shouldn’t be empty.
Post conditions:
Basic Course o f Action:
1. The user presses on the Debug button.
2. The system sends code to server.
3. The Server Recompile the code and check if validate
4. If the code validate, the user will receive output otherwise will show error list.

Name: Delete Project
Identifier: UC07
Actor: Member
Description: The user Delete current project
Pre conditions: The user must be logged in.
Post conditions: The Project is deleted.
Basic Course o f Action:
1. The user presses on the delete button.
2. The System send request to server for delete project and all file of this project
The system removes the Project Files and their transcripts from the index.

3.3 **Sequence Diagrams**

A sequence diagram shows, as parallel vertical lines ("lifelines"),
Different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

1- Sequence diagram for registration:

User enters his/her data in the form; system checks the database for the entered user name if not used before then the system create new account to the user, if used before or there is an error in the entered data system return error message and redisplay the registration form.

2- Sequence diagram for create new project:

User presses create new project button and enter required information about new project as name of project and descriptions about it, then system will sends request to server with data of user to create new project, if same name is exist return error message and required to enter different information about new project otherwise if data is accepted by server will return message that the project was created successful.

3- Sequence diagram for run/debug project:

User presses run button, then system will collect date of user and his code from editor to encrypt and send it to server.
Then server check validate of data and decrypt it ,so it will recompile it if there is an errors will return message box to user to debug it otherwise will send output of code to show it to user.

