**Chapter 4: IMPLEMENTATION**

**4.1 Model**

**AGILE MODEL**

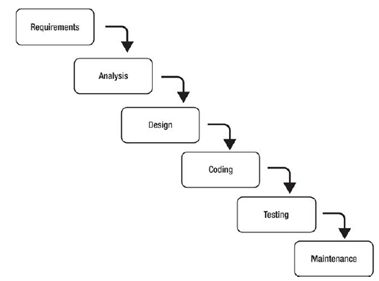
Agile is a software development methodology that is becoming more popular every day. It defines the mind set of many software development teams working across the globe.[2] This article will walk you through the entire Agile-scrum process and how, as a developer, you can contribute in an agile way and deliver value. BTW, Agile means ability to move quickly. [3]

**Agenda**

* Life Cycle of a Developer
* Barriers to value delivery
* Agile Adoption
* Plan Driven Vs. Value Driven
* Agile Manifesto
* Scrum End-to-End Process
* Agile Estimation
* Daily Standup
* Agile-Scrum Tools

**Life Cycle of a Developer**

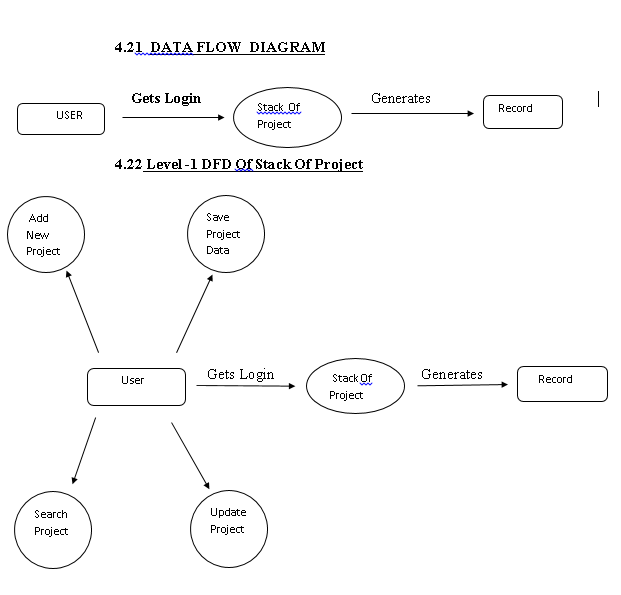
The 1st interaction a developer has is with a BA. They are responsible for documenting, tracking and describing the user requirements. But there are often gaps in understanding the requirements and that can cause issues in various ways. The following image shows what the user wanted and what happened when it was implemented. [4]



**FIGURE 4.1.1 Agile Software Development Model**

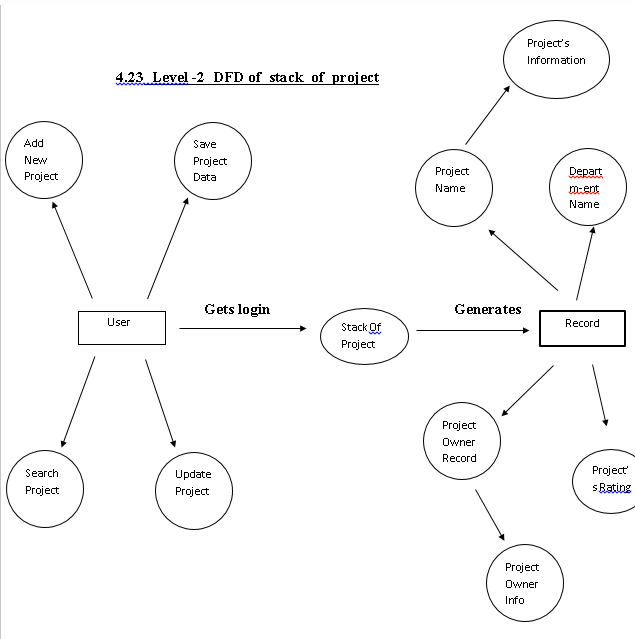
Predictive methods entirely depend on the requirement analysis and planning done in the beginning of cycle. Any changes to be incorporated go through a strict change control management and prioritization. Agile uses an adaptive approach where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future. [3] Customer Interaction is the backbone of this agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location. [7]

