INDEX

ABSTRACT

The main aim of our project is to prepare a Tax summary or Tax Returns of a client. In Tax Information system System, a client registers himself enters all the details and uploads various Documents that are necessary for preparation of Tax Summary and Schedules for an interview after successful submission of all the documents. After all the procedures are completed Tax Returns or Tax summary is prepared for all the clients by the admin who calls the clients and arranges an interview for discussing various issues regarding Tax summary .

Once the client pays the amount for preparation of Tax Summary he can download the PDF format of his Tax Summary.

This project is composed of two main modules which also includes many sub modules.

1. Client module
2. Admin module

Client Panel :

- Register
- Create client profile
- Upload documents
- Schedule interview
- Tax Summary
- Payment
- PDF/Review/Correction
- Authorization
- E-File Acknowledgement
- Refund Status

Admin Panel :

- Call the clients
- Collect Documents
- Schedule interviews
- Preview Client Profile and Documents
- Interview
- Tax Preparation
- Send Tax summary
- Tax Summary Queries
- Review Tax Summary
- Payment follow up
- Payment Receipt
- Discounts
- PDF upload /Correction
- Receiving E Files
- Manual filling
- Send E File
- Refund Status

Client module includes registration of clients, submission of various documents etc., Admin module includes the calling the clients ,Scheduling interviews, Observation of various documents and preparation of Tax Summary and sent a mail to the client upon successful payment of fee.

The project has been planned to be having the view of distributed architecture ,with centralized storage of the database. The application for the storage of the data has been planned .Using the construct of
MS-SQL Server 2005 and all the users interfaces have been designed using the ASP.Net technologies .The database connectivity is planned using the ‘SQL Connection’ methodology. The statement of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are their associated reported ,which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

Company Profile

BLESSO is a proactive organization located in Hyderabad,
Andhra Pradesh .India .Specializing in customized software training & development corporate training ,and consultancy service.

The business paradigm of Blesso , includes the following areas

Customized software development – Blesso ,develops customized software solutions for its clients and also for internal use. The library of on-the-self software's include Payroll Software, Educational Institute Management Software. Library Management Software etc..

Corporate Training and IT Education –Blesso , has also launched various initiatives in the area of corporate training and IT education. This has brought the company as a benchmark for Training solutions by some of the biggest names of the global corporate world and Govt. Institute, in Hyderabad and Southern India.

Mission

To be a proactive leader of the global information technology fraternity.

Vision

To develop a strong client base with an equally effective support structure which acts as a catalyst for effective deployment of futuristically complete and credible IT solutions. We strive to achieve this by focusing individually on each project and build a healthy relationship with our customers.

Quality

Quality is a comprehensive and fundamental rule or believe, for leading and operating an organization. And this helps in continually improving performance over the ling term.

Management Term

After having brought the company from the conceptualization stage onward, the management is confident that technologies, work force and determination are poised for growth and wide acceptance. The management is continually identifying some more niche segments, is desirable to ensure globalization of the organization's where the presence is desirable to ensure globalization of the organization's presence.

Corporate training

From more than past 4 years corporate training has been our domain. We provide quality computer education for all type of computer related courses, at the client's as well as our own premises. We manage whole if the training process that begins with the selection of the place of training ,and continues till course delivery and final certificate (and this includes industry level certification like .Net Certification.)

Technical Strength

So far have 40 employee working on various projects and also handles corporate batches.

Clientele

1. India Law Group
2. KCP Projects Ltd
3. Prasad Association
4. Orange Model School
5. ZSS technologies
6. Sri Sai Granites
7. B.S Enterprises

Technical Expertise:

Operating Systems	: Windows 2000 family products.LINUS
Frameworks	: .Net Framework ,.Net, Testing Tools ,J2SE,J2EE(EJB.JSM,JNDI, JTS,JAAS,JAXP,JAVA MAIL), Remoting
Database	Oracle 8i/9i, MySQL 4.0/5.0.SQL Sever
Markup language	: XML, HTML, WML
Security Testing	: Manual and Automation
Testing Tools	: QTP, Winrunner , Test Director, Load runner, Rational Robot

BLESSO SOFTWARE INDIA Pvt,Ltd

//#302,Balaji Towers, Beside Mythri Hospital
Gayatri Nagar., Ameerpet,
Hyderabad-500038
Ph: 91-040-55661333.

E_Mail: enquiries@visualmindindinfo.net

www.visualmindinfo.net

REQUIREMENT ANALYSIS

3.1 INTRODUCTION

A complete understanding of software requirement is essential to the success of software development effort. No matter how well design/well coded, a poorly analyzed and specified program will disappoint the user and bring the grief to the user.

The requirement analysis task is a process of discovery, refinement, modeling and specification. The software initially established by the system engineer and refined during software project, planning is refined in detail. Models of the required information and control flow, operational and data content are created. Alternative solutions are analyzed and allocation to various software elements.

Requirements analysis, software-engineering task that bridge the gap between system level software allocation and software design .Requirements analysis enables to specify software function and performance indicates software's interface with other system elements and established design constraints that the software must meet. Requirement analysis allows the software engineers to refine.

The software allocation and build models of processes, date and behavioural domains that will be treated by software designer with a representation of information and function that can be translated to data, architectural and procedural design. Finally, the requirement specification the developer and customer with the means to access quality once the software is built.

3.1.1 PURPOSE OF THE SYSTEM:

The main aim of our project is to prepare a Tax summary or Tax Returns of a client. In Tax Information system System, a client registers himself enters all the details and uploads various Documents that are necessary for preparation of Tax Summary and Schedules for an interview after successful submission of all the documents. After all the procedures are completed Tax Returns or Tax summary is prepared for all the clients by the admin who calls the clients and arranges an interview for discussing various issues regarding Tax summary .Once the client pays the amount for preparation of Tax Summary he can download the PDF format of his Tax Summary.

3.1.2 SCOPE OF THE SYSTEM:

Our system mainly focus on the Tax that need to be paid by the people who are migrated from India to foreign countries. The scope of the system is restricted to Indians. Our system is capable of including future advancements.

3.1.3 ACRONYMS AND ABBREVIATIONS:

The various acronyms and abbreviations used in our project are

TIS	: Tax information system
SSN	: Social Security number

3.2 CURRENT SYSTEM:

Tax Information System is a professional financial services company established by experienced CPA's and chartered accounts to provide cost effective and efficient financial and tax planning solutions to a wide range of clients across the globe.