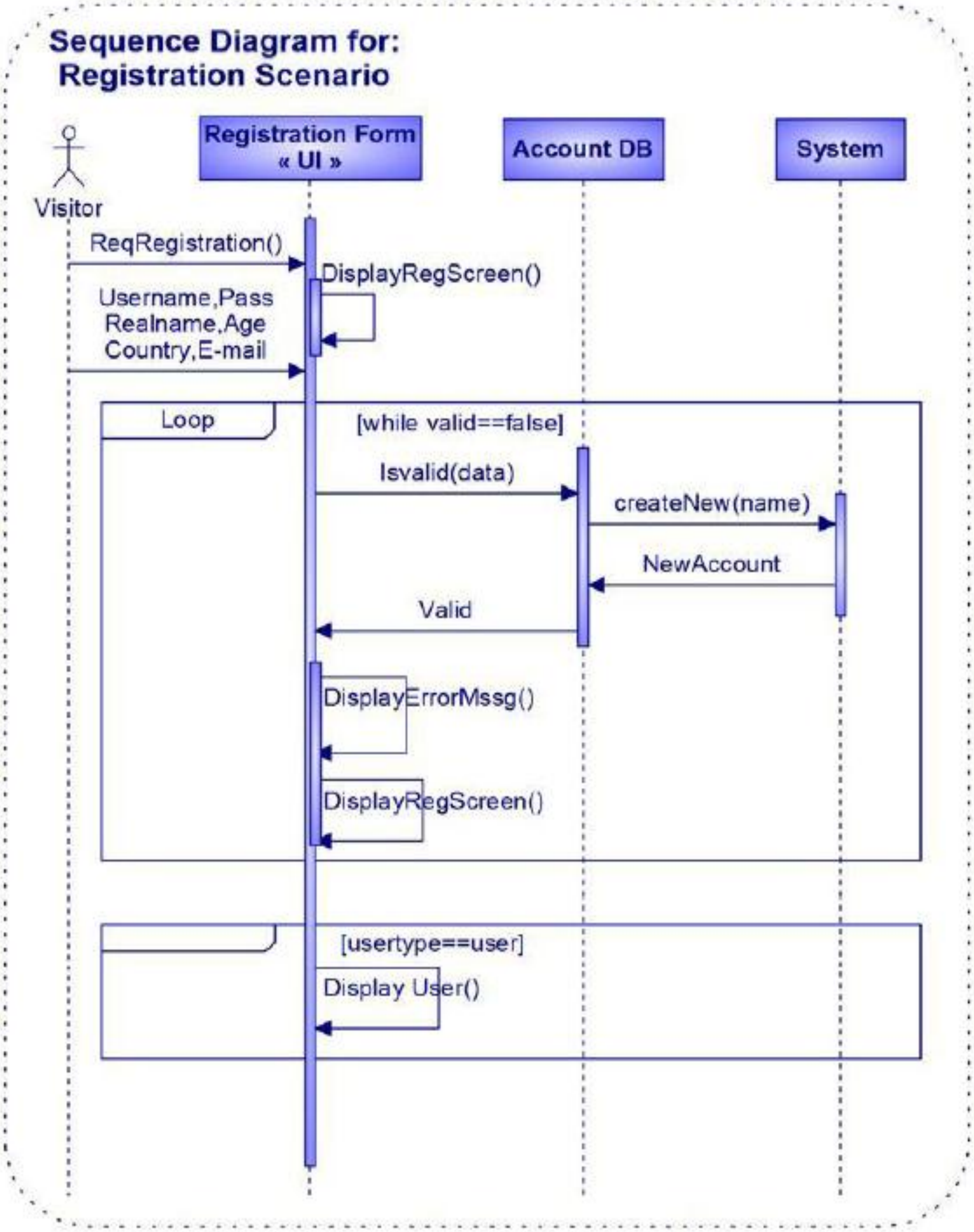


Figure 3.2: Sequence Diagram for Registration Scenario

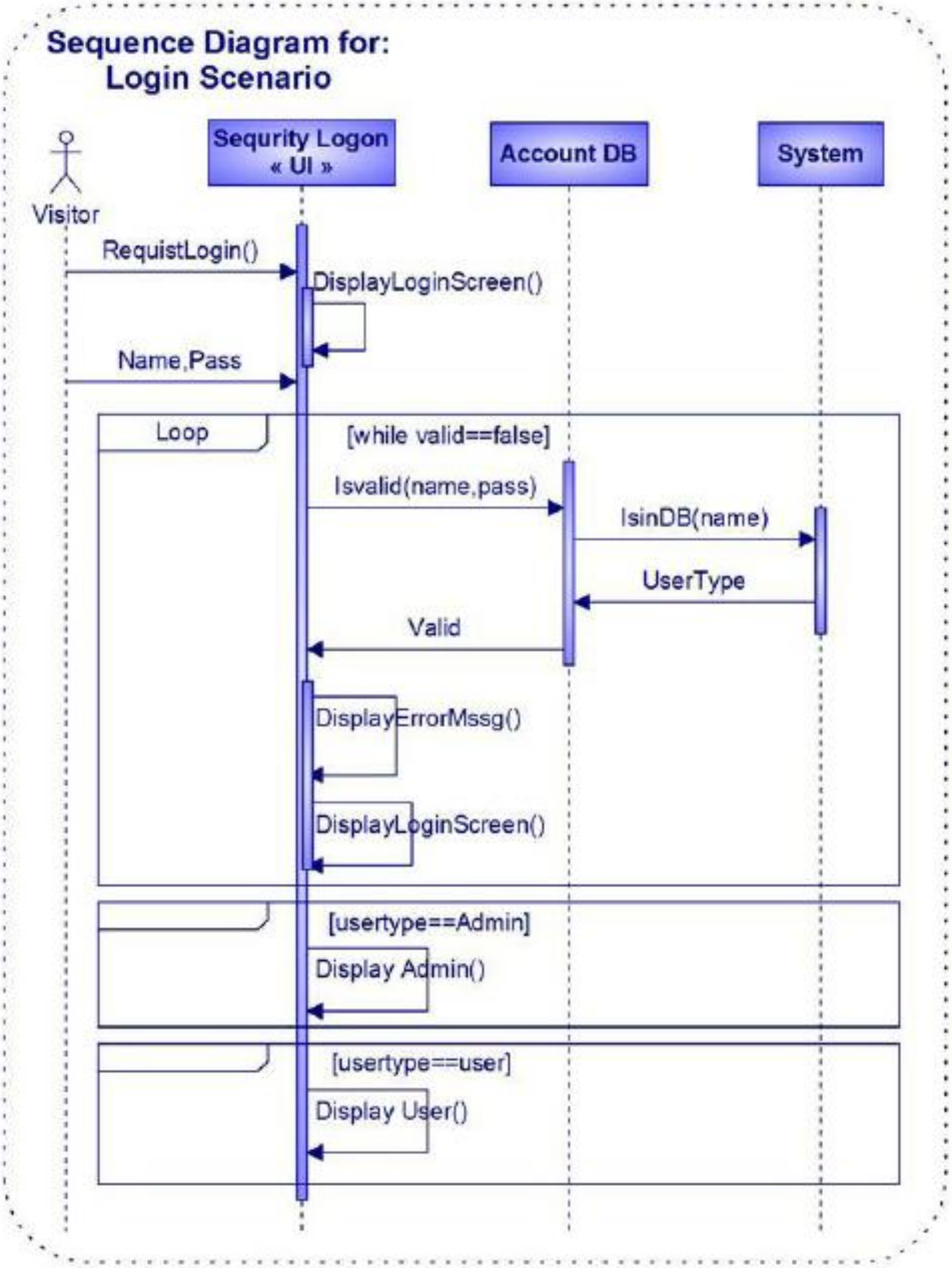


Figure 3.3: Sequence Diagram for Login Scenario

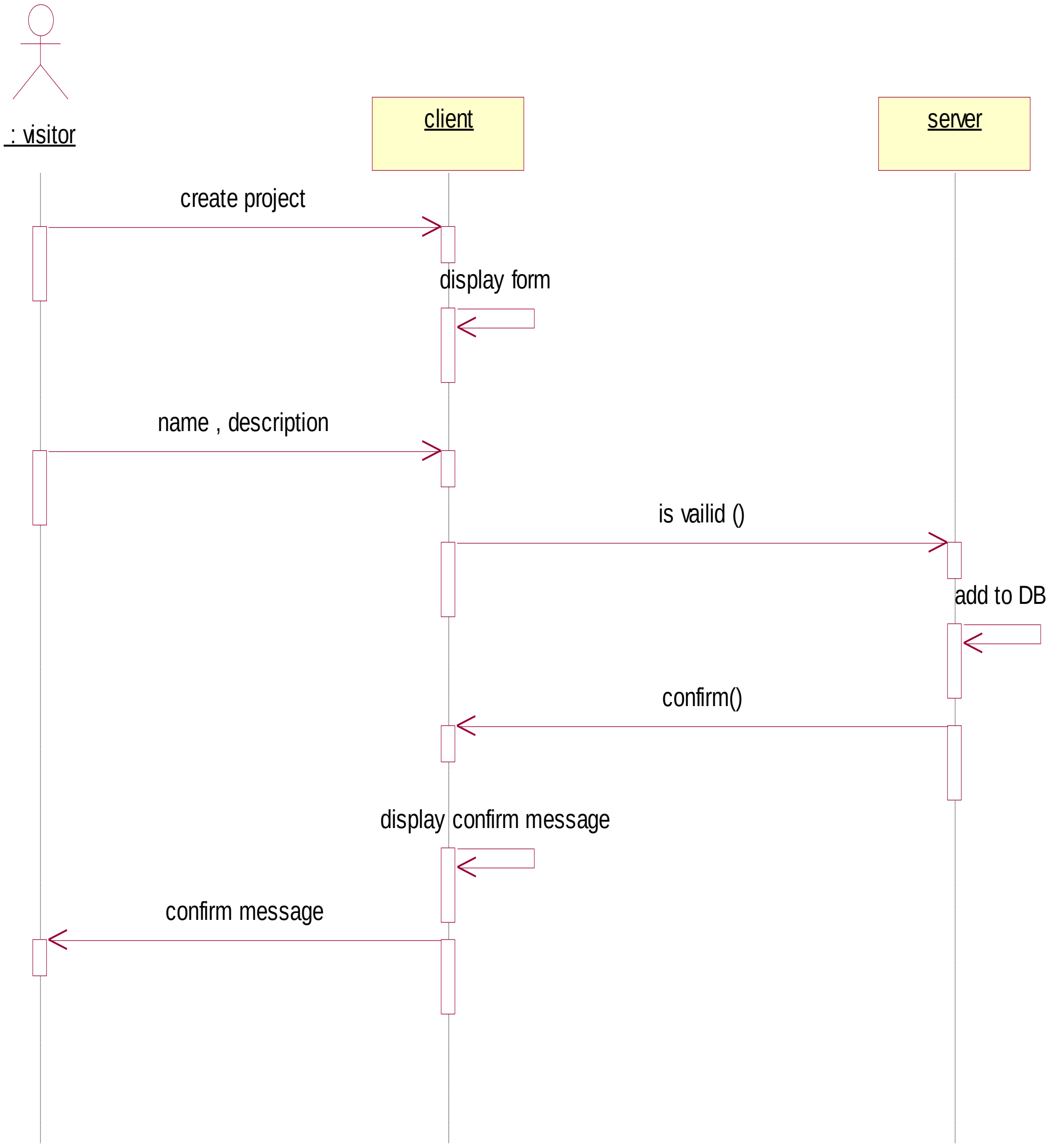


Figure 3.4: Sequence Diagram for Create project Scenario

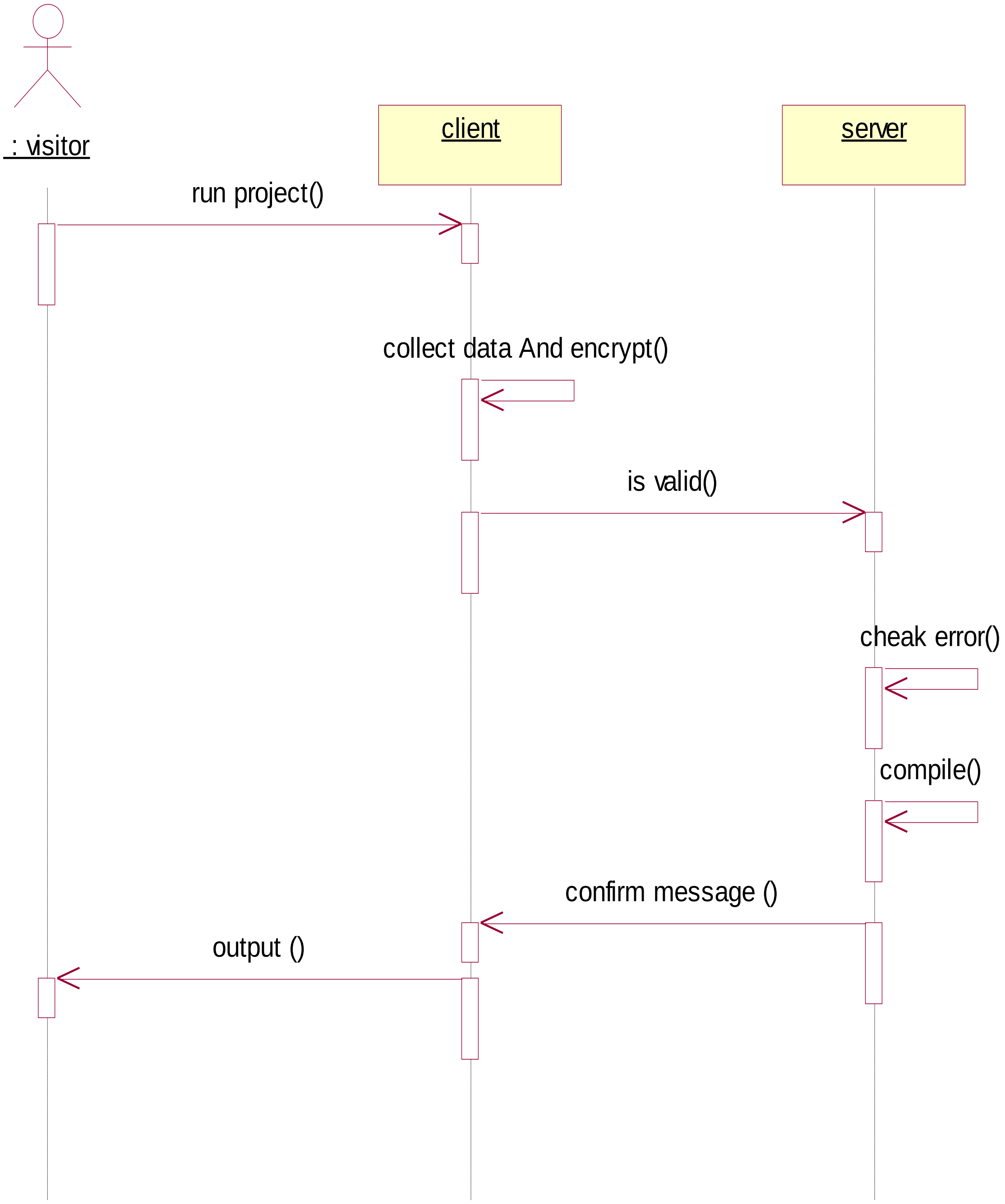


Figure 3.5: Sequence Diagram for Run project Scenario

3.4 **Activity Diagrams**

Activity diagrams are a loosely defined diagram technique for showing workflows of stepwise activities and actions, with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

1) Activity diagram for login:
   a. Enter username and password
   b. System check
   c. If valid then manage files and his account
   d. Else enter username and password again

2) Activity diagram for register:
   a. Enter his/her data
   b. System check
   c. Return new account
   d. Manage files and manage account

3) Activity diagram for create new project:
   a. Press create new project button
   b. Enter his/her information about project
   c. System check
   d. If valid manage project

4) Activity diagram for run/debug project
   a. Enter or Edit his code
   b. Press debug button
   c. Server compile code
   d. If valid ,send output of code
   e. Otherwise show error list