**4.2 Data Flow Diagram**



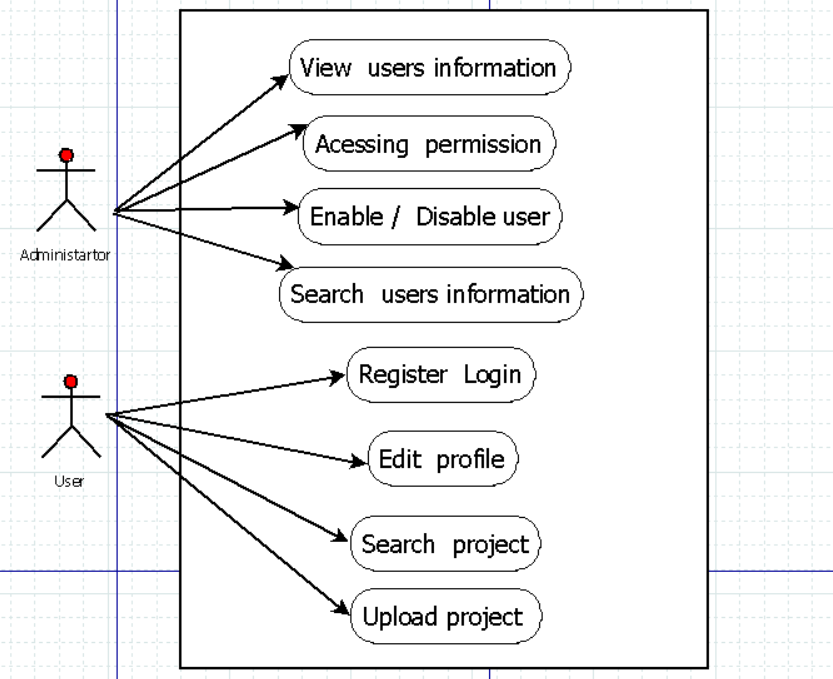
**FIGURE 4.2.1 FIRST LEVEL DFD**

This level one DFD explains about the feature when user registers on our website. Users can upload their own project when logged in. Users can search and add new projects on our website. When user adds new project admin approval is required.



**FIGURE 4.2.2 SECOND LEVEL DFD**

This level two DFD explains detailed procedure of how users enters their own project on our website. Users have to give their project basic information first to add their own project on our website. Users are asked for project name, its date of creation, little description about the project.



**FIGURE 4.2.3 USE CASE DIAGRAM**

Thisuse case diagram explains about the administrator functionalities and its benefits. It also gives information about the users who are registered on the website, When administrator accesses the content of website administrator has to login on the website. Administrator can enable and disable users and also projects hosted on the website. Users can search the projects and add their own projects on the website this use case explains about how users can do this.

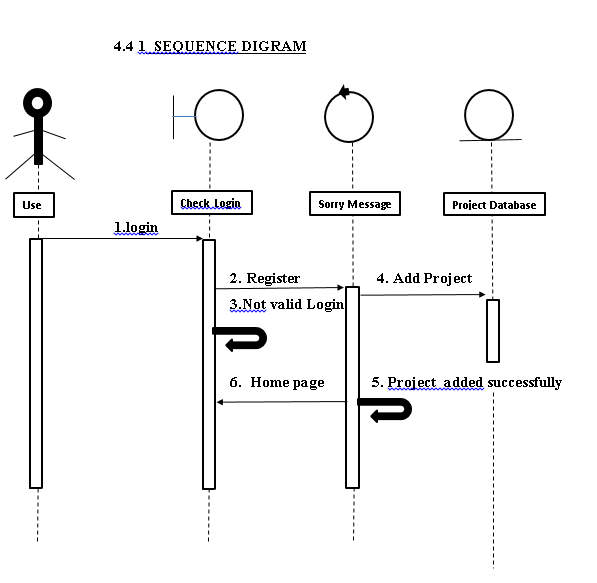


FIGURE 4.2.4 SEQUENCE DIAGRAM FOR LOGIN

This sequence diagram explains about how user logs in on our website. First users have to register, then they are asked for email id and password. When login is not valid then it sends a sorry message box on the interface and if it succeeds then it shows project interface.