It is too expensive to overlook the deductions that you are entitled to ?
Let out professionals plan and prepare your tax returns to make sure you get the maximum benefit you deserve .

Tax information system is providing single window tax service to the Indian software consultants having income from both India and USA by minimizing the overall tax burden claiming foreign tax credits and other tax planning strategies according to the US internal revenue code and India income tax Act,1961

3.3 PROPOSED SYSTEM :

3.3.1 OVERVIEW

Why the new system?

With the new system the following activities get more moment.

1. Login/Register

- Returning clients should login to our website to update then profiles.
- New clients should register at our website to create their profiles.

2. Documents

- Upload / Fax your W2s, 1099 s or any other statement relating the interest, divide or stocks.

3. Tax Interview

- Schedule your convenient date time for brief Tax interview.
- Keep all the tax relevant information ready for the Tax Interview.

4. Tax Summary

- Tax Summary will be ready within 24 – 48 hrs after Tax Interview.
- Login and check the Tax Summary.

5. **Make Payment**

- Pay the Tax Preparation Fees at our website using pay pal Account or credit card

6. **Tax Returns Download**

- Download Tax Returns in PDF on payment of Tax Preparation Fees.
- Review Tax Returns carefully to ensure that there are no omissions if misstatements.
- Send us an Email or call us for any questions or corrections.
- Sign the e- files authorization form and fax it.

7. **E-File**

- We E-File your Tax Returns at no extra cost on receipt of authorization.
- Finally, check your filling and refund status at our website

3.3.2 FUNCTIONAL REQUIREMENTTS

The system deals with Tax Information System

Module description:

This project is composed of two main modules which also includes Many sub modules

1. Client module
2. Admin module

Client Panel :

- Register
- Create client profile
- Upload documents

- Schedule interview
- Tax Summary
- Payment
- PDF/Review/Correction
- Authorization
- E-File Acknowledgement
- Refund Status

Admin Panel :

- Call the clients
- Collect Documents
- Schedule interviews
- Preview Client Profile and Documents
- Interview
- Tax Preparation
- Send Tax summary
- Tax Summary Queries
- Review Tax Summary
- Payment follow up
- Payment Receipt
- Discounts
- PDF upload /Correction
- Receiving E Files
- Manual filling
- Send E File
- Refund Status

Client module includes registration of clients, submission of various documents etc., Admin module includes the calling the clients ,Scheduling interviews, Observation of various documents and preparation of Tax Summary and sent a mail to the client upon successful payment of fee.

The project has been planned to be having the view of distributed architecture ,with centralized storage of the database. The application for the storage of the data has been planned .Using the construct of
MS-SQL Server 2005 and all the users interfaces have been designed using the ASP.Net technologies .The database connectivity is planned using the ‘SQL Connection’ methodology. The statement of security and data protective

mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, which are their associated reported ,which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff

3.3.3 NONFUNCTIONAL REQUIREMENTS

3.3.3.1 User Interface and Human Factor :

The system provides user-friendly interface with both Keyboard and mouse, the forms are

- Client Login form
- Register form
- Myprofile form
- Spouse form
- Dependent form
- Bank information form
- Vehicle information form
- Documents form
- Scheduled interview form
- Change password form
- Tax Summary form
- Payments form
- Tax return form
- Admin login form
- Register form
- Assign Interviewer

3.3.3.2 DOCUMENTATION

- Requirement Analysis Document (RAD)
- System Design Document

- Coding
- Testing

3.3.3.3 HARDWARE CONSIDERATIONS

- INTEL CELEON/P2/P4 : 600 MHZ OR ABOVE
- RAM(SD/DDR) : 256 OR ABOVE
- HARD DISC : 10GB OR ABOVE
- PRINTER : HP LASER JET
- INTERFACE : MOUSE, KEYBOARD

3.3.3.4 ERROR HANDLING AND EXTREAM CONDITIONS:

The users of the system either ADMIN or CLIENT should enter all the fields in the relevant forms. The entered fields should not violate the primary key constraints. If the user either wants to update or delete a record, which is not in database, it will display a massage.

3.3.3.5 SYSTEM MODIGICATIONS

Future enhancements are possible, our system can include any number of extensions that the Client want to include.

3.3.4 PSEUDO REQUIREMENT

3.3.4.1 Hardware requirement

- 486 Processor (a Pentium based computer)
- 32MB of RAM
- About 150MB of free Hard-Drive space
- Windows NT Server 4.0, Windows NT Workstation 4.0 or Windows98/Windows95 with TCP/IP Networking support properly installed and configured.
- A web server that supports ASP 2.0
- A Database that supports ODBC (Such as Microsoft Access or Microsoft SQL Server).
- Microsoft Visual interDev 6.0

3.3.4.2 Software specifications

- windows 2000 server/professional ofr above
- .Net Framework 2.0
- Asp.Net, C# .Net
- SQL Server 2005
- IIS 5.0 or Above

MICROSOFT .NET FRAMEWOR

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives.

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally. Executed locally but internet – distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of application, such as Windows-based application and Web-based application.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components; the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting,
while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental

principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.
The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing.

Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance. Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

Common Type System

The common type system defines how types are declared, used, and managed in the runtime, and is also an important part of the runtime's support for cross-language integration. The common type system performs the following functions:

Establishes a framework that enables cross-language integration, type safety, and high performance code execution.
Provides an object-oriented model that supports the complete implementation of many programming languages.
Defines rules that languages must follow, which helps ensure that objects written in different languages can interact with each other.
In This Section Common Type System Overview
Describes concepts and defines terms relating to the common type system.

Type Definitions

Describes user-defined types.

Type Members

Describes events, fields, nested types, methods, and properties, and concepts such as member overloading, overriding, and inheritance.

Value Types

Describes built-in and user-defined value types.

Classes

Describes the characteristics of common language runtime classes.

Delegates

Describes the delegate object, which is the managed alternative to unmanaged function pointers.

Arrays

Describes common language runtime array types.

Interfaces

Describes characteristics of interfaces and the restrictions on interfaces imposed by the common language runtime.

Pointers

Describes managed pointers, unmanaged pointers, and unmanaged function pointers.

Related Sections

. NET Framework Class Library

Provides a reference to the classes, interfaces, and value types included in the Microsoft .NET Framework SDK.

Common Language Runtime

Describes the run-time environment that manages the execution of code and provides application development services.