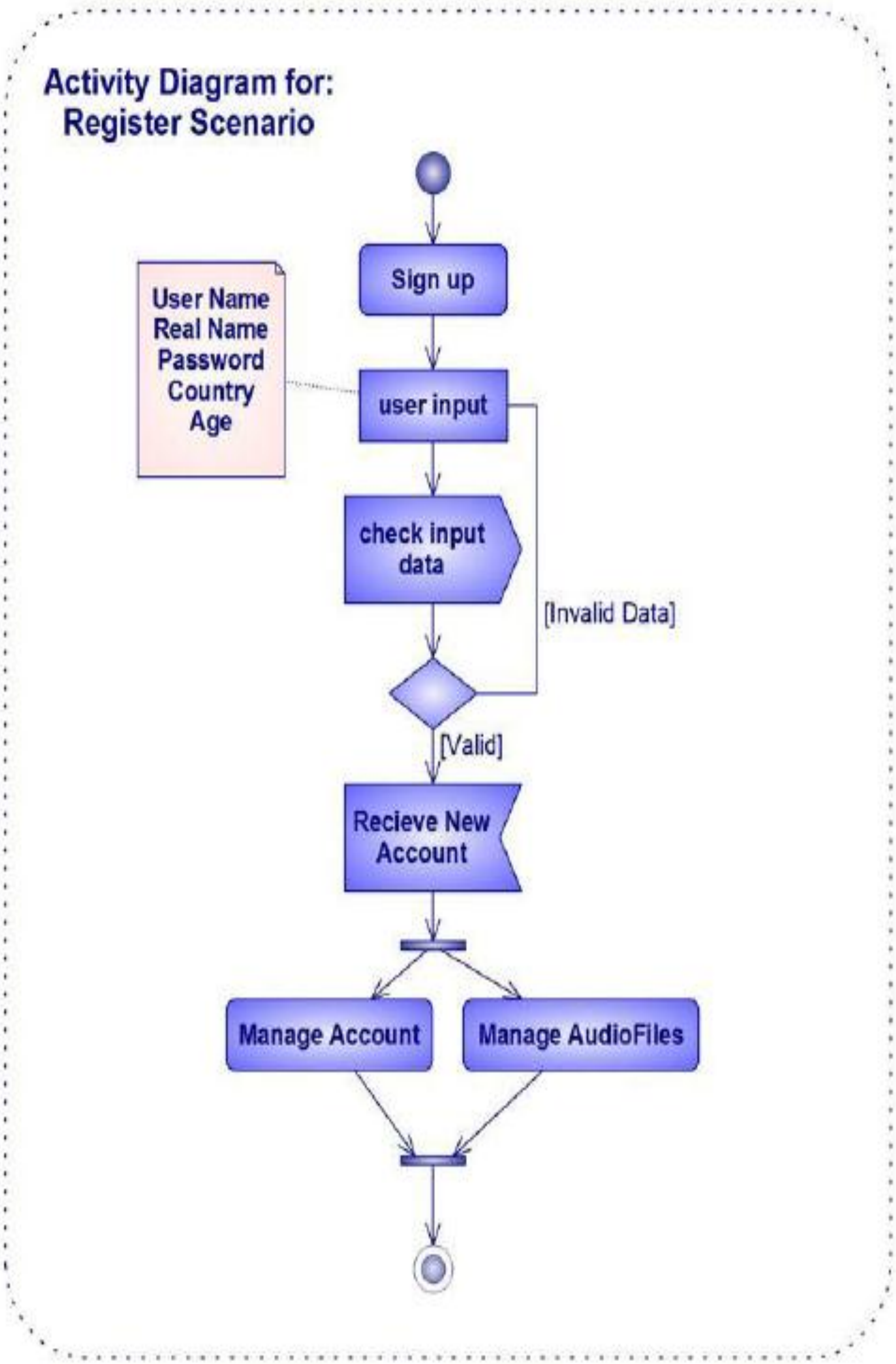


Figure 3.6: Activity Diagram for Register Scenario

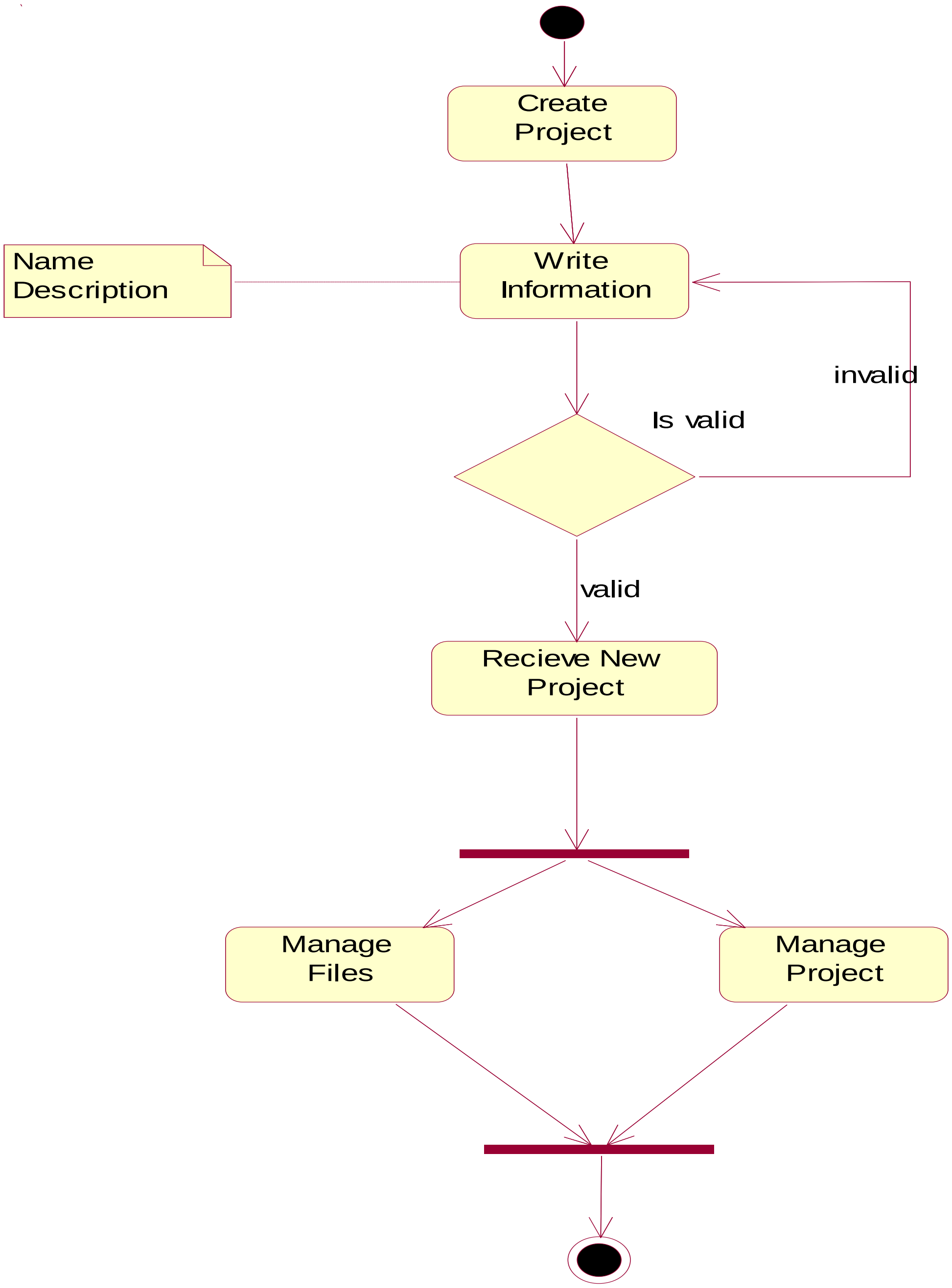


Figure 3.7: Activity Diagram for Create Project Scenario

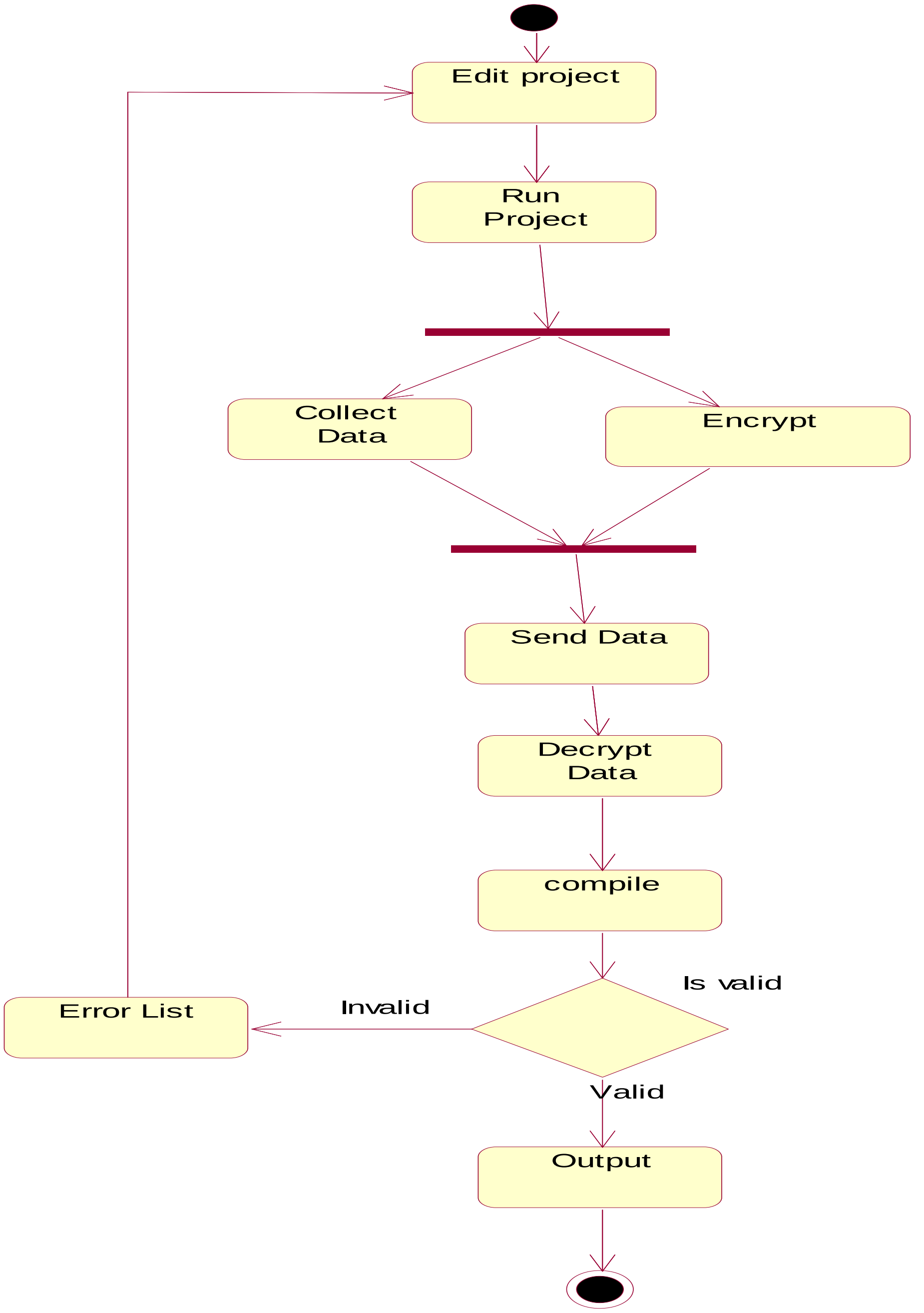


Figure 3.8: Activity Diagram for Run Project Scenario

3.5 **Class Diagram**

Class diagram in the Unified Modeling Language (UML), is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