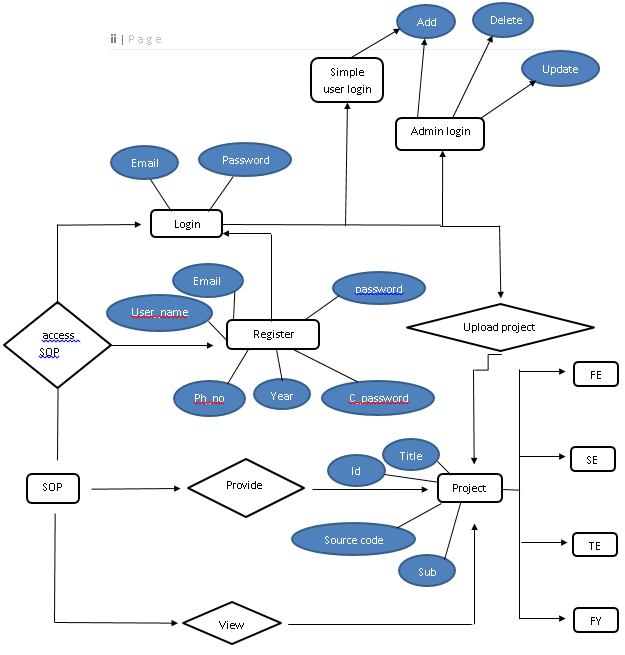
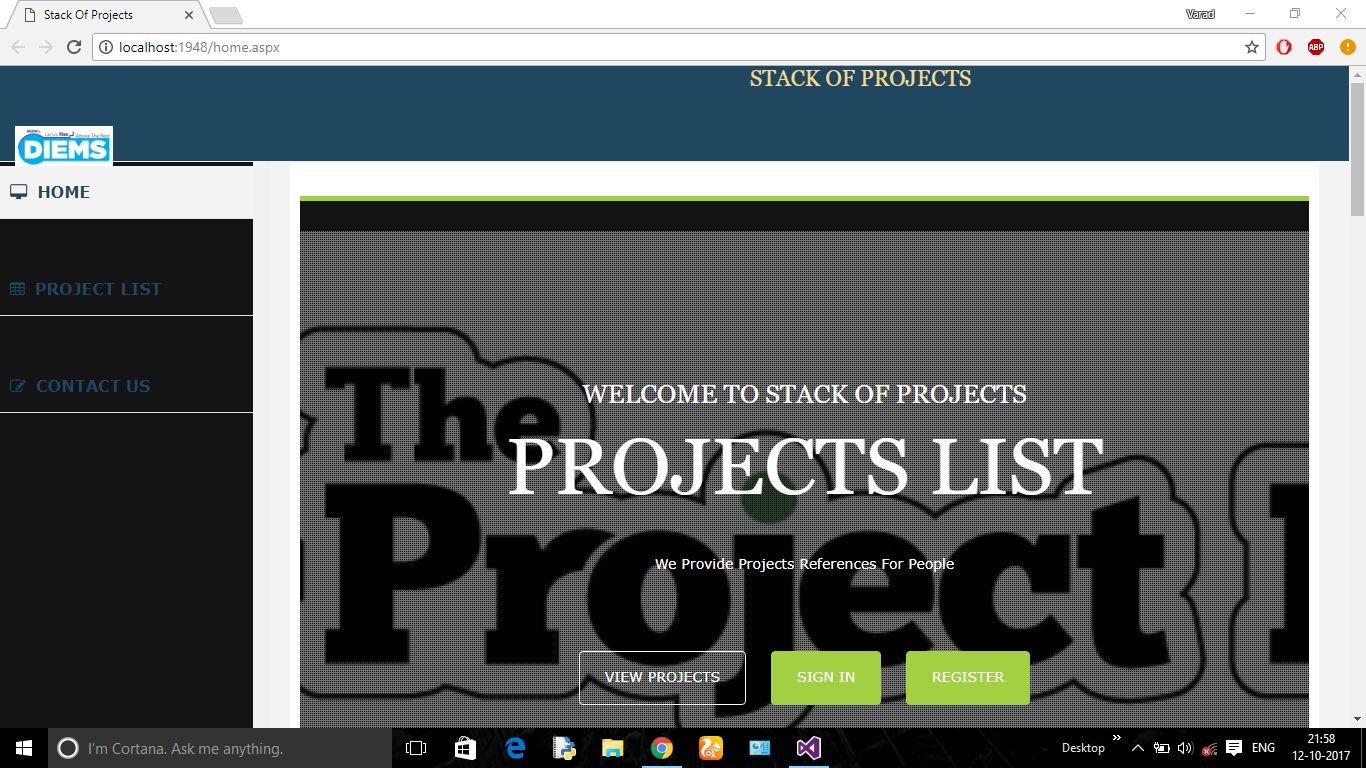
**4.3 ER DIAGRAM**

FIGURE 4.3.1 ER DIAGRAM FOR SOP ACCESS

This ER diagram explains about the how users and administrator login actually works on our website. When a person tries to log in our website, Administrator is asked for email and password. If it is admin login then administrator is provided with additional functionalities such as delete, update project and for normal user administrator is given only add new project functionality.