Cross-Language Interoperability

The common language runtime provides built-in support for language interoperability. However, this support does not guarantee that developers using another programming language can use code you write. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

This section describes the common language runtime's built-in support for language interoperability and explains the role that the CLS plays in enabling guaranteed cross-language interoperability. CLS features and rules are identified and CLS compliance is discussed.

In This Section

Language Interoperability

Describes built-in support for cross-language interoperability and introduces the Common Language Specification.

What is the Common Language Specification?

Explains the need for a set of features common to all languages and identifies CLS rules and features.

Writing CLS-Compliant Code

Discusses the meaning of CLS compliance for components and identifies levels of CLS compliance for tools.

Common Type System

Describes how types are declared, used, and managed by the common language runtime.

Metadata and Self-Describing Components

Explains the common language runtime's mechanism for describing a type and storing that information with the type itself.

. NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

- Console applications.
- Scripted or hosted applications.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.
- XML Web services.
- Windows services.

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

Client Application Development

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI

elements, and they likely access local resources such as the file system and peripherals such as printers.
Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

In the past, developers created such applications using C/C++ in conjunction with the Microsoft Foundation Classes (MFC) or with a rapid application development (RAD) environment such as Microsoft® Visual Basic®. The .NET Framework incorporates aspects of these existing products into a single, consistent development environment that drastically simplifies the development of client applications.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

Unlike ActiveX controls, Windows Forms controls have semi-trusted access to a user's computer. This means that binary or natively executing code can access some of the resources on the user's system (such as GUI elements and limited file access) without being able to access or compromise other resources. Because of code access security, many applications that once needed to be installed on a user's system can now be safely deployed through the Web. Your applications can implement the features of a local application while being deployed like a Web page.

Managed Execution Process

The managed execution process includes the following steps:

Choosing a Complier

To obtain the benefits provided by the common language runtime, you must use one or more language compilers that target the runtime.

Compiling your code to Microsoft Intermediate Language (MSIL)

Compiling translates your source code into MSIL and generates the required metadata.

Compiling MSIL to native code

At execution time, a just-in-time (JIT) compiler translates the MSIL into native code. During this compilation, code must pass a verification process that examines the MSIL and metadata to find out whether the code can be determined to be type safe.

Executing your code

The common language runtime provides the infrastructure that enables execution to take place as well as a variety of services that can be used during execution.

Assemblies Overview

Assemblies are a fundamental part of programming with the .NET Framework. An assembly performs the following functions:

It contains code that the common language runtime executes. Microsoft intermediate language (MSIL) code in a portable executable (PE) file will not be executed if it does not have an associated assembly manifest. Note that each assembly can have only one entry point (that is, DllMain, WinMain, or Main).

It forms a security boundary. An assembly is the unit at which permissions are requested and granted. For more information about security boundaries as they apply to assemblies, see Assembly Security Considerations

It forms a type boundary. Every type's identity includes the name of the assembly in which it resides. A type called MyType loaded in the scope of one assembly is not the same as a type called MyType loaded in the scope of another assembly.

It forms a reference scope boundary. The assembly's manifest contains assembly metadata that is used for resolving types and satisfying resource requests. It specifies the types and resources that are exposed outside the assembly. The manifest also enumerates other assemblies on which it depends.

It forms a version boundary. The assembly is the smallest versionable unit in the common language runtime; all types and resources in the same assembly are versioned as a unit. The assembly's manifest describes the version dependencies you specify for any dependent assemblies. For more information about versioning, see Assembly Versioning

It forms a deployment unit. When an application starts, only the assemblies that the application initially calls must be present. Other assemblies, such as localization resources or assemblies containing utility classes, can be retrieved on demand. This allows applications to be kept simple and thin when first downloaded. For more information about deploying assemblies, see Deploying Applications

It is the unit at which side-by-side execution is supported. For more information about running multiple versions of the same assembly, see Side-by-Side Execution

Assemblies can be static or dynamic. Static assemblies can include .NET Framework types (interfaces and classes), as well as resources for the assembly (bitmaps, JPEG files, resource files, and so on). Static assemblies are stored on disk in PE files. You can also use the .NET Framework to create dynamic assemblies, which are run directly from memory and are not saved to disk before execution. You can save dynamic assemblies to disk after they have executed.

There are several ways to create assemblies. You can use development tools, such as Visual Studio .NET, that you have used in the past to create .dll or .exe files. You can use tools provided in the .NET Framework SDK to create assemblies with modules created in other development environments. You can also use common language runtime APIs, such as Reflection. Emit, to create dynamic assemblies.

ASP.NET

Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

Server-side managed code

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always

scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.
The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

For example, the Web Services Description Language tool included with the .NET Framework SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that your application can use to become a client of the XML Web service. The source code can create classes derived from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although you can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate your development efforts with the .NET Framework.

If you develop and publish your own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without concerning yourself with the communications infrastructure required by distributed software development.

Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

Programming with the .NET Framework

This section describes the programming essentials you need to build .NET applications, from creating assemblies from your code to securing your application. Many of the fundamentals covered in this section are used to create any application using the .NET Framework. This section provides conceptual information about key programming concepts, as well as code samples and detailed explanations.

Accessing Data with ADO.NET

Describes the ADO.NET architecture and how to use the ADO.NET classes to manage application data and interact with data sources including Microsoft SQL Server, OLE DB data sources, and XML.
Accessing Objects in Other Application Domains using .NET Remoting
Describes the various communications methods available in the .NET Framework for remote communications.

Accessing the Internet

Shows how to use Internet access classes to implement both Web- and Internet-based applications.

Creating Active Directory Components

Discusses using the Active Directory Services Interfaces.

Creating Scheduled Server Tasks

Discusses how to create events that are raised on reoccurring intervals.

Developing Components

Provides an overview of component programming and explains how those concepts work with the .NET Framework.

Developing World-Ready Applications

Explains the extensive support the .NET Framework provides for developing international applications.

Discovering Type Information at Runtime

Explains how to get access to type information at run time by using reflection.

Drawing and Editing Images
Discusses using GDI+ with the .NET Framework.
Emitting Dynamic Assemblies
Describes the set of managed types in the System.Reflection.Emit namespace.

Employing XML in the .NET Framework

Provides an overview to a comprehensive and integrated set of classes that work with XML documents and data in the .NET Framework.