1) Login:
    a. Check login data function.
    b. User Account

2) Register:
    a. Check unique username
    b. Create new account

3) Profile:
    a. Edit profile
    b. Delete account
    c. Add/Edit/Delete project files

4) Database:
    a. Contain all information about users and files

5) Create Project
    a. Check data
    b. Create new project

6) Run Project
    a. Check error list
    b. Compile

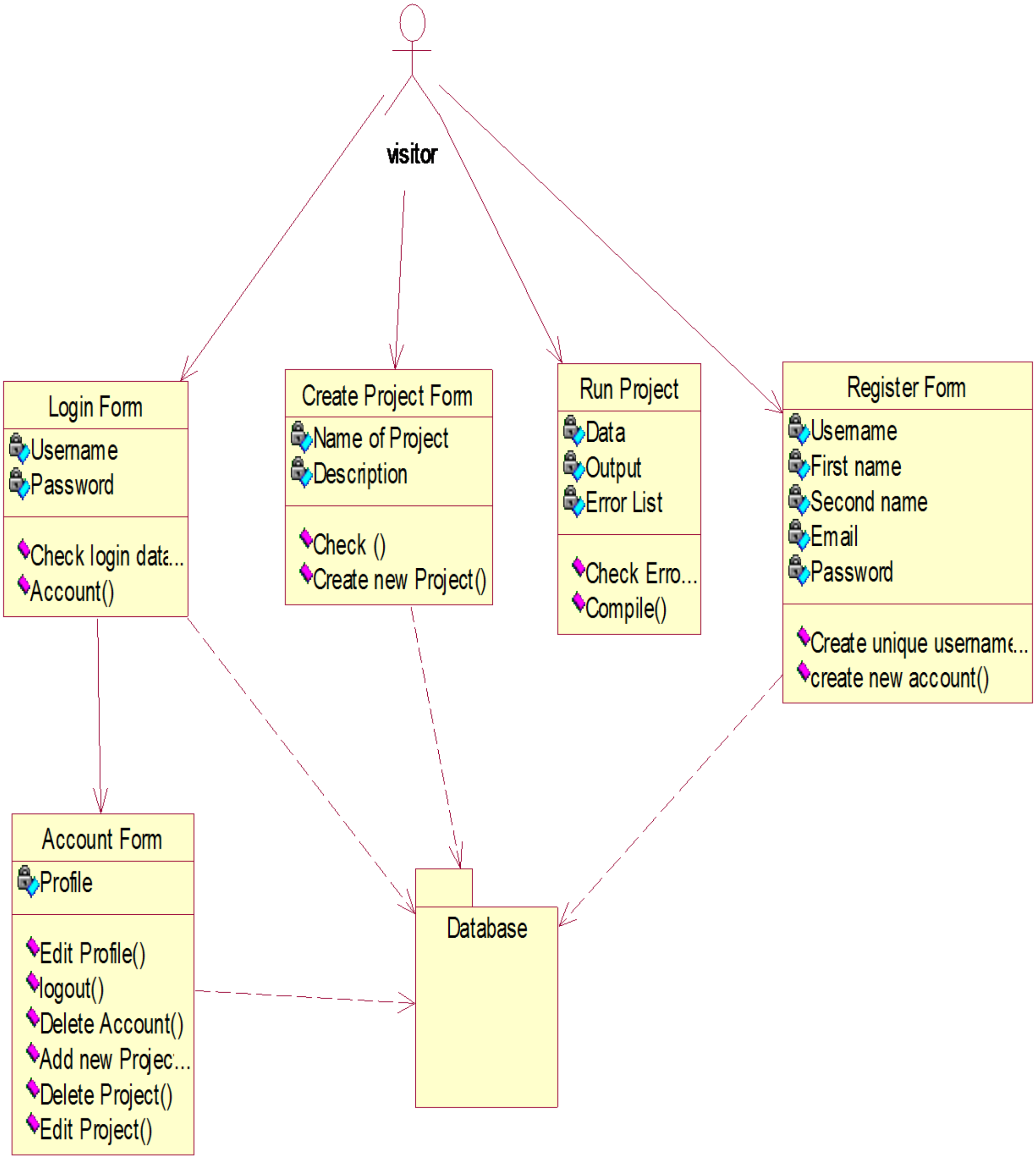


Figure 3.9: Class Diagram

Chapter Four:

System Implementation

Our web-based IDE system is divided into 3 main parts using the design pattern of model view control:

1- Model
2- View
3- Control

Each part will be discussed in details through the chapter

4.1 **Views**

As the project indicates that the project will run on a web browser so the view part will be HTML pages that will be displayed to the users.

For the ease of implementation a library was used to make templates of html pages and populate it with different data this library is called embedded javascript (EJS) , which allows clean developing of html pages and function calling and passing a lot of data to display in the html.

4.1.1 **EJS files**

Template files that holds the view of every web page in the project given the data

File	Function
acl.ejs	Permission template that will be shown to the user to help him set permission to his projects.
java.ejs	Lunch template that will be shown to the user when he press Run with the console part.
launch.ejs	Lunch template that will be shown to the user when he press Run.
acl_delete.ejs	Remove Permission template that will be shown to the user to help him removes permission from his projects.
console.ejs	Console template that will be shown to the user when he press Run.
file.ejs	File template that will be shown to the user when To manage his files.
folder.ejs	Folder template that will be shown to the user when To manage his Folder .
login.ejs	Login template that will be shown to the user when To login and access his projects .
none.ejs	Empty template that will be shown to the user when the users try to access an empty directory .
path_delete.ejs	Confirm Deletion template that will be shown to the user when the users deletes a resource from his project.
project.ejs	Configuration template that will be shown to the user when the users wants to edit the project settings.
project_create.ejs	Create project template that will be shown to the user when the users creates a new project.

settings.ejs	User settings template.
wizard.ejs	Explorer template that will be shown to the user in the side bar when he tries to add resources to the project from this side bar.
403.ejs	Error template that will be shown to the user when the users edit the path of the headers to a non-authorized path .
404.ejs	Error template that will be shown to the user when the users edit the path of the headers to a non-existing path .
project.ejs	Project Template.
marker.ejs	Error in code template that will be shown to the user when the users tries to run the project and the project have error in it’s code.
create.ejs	Create template that will be shown to the user when the users creates a new file or folder or sub-folder.
docbar.ejs	Bar template that will be shown to the user in the main pages indicates the current logged user.
filegrid.ejs	File template that will be shown to the user when he wants to view his files in one project.
html.ejs	Root page that holds the css and javascript basic functions.