**4.4 USER INTERFACE DESIGN**

**1. HOME PAGE**

****

**Screen 4.4.1 Home page snap shot**

This above snapshot 4.4.1 explains about the source code of home page, were user can login, register or can contact us as shown in the above snapshot. For login session user has to fill the necessary details in validatory manner.

**SOURCE CODE**

Using System;

Using System.Collections.Generic;

Using System.Linq;

Using System. Web;

Using System.Web.UI;

Using System.Web.UI.WebControls;

Using System.Data.SqlClient;

public partial class SessionCreated\_AddProject : System.Web.UI.Page

{

SqlConnection con;

SqlCommand cmd;

SqlDataReader rdr;

protected void Page\_Load(object sender, EventArgs e)

{

Con=newSqlConnection(@"DataSource=(LocalDB)\v11.0;AttachDbFilename=C:\Users\VARAD\Documents\users.mdf;Integrated Security=True;Connect Timeout=30");

if (Session["username"] == null)

Response.Redirect("~/login1.aspx");

}

}

protected void Button1\_Click(object sender, EventArgs e)

{

String name = TextBox1.Text;

String tech= TextBox2.Text;

String owner = TextBox3.Text;

String email = TextBox4.Text;

String date = TextBox5.Text;

String desc = TextBox6.Text;

String refe = TextBox7.Text;

String year = TextBox8.Text;

String clgname = TextBox9.Text;

String address = TextBox10.Text;

String que = "insert into PendingProjects values('" + name + "','" + tech + "','" + owner + "','" + . email + "','" + date + refe + "','" + year + "','" + clgname + "','" + address + "')";

cmd = new SqlCommand(que,con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

// ScriptManager.RegisterStartupScript (this, this.GetType ()),"alert","alert ('User details saved successfully'); window.location='~/SessionCreated/projects1.aspx';", true);

//Response.Redirect("~/SessionCreated/projects1.aspx");

Response.Write ("<script language='javascript'>window.alert ('Your Project Was Added!'); window.location='projects1.aspx';< /script>");

}

protected void Button2\_Click (object sender, EventArgs e)

{//update

String name = TextBox1.Text;

String tech = TextBox2.Text;

String owner = TextBox3.Text;

String email = TextBox4.Text;

String date = TextBox5.Text;

String desc = TextBox6.Text;

String refe = TextBox7.Text;

String year = TextBox8.Text;

String clgname = TextBox9.Text;

String address = TextBox10.Text;

String que = "Update PendingProjects set projecttech='" + tech + "',projectowner='" + owner + "',projectowneremail='" + email + "',projectdate='" + date + "',projectdesc='" + desc + "',projectref='" + refe + "',currentyear='" + year + "',collegename='" + clgname + "',address='" + address + "' where projectname='" + name + "'";

cmd = new SqlCommand(que, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Response.Write ("<script language='javascript'>window.alert ('Your Project Was Updated!'); window.location='projects1.aspx';< /script>");

}

protected void Button3\_Click (object sender, EventArgs e)

{//delete

String name = TextBox1.Text;

String que = "Delete from PendingProjects where projectname='" + name + "'";

cmd = new SqlCommand(que, con);

con.Open();

cmd.ExecuteNonQuery();