Extending Metadata Using Attributes

Describes how you can use attributes to customize metadata.
Generating and Compiling Source Code Dynamically in Multiple Languages
Explains the .NET Framework SDK mechanism called the Code Document Object Model (CodeDOM) that enables the output of source code in multiple programming languages.

Grouping Data in Collections

Discusses the various collection types available in the .NET Framework, including stacks, queues, lists, arrays, and structs.

Handling and Raising Events

Provides an overview of the event model in the .NET Framework.

Handling and Throwing Exceptions

Describes error handling provided by the .NET Framework and the fundamentals of handling exceptions.

Microsoft SQL Server 2005 Storage Engine

Introduction

SQL Server™ 2005 a scalable, reliable, and easy-to-use product that will provide a solid foundation for application design for the next 20 years.

Storage Engine Design Goals

Database applications can now be deployed widely due to intelligent, automated storage engine operations. Sophisticated yet simplified architecture improves performance, reliability, and scalability.

Feature	**Description and Benefits**
Reliability	Concurrency, scalability, and reliability are improved with simplified data structures and algorithms. Run-time checks of critical data structures make the database much more robust, minimizing the need for consistency checks.
Scalability	The new disk format and storage subsystem provide storage that is scalable from very small to very large databases. Specific changes include: Simplified mapping of database objects to files eases management and enables tuning flexibility. DB objects can be mapped to specific disks for load balancing. More efficient space management including increasing page size from 2 KB to 8 KB, 64 KB I/O, variable length character fields up to 8 KB, and the ability to delete columns from existing tables without an unload/reload of the data. Redesigned utilities support terabyte-sized databases efficiently.
Ease of Use	DBA intervention is eliminated for standard operations—enabling branch office automation and desktop and mobile database applications. Many complex server operations are automated.

Storage Engine Features

Feature	**Description and Benefits**
Data Type Sizes	Maximum size of character and binary data types is dramatically increased.
Databases and Files	Databases creation is simplified, now residing on operating system files instead of logical devices.
Dynamic Memory	Improves performance by optimizing memory allocation and usage. Simplified design minimizes contention with other resource managers.
Dynamic Row-Level Locking	Full row-level locking is implemented for both data rows and index entries. Dynamic locking automatically chooses the optimal level of lock (row, page, multiple page, table) for all database operations. This feature provides improved concurrency with no tuning. The database also supports the use of "hints" to force a particular level of locking.
Dynamic Space Management	A database can automatically grow and shrink within configurable limits, minimizing the need for DBA intervention. It is no longer necessary to pre allocate space and manage data structures.
Evolution	The new architecture is designed for extensibility, with a foundation for object-relational features.
Large Memory Support	SQL Server 7.0 Enterprise Edition will support memory addressing greater than 4 GB, in conjunction with Windows NT Server 5.0, Alpha processor-based systems, and other techniques.
Unicode	Native Unicode, with ODBC and OLE DB Unicode APIs, improves multilingual support.

Storage Engine Architectural Overview

Overview

The original code was inherited from Sybase and designed for eight-megabyte Unix systems in 1983.These new formats improve manageability and scalability and allow the server to easily scale from low-end to high-end systems, improving performance and manageability.

Benefits

There are many benefits of the new on-disk layout, including:

- **Improved scalability and integration with Windows NT Server**
- **Better performance with larger I/Os**
- **Stable record locators allow more indexes**
- **More indexes speed decision support queries**
- **Simpler data structures provide better quality**
- **Greater extensibility, so that subsequent releases will have a cleaner development process and new features are faster to implement**

Storage Engine Subsystems

Most relational database products are divided into relational engine and storage engine components. This document focuses on the storage engine, which has a variety of subsystems:

- **Mechanisms that store data in files and find pages, files, and extents.**
- **Record management for accessing the records on pages.**
- **Access methods using b-trees that are used to quickly find records using record identifiers.**
- **Concurrency control for locking,used to implement the physical lock manager and locking protocols for page- or record-level locking.**
- **I/O buffer management.**
- **Logging and recovery.**
- **Utilities for backup and restore, consistency checking, and bulk data loading.**

Databases, Files, and File groups

Overview

SQL Server 2005 is much more integrated with Windows NT Server than any of its predecessors. Databases are now stored directly in Windows NT Server files .SQL Server is being stretched towards both the high and low end.

Files

SQL Server 7.0 creates a database using a set of operating system files, with a separate file used for each database. Multiple databases can no longer share the same file. There are several important benefits to this simplification. Files can now grow and shrink, and space management is greatly simplified. All data and objects in the database, such as tables, stored procedures, triggers, and views, are stored only within these operating system files:

File Type	Description
Primary data file	This file is the starting point of the database. Every database has only one primary data file and all system tables are always stored in the primary data file.
Secondary data files	These files are optional and can hold all data and objects that are not on the primary data file. Some databases may not have any secondary data files, while others have multiple secondary data files.
Log files	These files hold all of the transaction log information used to recover the database. Every database has at least one log file.

When a database is created, all the files that comprise the database are zeroed out (filled with zeros) to overwrite any existing data left on the disk by previously deleted files. This improves the performance of day-to-day operations.

File groups

A database now consists of one or more data files and one or more log files. The data files can be grouped together into user-defined filegroups. Tables and indexes can then be mapped to different filegroups to control data placement on physical disks. Filegroups are a convenient unit of administration, greatly improving flexibility. SQL Server 7.0 will allow you to back up a different portion of the database each night on a rotating schedule by choosing which filegroups to back up. Filegroups work well for sophisticated users who know where they want to place indexes and tables. SQL Server 7.0 can work quite effectively without filegroups.

Log files are never a part of a file group. Log space is managed separately from data space.

Using Files and File groups

Using files and file groups improves database performance by allowing a database to be created across multiple disks, multiple disk controllers, or redundant array of inexpensive disks (RAID) systems. For example, if your computer has four disks, you can create a database that comprises three data files and one log file, with one file on each disk. As data is accessed, four read/write heads can simultaneously access the data in parallel, which speeds up database operations. Additionally, files and file groups allow better data placement because a table can be created in a specific file group. This improves performance because all I/O for a specific table can be directed at a specific disk. For example, a heavily used table can be placed on one file in one file group and located on one disk. The other less heavily accessed tables in the database can be placed on other files in another file group, located on a second disk.