Table 4.1: ejs files

4.1.2 **Helpers file**

This page mainly uses functions form the from helpers file which contains a lot of functions that would be used through the project to do certain tasks such as including the css and displaying the data on a certain formats , with render view , render partial , and render list , some other functions also is added such as :

Function Name	Parameters	Functionality
includeJQuery()		Includes the famous library of JQuery
includes(path)	path to javascript file	Includes a javascript file to this page to be used with this content
includeCss(path)	path to CSS file	Includes a CSS file to this page to be used with this content.
htmlTitle()		Sets the current web page title.
bodyId()		Returns the ID of a given element of the HTML page.
baseHref()		Returns the base url of the current page
cssIncludes()		Sets the css of every element in the web page
renderView(name, item)	Name : ejs template name Item : the page which will be rendered	Sets the rendering template of ejs to this page
renderPartial(name, data)	Name : ejs template name data : the item which will be rendered	Sets the rendering template of ejs to items in this page
_hd()		Main function it generates the web pages and return them in array format for the browser to process

addBodyClass(class)	Class : adds new item to the view page	Adds and joins new item class to the current web page
addClientVars(vars)	Vars : client variables	If the page requires paths from the client side of an input this function handles adding them to the current generated page

Table 4.2: Helpers

There are some other functions but they are not critical to the implementation.

4.2 Controls

This files are responsible for preparing data to view are intermediate file between models and views which can talk to the IDE and send the data to the view to the user, this files are pure javascript files that handles a lot of functions that connects the views and controls together .

4.2.1 Javascript files

File	Function
easysync2.js	Initialization file that sets the first use and parameters of ethirpad component.
Server.js	Handles server start up and tracks users and the pad they are using , handles server side errors such as authentications.
Room_server.js	Handles server current working pads adding the pool feature to the cursors of the users.
Auth_control.js	Controls the authentications , logins, logouts and viewings
clone_control.js	Controls the pad clone and the users allowed to view this pad.
console_control.js	Controls the console part and it’s parameters in the IDE side.
editor_control.js	Controls the editor , errors , syntax highlighting.
static_control.js	Controls the static parts of the server like CSS , Images.
stats_control.js	Controls server state , up , suspended and can’t startup.
turk_control.js	Controls the turk mechanisms for controlling versions.
auth.js	Controls the editor authentications.
turk.js	Controls the editor versions.
workspace.js	Workspace controls.
activepads.js	Controls active pads.
dbwriter.js	Controls writing to database.
model.js	The model which holds the standard version of the editor and all basic functions and calls.
pad_events.js	Controls the event and detects the new events like : editing , deleting , modifying.
pad_users.js	Controls the actions of the users that are allowed to use

	this pad.
pad_utils.js	Basic functions clearing and copying the pads.
revisions.js	Sets the label and time stamps and manages versions of the content of the pads.

Table 4.3: javascript Files

4.3 **Models**

Primary the data which flows across the system, these files is responsible for retrieving the data from the IDE according to the request and response with the data that's transferred to the controls and then to the views where the user can see the data in a friendly format.

4.3.1 **Java files**

File	Function
Annotation.java	Controlling the IDE lineNumber,subtype,messages of the IDE.
Application.java	Controlling the initialization of the etherpadpad application and start of the server with a given configuration string that holds the settings of the application.
ChangeSetOp.java	Controlling the attribs of the code.
ChangeSetOpIterator.java	Controlling the change of the colors of the code or the string of code that have some functions that detects the end of the code.
Debug.java	Controlling the debugging of the code and creates the log file and manages the entries of the debugging queue
JavaPadDocument.java	Controlling the pad and listens to changes on the pad and also have some functions that provides information about the string written in the pad , notify the IDE to review the code written and check for errors.
JavaPadReconcileListener.java	Controlling the updates between the IDE and the pad.
PadDocument.java	Adding more functions and options to the pad to make it a document that will have the options of any IDE document , including permissions and authentications of this pad.
PadDocumentOwner.java	Holds the information about the owner of each document and some functions that allows to define his other pads and doucments.

Workspace.java	Controls the workspace of each user allowing him to edit projects and files , retrieving the files and projects and also editing his workspace adding other files like txt files.
RTC.java	Controlling collaboration between users who share changes, making sure that everyone works on a .java file only , Create a new document in the collaboration. Creates a {@link CollabDocument} for the given file, and a {@link PadDocument} for the given user, Notify listeners that edits were synchronized from pad to documents ,Notify listeners that edits were committed to disk.
RTCDoucment.java	A filed edited in a {@link Collab collaboration}. Maintains a collection of {@link PadDocument} working copies and the version on disk, Synchronize edits from pad to documents, Commit edits to disk, Convert a union offset to a local offset, Convert a local offset to a union offset, Convert local annotations to union annotations, Convert local edits to a union changeset, Convert local presentation to a union changeset, Convert union presentation to a union changeset, Represents the document on disk, Commit the contents of this document to the filesystem, Visitor that rebuilds text edits, mapping local offsets to union offsets.
RTCListener.java	Detects the updates and the committed changes to the disk.
CoordinateMap.java	Map between character offsets in local and union documents,Convert a union offset to a local offset,Convert a local offset to a union offset,Immutable character offset identity mapping,Mutable character offset mapping, Update the map to reflect an edit applied only to the union document,Update this map to reflect an edit applied to both union and local documents,Update this map to reflect an edit

	previously applied only to the union document, now applied to the local document,Enumerate regions appearing only in the union document,Enumerate regions appearing only in the local document.
Comitter.java	Commits error-free changes in Java files,
PadCompilationUnitEditor.java	Detects the units that is being edited and which user is editing them, marking the compilation of the unit and sending notification to connect the together
PadJavaSourceViewer.java	Presents the complete code of the pads after completing each unit.
PadSemanticHighlighter.java	Highlights the written java syntax
PadImportOrganizer.java	Returns the text edit that organizes imports, May block waiting for a call to {@link #chose} with user resolutions of ambiguous names, Report user resolutions of ambiguous class names.
FileRunOwner.java	Keeps track of launches, Run a file, for appropriate definition of run, Detects and run the main method of the primary class in a file, Returns the current launch state of a file, Terminate the launch associated with a file, if any.
ServerView.java	Controls server action and server startup

Table 4.4: java Files

Those are the three types of files that exists in this project which has been used to create this service

A simple SQL database has been used to store the users.