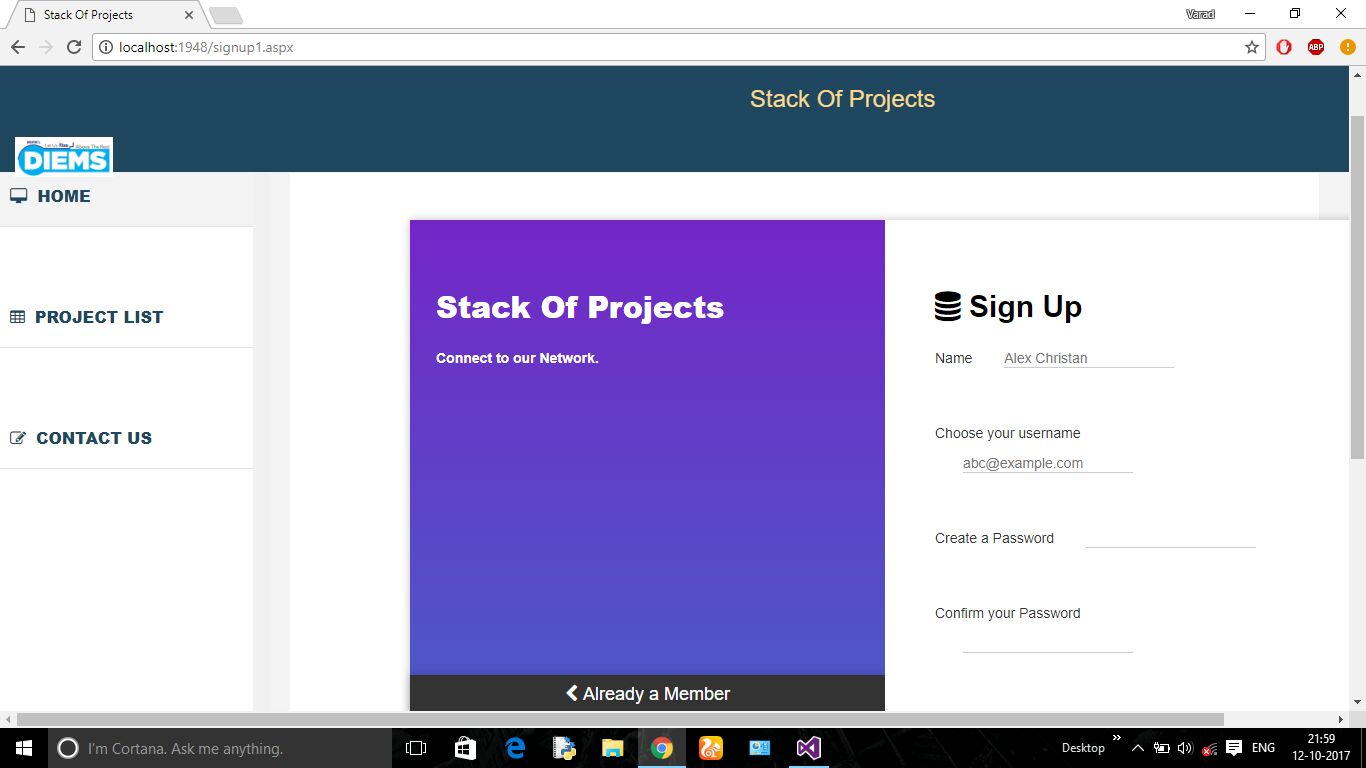
con.Close();

Response.Write ("<script language='javascript'>window.alert ('Your Project Was Deleted!'); window.location='projects1.aspx';< /script>");

}

}

**2. SIGN UP SESSION**

****

**Screen 4.4.2 Snap shot for signup session**

This snapshot explains about the functionality of how admin login is performed. When user id and password are matched with that of administrator credentials then administrator is directed on the respective page.

**SOURCE CODE**

using System;

using System.Collections.Generic;

using System.Linq;

using System. Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

public partial class signup1 : System.Web.UI.Page

{

SqlConnection con;

SqlCommand cmd;

SqlDataReader rdr;

protected void Page\_Load(object sender, EventArgs e)

{

con=newSqlConnection(@"DataSource=(LocalDB)\v11.0;AttachDbFilename=C:\Users\VARAD\Documents\users.mdf;Integrated Security=True;Connect Timeout=30");

}

protected void submit\_Click(object sender, EventArgs e)

{

String name1 = namebox.Text;

String username1 = usernamebox.Text;

String pass1 = passbox.Text;

String cpass1 = cpassbox.Text;

String phone1 = phonebox.Text;

String que = "Insert into userdata values('" + name1 + "','" + username1 + "','" + pass1 + "','" + cpass1 + "','" + phone1 + "')";

cmd = new SqlCommand(que, con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

// Response.Write("Registration Successful");

// Response.Redirect("~/login1.aspx");

Response.Write("<script language='javascript'>window.alert('You are Registered Successfully!');window.location='login1.aspx';</script>");

}

}