Space Management

There are many improvements in the allocations of space and the management of space within files. The data structures that keep track of page-to-object relationships were redesigned. Instead of linked lists of pages, bitmaps are used because they are cleaner and simpler and facilitate parallel scans. Now each file is more autonomous; it has more data about itself, within itself. This works well for copying or mailing database files.

SQL Server now has a much more efficient system for tracking table space. The changes enable

- Growing and shrinking files
- Better support for large I/O
- Row space management within a table
- Less expensive extent allocations

SQL Server is very effective at quickly allocating pages to objects and reusing space freed by deleted rows. These operations are internal to the system and use data structures not visible to users, yet are occasionally referenced in SQL Server messages.

File Shrink

The server checks the space usage in each database periodically. If a database is found to have a lot of empty space, the size of the files in the database will be reduced. Both data and log files can be shrunk. This activity occurs in the background and does not affect any user activity within the database. You can also use the SQL Server Enterprise Manager or DBCC to shrink files as individually or as a group, or use the DBCC commands SHRINKDATABASE or SHRINKFILE.

SQL Server shrinks files by moving rows from pages at the end of the file to pages allocated earlier in the file. In an index, nodes are moved from the end of the file to pages at the beginning of the file. In both cases pages are freed at the end of files and then returned to the file system. Databases can only be shrunk to the point that no free space is remaining; there is no data compression.

File Grow

Automated file growth greatly reduces the need for database management and eliminates many problems that occur when logs or databases run out of space. When creating a database, an initial size for the file must be given. SQL Server creates the data files based on the size provided by the database creator and data is added to the database these files fill. By default, data files are allowed to grow as much as necessary until disk space is exhausted. Alternatively, data files can be configured to grow automatically, but only to a predefined maximum size. This prevents disk drives from running out of space.

Allowing files to grow automatically can cause fragmentation of those files if a large number of files share the same disk. Therefore, it is recommended that files or filegroups be created on as many different local physical disks as available. Place objects that compete heavily for space in different filegroups.

Physical Database Architecture

Microsoft SQL Server version 7.0 introduces significant improvements in the way data is stored physically. These changes are largely transparent to general users, but do affect the setup and administration of SQL Server databases.

Pages and Extents

The fundamental unit of data storage in SQL Server is the page. In SQL Server version 7.0, the size of a page is 8 KB, increased from 2 KB. The start of each page is a 96-byte header used to store system information, such as the type of page, the amount of free space on the page, and the object ID of the object owning the page. There are seven types of pages in the data files of a SQL Server 7.0 database.

Page Type	Contains
Data	Data rows with all data except text, ntext, and image.
Index	Index entries
Log	Log records recording data changes for use in recovery
Text/Image	Text, ntext, and image data
Global Allocation Map	Information about allocated extents
Page Free Space	Information about free space available on pages
Index Allocation Map	Information about extents used by a table or index.

Torn Page Detection

Torn page detection helps insure database consistency. In SQL Server 7.0, pages are 8 KB, while Windows NT does I/O in 512-byte segments. This discrepancy makes it possible for a page to be partially written. This could happen if there is a power failure or other problem between the time when the first 512-byte segment is written and the completion of the 8 KB of I/O.

There are several ways to deal with this. One way is to use battery-backed cached I/O devices that guarantee all-or-nothing I/O. If you have one of these systems, torn page detection is unnecessary.

In SQL Server 7.0, you can enable torn page detection for a particular database by turning on a database option.

Locking Enhancements

Row-Level Locking

SQL Server 6.5 introduced a limited version of row locking on inserts. SQL Server 7.0 now supports full row-level locking for both data rows and index entries. Transactions can update individual records without locking entire pages. Many

OLTP applications can experience increased concurrency, especially when applications append rows to tables and indexes.

Dynamic Locking

SQL Server 7.0 has a superior locking mechanism that is unique in the database industry. At run time, the storage engine dynamically cooperates with the query processor to choose the lowest-cost locking strategy, based on the characteristics of the schema and query.

Dynamic locking has the following advantages:

- Simplified database administration, because database administrators no longer need to be concerned with adjusting lock escalation thresholds.
- Increased performance, because SQL Server minimizes system overhead by using locks appropriate to the task.
- Application developers can concentrate on development, because SQL Server adjusts locking automatically.

Multigranular locking allows different types of resources to be locked by a transaction. To minimize the cost of locking, SQL Server automatically locks resources at a level appropriate to the task. Locking at a smaller granularity, such as rows, increases concurrency but has a higher overhead because more locks must be held if many rows are locked. Locking at a larger granularity, such as tables, is expensive in terms of concurrency. However, locking a larger unit of data has a lower overhead because fewer locks are being maintained.

Lock Modes

SQL Server locks resources using different lock modes that determine how the resources can be accessed by concurrent transactions.
SQL Server uses several resource lock modes:

Lock mode	Description
Shared	Used for operations that do not change or update data (read-only operations), such as a SELECT statement.
Update	Used on resources that can be updated. Prevents a common form of deadlock that occurs when multiple sessions are reading, locking, and then potentially updating resources later.

Exclusive	Used for data-modification operations, such as UPDATE, INSERT, or DELETE. Ensures that multiple updates cannot be made to the same resource at the same time.
Intent	Used to establish a lock hierarchy.
Schema	Used when an operation dependent on the schema of a table is executing. There are two types of schema locks: schema stability and schema modification.

Table and Index Architecture

Overview

Fundamental changes were made in table organization. This new organization allows the query processor to make use of more nonclustered indexes, greatly improving performance for decision support applications. The query optimizer has a wide set of execution strategies and many of the optimization limitations of earlier versions of SQL Server have been removed. In particular, SQL Server 7.0 is less sensitive to index-selection issues, resulting in less tuning work.

Table Organization

The data for each table is now stored in a collection of 8-KB data pages. Each data page has a 96-byte header containing system information such as the ID of the table that owns the page and pointers to the next and previous pages for pages linked in a list. A row-offset table is at the end of the page. Data rows fill the rest of the page.

SQL Server 7.0 tables use one of two methods to organize their data pages:

- Clustered tables are tables that have a clustered index. The data rows are stored in order based on the clustered index key. The data pages are linked in a doubly linked list. The index is implemented as a b-tree index structure that supports fast retrieval of the rows based on their clustered index key values.

- Heaps are tables that have no clustered index. There is no particular order to the sequence of the data pages and the data pages are not linked in a linked list.