4.4 Core Functions

Function	Parameters	Task
Login(destination)	The path that the user want to be directed to after logging in.	Checks the login authentications.
syncUnionCoordinateEdits (PadDocument doc, int newRevision, ReplaceEdit[] edits)	Doc : is the pad that it's now edited. Int : revision number Edits : the edits and the changes of the pad	Synchronize edits from pad to documents.
FileRunOwner(String id)	Id : user id who owns the file	Prepares the launch settings and reports if there is an error or not
StartServerAction()		Starts the server
setupAndStart(String configFile)	configFile : string that start the application with certain configurations	Prepare the application setup and startup of the application
changeTextPresentation(TextPresentation presentation)	Presentation : the configuration of the syntax highlighting	Update syntax highlighting
Entry begin()		Starts thread with the new code and prints the stack trace
getLength()		Returns string length on the current pad.
isAllowed(ReplaceEdit[] edits, String[] permissions)	Edits : the changes of the current pad Permissions : permissions of this pad	Returns true if the edits are allowed.
documents()		Obtain all the documents owned by this owner.

Function	Parameters	Task
renderError(code)	Code : error code	Returns the template that should be rendered for this error
renderHtml(bodyFileName, data)	bodyFileName : is the body of the html generated page data : the data that needs to be viewed in the html page	Returns the rendered html with the given data
FileRunOwner(String id)	Id : user id who owns the file	Prepares the launch settings and reports if there is an error or not
StartServerAction()		Starts the server
onNewPad(pad)	Pad : is the pad that being currently created	Detects the creation of a new pad
setCurrentPad(p)	P: is the current pad	Sets the attributes of the current pad
clearCurrentPad()		Clear the text of the pad
acl(project)	Project : is the project that wants to retrieve it's permission	Returns the permissions of the current project
updatePadClients(pad)	Pad : is the current pad for the owner of the project	Updates all pads who have permission to see this pad.
sendMessage(connectionId, msg)	connectionId: is the id of the working collaborators in this document or pad msg : the message to be send to all collaborators in this project	Sends message to all working users in this project.

Table 4.5: Core Functions

4.5 **Data Flow**

The user views a ejs file that can be processed by a web browser that gets it's information from a javascript file , java script files sends requests to the java file that connects to the IDE with the given request and response with the data needed to be presented to the user , java file connects to different parts of the IDE such as the console and the markers that responds to errors , and debug core which holds the debug info and the log to the runs and the stack call ,ejs files represents all the data given by the java files to the users in html formats and a user friendly interface, already prepared by the javascript files.

4.6 **Unit Flow**

The user writes the code on to the pad the pad detects changes in the string within it, the pad invokes the listener to check for errors in the new code , if there is an error the IDE sends the error code then the javascript detects the error code sending the data to the correct template to render the page with the new error, if there is no error, then the user continues to write his code , if some user is sharing the pad then the listener checks if the user has completed a unit of the code then it detects all the users cursors and all pads are gathered in one pool detecting changes in each pad then updating every pad , allowing all collaborators to see the updates, if the owner runs the code then the editing stops and the code is sent to the IDE by debug core , then the IDE runs the code returning the results , which can be in the form of error or the output , then it's sent to the javascript to send to the right template of the EJS and sent to the user to see the results.

Chapter Five:

System Testing & Deployment

We faced some problems while we were developing our system, and we discovered and overcome these problems in testing and deployment phase testing. And this is how we test our project:

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). It tests the behavior and also the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specifications.

Testing the behavior of the whole software/system as defined in software requirements specification (SRS) is known as system testing, its main focus is to verify that the customer requirements are fulfilled.

System testing is done after integration testing is complete. System testing should test functional and non-functional requirements of the software.

- Software testing can be stated as the process of validating and verifying that a software program/application/product:

    1. Meets the requirements that guided its design and development.
    2. Works as expected.
    3. Can be implemented with the same characteristics.
    4. Satisfies the needs of stakeholders.

- Software testing, depending on the testing method employed, can be implemented at any time in the development process. Traditionally most of the test effort occurs after the requirements have been defined and the coding process has been completed, but in the agile approaches most of the test effort is on-going. As such, the methodology of the test is governed by the chosen software development methodology.

5.1 **Graphical user interface testing**

Graphical user interface testing is the process of testing a product's user interface to ensure it meets its written specifications. This is normally done through the use of a variety of test cases.

Test Case Generation

To generate a ‘good’ set of test cases, the test designers must be certain that their suite covers all the functionality of the system and also has to be sure that the suite fully exercises the GUI itself.

5.2 **Usability testing**

Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system. This is in contrast with usability inspection methods where experts use different methods to evaluate a user interface without involving users.

Usability testing focuses on measuring a human-made product's capacity to meet its intended purpose. Examples of products that commonly benefit from usability testing are foods, consumer products, web sites or web applications, computer interfaces, documents, and devices. Usability testing measures the usability, or ease of use, of a specific object or set of objects, whereas general human-computer interaction studies attempt to formulate universal principles.

Goals of usability testing:

Usability testing is a black-box testing technique. The aim is to observe people using the product to discover errors and areas of improvement.

Usability testing generally involves measuring how well test subjects respond in four areas: efficiency, accuracy, recall, and emotional response. The results of the first test can be treated as a baseline or control measurement; all subsequent tests can then be compared to the baseline to indicate improvement.

- **Performance:**
  How much time, and how many steps, are required for people to complete basic tasks?

- **Accuracy:**
  How many mistakes did people make? (And were they fatal or recoverable with the right information?)

- **Recall:**
  How much does the person remember afterwards or after periods of non-use?

- **Stickiness:**
  How much time he/she spends

- **Emotional response:**
  How does the person feel about the tasks completed? Is the person confident, stressed? Would the user recommend this system to a friend?

5.3 **<u>Software performance testing</u>**

In software engineering, performance testing is in general testing performed to determine how a system performs in terms of responsiveness and stability under a particular workload. It can also serve to investigate measure, validate or verify other quality attributes of the system, such as scalability, reliability and resource usage.

Performance testing:

- **Load testing**

  Load testing is the simplest form of performance testing. A load test is usually conducted to understand the behavior of the system under a specific expected load. This load can be the expected concurrent number of users on
  the application performing a specific number of transactions within the set duration. This test will give out the response times of all the important business critical transactions. If the database, application server, etc. are also monitored, then this simple test can itself point towards any bottlenecks in the application software.

- **Stress testing**

  Stress testing is normally used to understand the upper limits of capacity within the system. This kind of test is done to determine the system's robustness in terms of extreme load and helps application administrators to determine if the system will perform sufficiently if the current load goes well above the expected maximum.

- **Endurance testing (soak testing)**

  Endurance testing is usually done to determine if the system can sustain the continuous expected load. During endurance tests, memory utilization is monitored to detect potential leaks. Also important, but often overlooked is performance degradation. That is, to ensure that the throughput and/or response times after some long period of sustained activity are as good or better than at the beginning of the test. It essentially involves applying a significant load to a system for an extended, significant period of time. The goal is to discover how the system behaves under sustained use.

- **Spike testing**

  Spike testing is done by suddenly increasing the number of, or load generated by, users by a very large amount and observing the behavior of the system. The goal is to determine whether performance will suffer, the system will fail, or it will be able to handle dramatic changes in load.

- **Configuration testing**

  Rather than testing for performance from the perspective of load, tests are created to determine the effects of configuration changes to the system's components on the system's performance and behavior. A common example would be experimenting with different methods of load-balancing.