Table Indexes

A SQL Server index is a structure associated with a table that speeds retrieval of the rows in the table. An index contains keys built from one or more columns in the table. These keys are stored in a structure that allows SQL Server to quickly and efficiently find the row or rows associated with the key values. This structure is called a heap. The two types of SQL Server indexes are clustered and nonclustered indexes

Clustered Indexes

A clustered index is one in which the order of the values in the index is the same as the order of the data stored in the table.
The clustered index contains a hierarchical tree. When searching for data based on a clustered index value, SQL Server quickly isolates the page with the specified value and then searches the page for the record or records with the specified value. The lowest level, or leaf node, of the index tree is the page that contains the data.

Nonclustered Indexes

A nonclustered index is analogous to an index in a textbook. The data is stored in one place; the index is stored in another, with pointers to the storage location of the indexed items in the data. The lowest level, or leaf node, of a nonclustered index is the Row Identifier of the index entry, which gives SQL Server the location of the actual data row. The Row Identifier can have one of two forms. If the table has a clustered index, the identifier of the row is the clustered index key. If the table is a heap, the Row Identifier is the actual location of the data row, indicated with a page number and offset on the page. Therefore, a nonclustered index, in comparison with a clustered index, has an extra level between the index structure and the data itself.
When SQL Server searches for data based on a nonclustered index, it searches the index for the specified value to obtain the location of the rows of data and then retrieves the data from their storage locations. This makes nonclustered indexes the optimal choice for exact-match queries.

Some books contain multiple indexes. Since nonclustered indexes frequently store clustered index keys as their pointers to data rows, it is important to keep clustered index keys as small as possible.

SQL Server supports up to 249 nonclustered indexes on each table. The nonclustered indexes have a b-tree index structure similar to the one in clustered indexes. The difference is that nonclustered indexes have no effect on the order of the data rows. The collection of data pages for a heap is not affected if nonclustered indexes are defined for the table.

Data Type Changes

Unicode Data

SQL Server now supports Unicode data types, which makes it easier to store data in multiple languages within one database by eliminating the problem of converting characters and installing multiple code pages. Unicode stores character data using two bytes for each character rather than one byte. There are 65,536 different bit patterns in two bytes, so Unicode can use one standard set of bit patterns to encode each character in all languages, including languages such as Chinese that have large numbers of characters. Many programming languages also support Unicode data types.
The new data types that support Unicode are ntext, nchar, and nvarchar. They are the same as text, char, and varchar, except for the wider range of characters supported and the increased storage space used.

Improved Data Storage

Data storage flexibility is greatly improved with the expansion of the maximum limits for char, varchar, binary, and varbinary data types to 8,000 bytes, increased from 255 bytes. It is no longer necessary to use text and image data types for data storage for anything but very large data values. The Transact-SQL string functions also support these very long char and varchar values, and the SUBSTRING function can be used to process text and image columns. The handling of Nulls and empty strings has been improved. A new unique identifier data type is provided for storing a globally unique identifier (GUID).

Normalization

Normalization is the concept of analyzing the "inherent" or normal relationships between the various elements of a database. Data is normalized in different forms.

First normal form: Data is in first normal form if data of the tables is moved in to separate tables where data in each table is of a similar type, giving each table a primary key – a unique label or an identifier. This eliminates repeating groups of data.

Second normal form: Involves taking out data that is only dependent on part of key.

Third normal form: Involves removing the transitive dependencies. This means getting rid of any thing in the tables that doesn't depend Solely on the primary key. Thus, through normalization, effective data storage can be achieved eliminating redundancies and repeating groups.

SQL

The structured query language is used to manipulate data in the oracle database. It is also called SEQUEL.

SQL *plus- the user – friendly interface:

SQL *plus Is a superset of the standard SQL .it conforms to the standards of an SQL – compliant language and it has some specific oracle add – ones, leading to its name SQL and plus. SQL *plus was always called UFI (user –friendly interface). The oracle server only understands statements worded using SQL. Other front-end tools interact with the oracle database using the SQL statements. Oracle's implementation of SQL through SQL *plus is compliant with ANSI (American national standard institute) and the ISO (international standards organization) standards. Almost all oracle tools support identical SQL syntax

Data can be manipulated upon by using the Data Manipulation Language (DML). The DML statements provided by SQL are select, update, and delete. SQL *plus 3.3 can be accessed only by giving the valid username and password. This is one of the security features imposed by oracle to restrict unauthorized data accessed. SQL allows provides commands for creating new users, granting privileges etc.

All such features of SQL*plus make it a power data access tool especially for oracle products.

3.4 SYSTEM MODEL

3.4.1 SCENARIOS:

1. Scenario for the usecase ‘Myprofile’ :

Scenario name : ProfileRegistration

Participating actors instance : Likit :Client

Flow of events :

1) After successful login , Likit goes to the Home page

2) Likit clicks on Myprofile and enters the details as follows in the corresponding fields

Fmane : Venkata

Mname : Sai

Lname : Likit

Gender ○ **Male** ○ **Female**

SSN 558794

City Hyderabad

Andhra Pradesh

State

Zip 500013

Email likit@gmail.com

Occupation Software Engineer

Marital status ○ married ○ unmarried

3) After entering all the details, Likit presses the submit button

4) The details are stored in the database

2. A Scenario for the use case ‘ Schedule Interview ‘

Scenario name : AssigningInterviews

Participating actors instance : Mr. Rao : Admin

Flow of events :

1) After successful login , Mr. Rao goes to the Home page

2) Mr. Rao clicks on the Register and see who are all the new clients requesting for the Interveiw