- **Isolation testing**

  Isolation testing is not unique to performance testing but a term used to describe repeating a test execution that resulted in a system problem. Often used to isolate and confirm the fault domain.

5.4 **Compatibility testing**

Compatibility testing, part of software non-functional tests, is testing conducted on the application to evaluate the application's compatibility with the computing environment. Computing environment may contain some or all of the below mentioned elements.

5.5 **Exception handling**

Exception handling is the process of responding to the occurrence, during computation, of exceptions – anomalous or exceptional situations requiring special processing – often changing the normal flow of program execution. It is provided by specialized programming language constructs or computer hardware mechanisms.

5.6 **Security testing**

Security testing is a process to determine that an information system protects data and maintains functionality as intended.

The six basic security concepts that need to be covered by security testing are:

1. Confidentiality
2. Integrity
3. Authentication
4. Availability
5. Authorization
6. Non-repudiation

5.6.1 **Confidentiality**

- A security measure which protects against the disclosure of information to parties other than the intended recipient that is by no means the only way of ensuring the security.

5.6.2 **Integrity**

- A measure intended to allow the receiver to determine that the information which it is providing is correct.

- Integrity schemes often use some of the same underlying technologies as confidentiality schemes, but they usually involve adding additional information to a communication to form the basis of an algorithmic check rather than the encoding all of the communication.

5.6.3 **Authentication**

This might involve confirming the identity of a person, tracing the origins of an artifact, ensuring that a product is what it’s packaging and labeling claims to be, or assuring that a computer program is a trusted one.

5.6.4 **Authorization**

- The process of determining that a requester is allowed to receive a service or perform an operation.

- Access control is an example of authorization…

5.6.5**Availability**

- Assuring information and communications services will be ready for use when expected.
- Information must be kept available to authorized persons when they need it.

5.6.6 **Non-repudiation**

- In reference to digital security, nonrepudiation means to ensure that a transferred message has been sent and received by the parties claiming to have sent and received the message. Nonrepudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

5.7 **Regression testing**

Regression testing is any type of software testing that seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes, such as enhancements, patches or configuration changes, have been made to them.

The intent of regression testing is to ensure that a change, such as a bug fix, did not introduce new faults. One of the main reasons for regression testing is to determine whether a change in one part of the software affects other parts of the software.

5.8 **Reliability testing**

Reliability theory describes the probability of a system completing its expected function during an interval of time.

5.9 **Maintenance testing**

Maintenance testing is that testing which is performed to either identify equipment problems, diagnose equipment problems or to confirm that repair measures have been effective. It can be performed at either the system level, the equipment level, or the component level.

Types of maintenance for which testing can be used

Maintenance falls into the following four categories:

- **Preventive maintenance:**
  Changes to the existing system so as to reduce the risk of failure while operating.

- **Corrective maintenance:**
  correcting problems that arise while using the system.

- **Perfective maintenance:**
  Enhancements (modifications) to improve the safety, reliability, efficiency or cost-effectiveness of operation.

- **Adaptive maintenance:**
  Adaptations to address requirements that crop up due to changes in the environment or new regulations.

5.10 **Recovery testing**

In software testing, recovery testing is the activity of testing how well an application is able to recover from crashes, hardware failures and other similar problems.

Recovery testing is the forced failure of the software in a variety of ways to verify that recovery is properly performed. Recovery testing should not be confused with reliability testing, which tries to discover the specific point at which failure occurs. Recovery testing is basically done in order to check how fast and better the application can recover against any type of crash or hardware failure etc. Type or extent of recovery is specified in the requirement specifications. It is basically testing how well a system recovers from crashes, hardware failures, or other catastrophic problems

5.11 **Accessibility testing**

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is available to as many people as possible. Accessibility can be viewed as the "ability to access" and benefit from some system or entity. Accessibility is often used to focus on people with disabilities or special needs and their right of access to entities, often through use of assistive technology.

Web accessibility

Web accessibility refers to the inclusive practice of making websites usable by people of all abilities and disabilities.

When sites are correctly designed, developed and edited, all users can have equal access to information and functionality. For example, when a site is coded with semantically meaningful HTML, with textual equivalents provided for images and with links named meaningfully, this helps blind users using text-to-speech software and/or text-to-Braille hardware.

When text and images are large and/or enlargeable, it is easier for users with poor sight to read and understand the content.

When links are underlined (or otherwise differentiated) as well as coloured, this ensures that color blind users will be able to notice them.

When clickable links and areas are large, this helps users who cannot control a mouse with precision. When pages are coded so that users can navigate by means of the keyboard alone, or a single switch access device alone, this helps users who cannot use a mouse or even a standard keyboard.

When videos are closed captioned or a sign language version is available, deaf and hard-of-hearing users can understand the video.

When flashing effects are avoided or made optional, users prone to seizures caused by these effects are not put at risk. And when content is written in plain language and illustrated with instructional diagrams and animations, users with dyslexia and learning difficulties are better able to understand the content.

When sites are correctly built and maintained, all of these users can be accommodated without decreasing the usability of the site for non-disabled users.

The needs that Web accessibility aims to address include:

- **Visual:**
  Visual impairments including blindness, various common types of low vision and poor eyesight, various types of color blindness;

- **Mobility:**
  difficulty or inability to use the hands, including tremors, muscle slowness, loss of fine muscle control, etc., due to conditions such as Parkinson's Disease, muscular dystrophy, cerebral palsy, stroke;

- **Auditory:**
  Deafness or hearing impairments, including individuals who are hard of hearing;

- **Seizures:**
  Photo epileptic seizures caused by visual strobe or flashing effects.

- **Cognitive/Intellectual:**
  Developmental disabilities, learning disabilities (dyslexia, dyscalculia), and cognitive disabilities of various origins, affecting memory, attention, developmental "maturity," problem-solving and logic skills, etc.

Chapter Six:

Conclusion & Future Work

6.1 **Conclusion**

Web Based-IDE is a cross-platform Integrated Development Environment (IDE), designed for the cloud. It enables you to easily develop, debug and deploy web applications using your browser. Our Project can be used instead or alongside your existing desktop IDE. You can upload existing code in order to test it in the cloud or for sharing with your peers. Much more testing is likely required to verify that this system can work using real world, more complex data. However, it has been demonstrated that the project with its current state can serve as the core of a Cloud IDE, system, and that such a system can deliver useful results. It has also been demonstrated that the system should be able to handle greater amounts of requested after some modifications.

As such, while there are many services needing improvement in the current implementation of this system, the basic design seems to be run multiprogramming languages. However, as mentioned, there is a need for more testing of the system. In particular, tests from a wider range of developer with different programming language.

Our System rely on the user interactive with the client and server, and improve itself with learn from its error and experiments.

6.2 **Future Work**

To Increase performance of Web Based-IDE, may be using a more Facilities for management all requests between Client and Server, to be able to observe the real contribution of this technology, the experiment should be done on a sufficiently large training.

Also, alternative filler model types can be examined.

We also plan to add more futures to server as allow client to take input from user and interactive with him, improve the user interface, and some modifications on programming language.

To improve the server managements for compile code with the best algorithms, you should rewrite the own code with assembler, compiler and your design architecture and using latest technology that handle web requests between client and server.