3) Mr. Rao assign a staff member for the respective client who has requested for the interview on a particular day or if possible he arranges the

interview on the same day that was requested by client

4) Mr. Rao presses the schedule button and assign the interviewer as follows

Client name : Anil

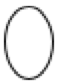
ClientScheduleDate : 05/21/2008

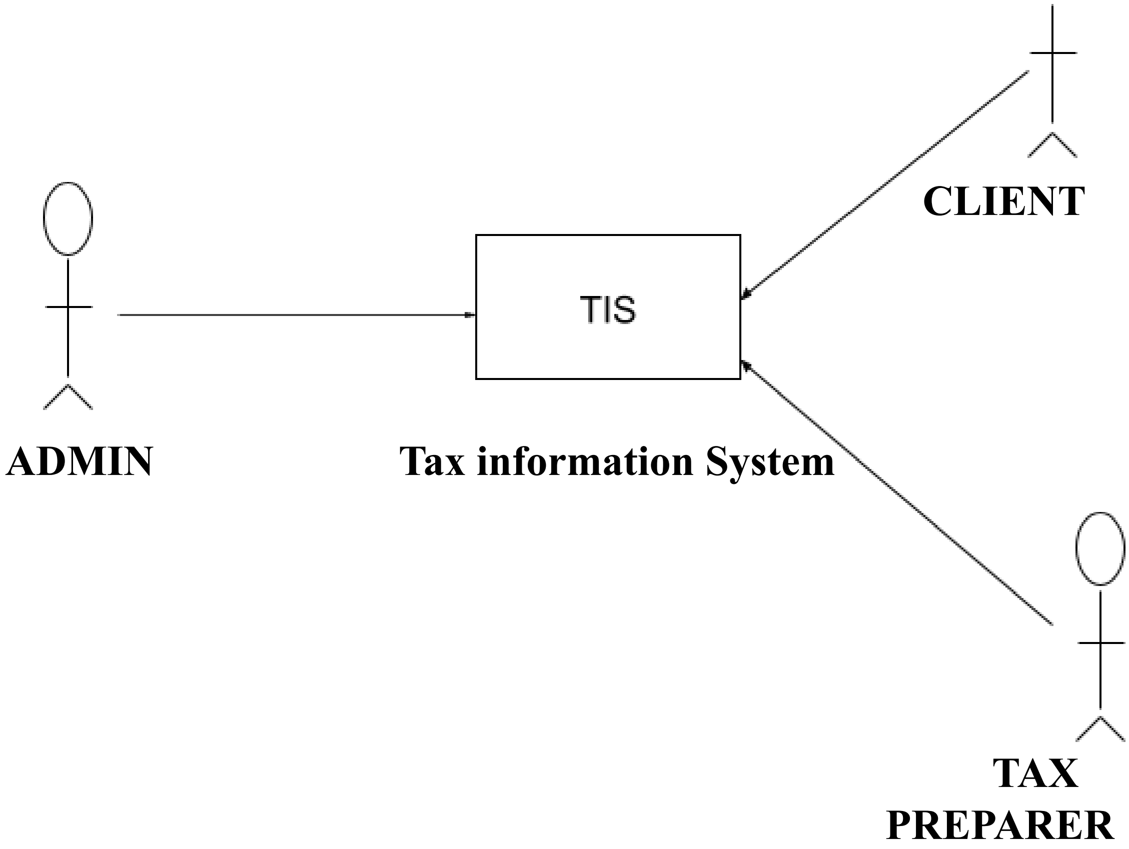
Admin given date : 05/25/2008

Assign to : Sudha

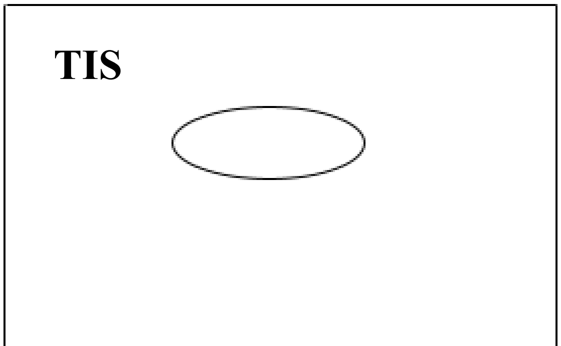
5) Finally Mr. Rao sends interview date and interviewer to Anil through mail.

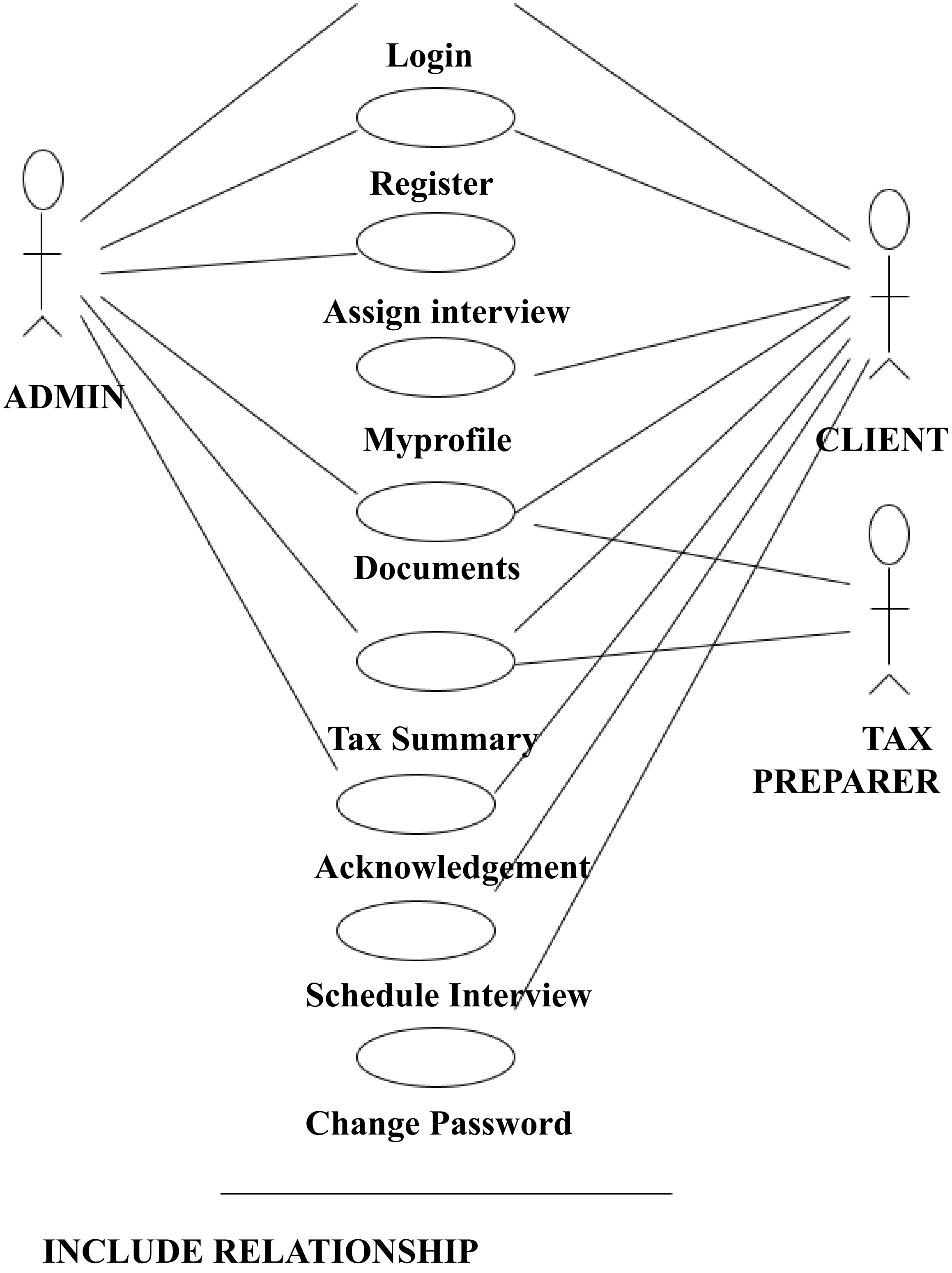
3.4.2 USE CASE MODEL:

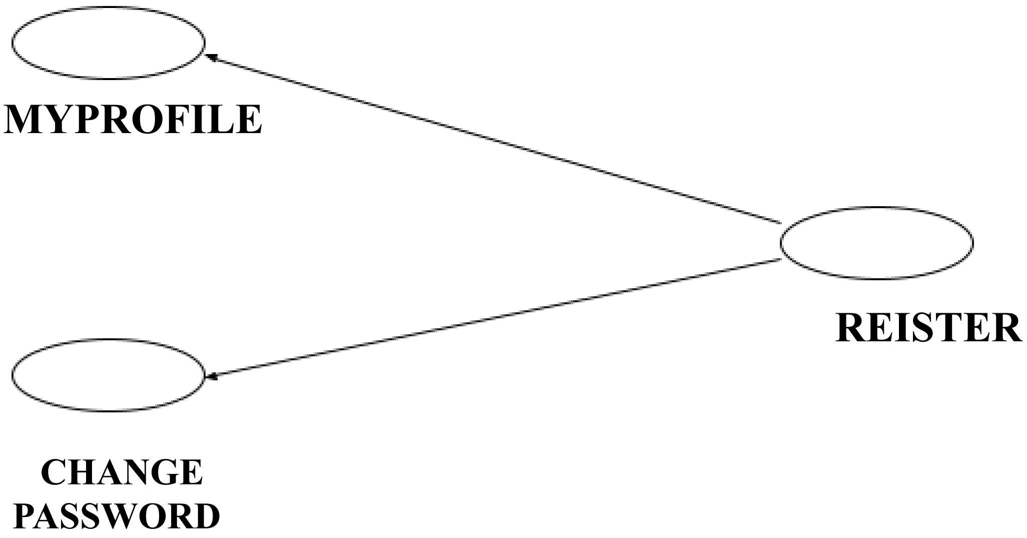




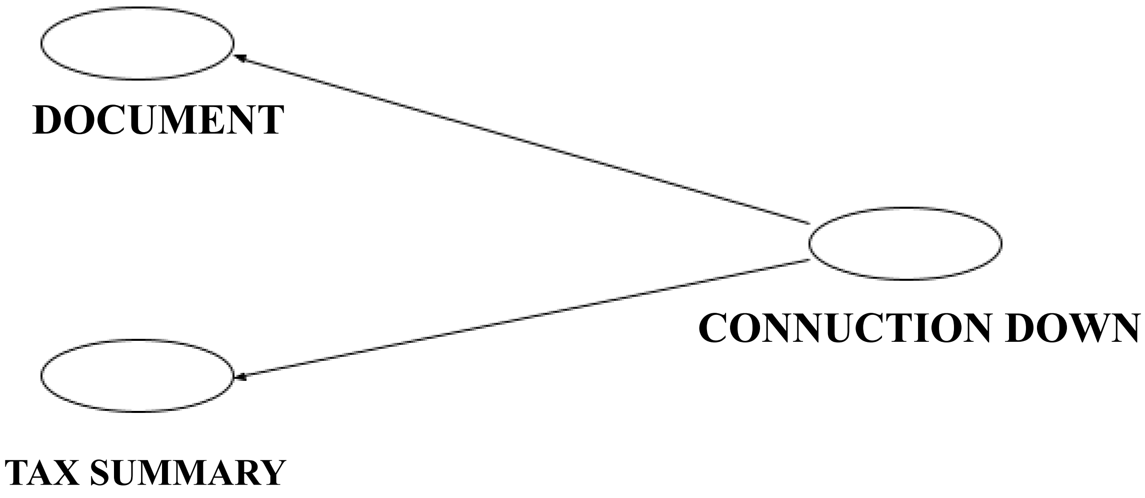
USE CASE DIAGRAM FOR TIS




Login
Register
Assign interview
Myprofile
Documents
Tax Summary
Acknowledgement
Schedule Interview
Change Password
ADMIN
CLIENT
TAX PREPARER
INCLUDE RELATIONSHIP



EXTEND RELATIONSHIP



USE CASE MODEL

Usecase name : Tax Summary

**Participating Actor instance : Calculated by Tax Preparer
Details are submitted by Admin
to Client**

**Entry condition : Tax Preparer takes the details that
are submitted my client**

Flow of events :

1) **Tax preparer invokes the TaxCalculation use case**

2) **He enters the details that are submitted by client**

3) **The documents submitted by the Client consist of Field, Rental, Insurance, Share, Mutual Fund.**

4) **The amount coated by the client for Field, Rental, Insurance, Share, Mutual Fund is deducted from the actual income of the client and for the remaining amount Tax is calculated that is 10% of remaining amount**

5) **Tax preparer foreword this to Admin**

6) **Admin mail the details to Client**

7) **Tax calculation is done only when to the client pays the tax calculation Admin**

Exit Condition : Client is acknowledge by receiving

mail

3.5 OBJECT MODEL:

3.5.1 DATA DICTIONARY

TABLE NAME : TAXREGISTRATION

FIELS NAME	DATATYPE
taxid	varchar
fname	varchar
mname	varchar
lname	varchar
phone	int
userid	varchar
pwd	varchar

TABLE NAME : ITINDEPENDENT

FIELS NAME	DATATYPE
taxid	varchar
fname	varchar
mname	varchar
lname	varchar
phone	int
pedate	datetime
vnumber	int
vdate	datetime
pbirth	varchar
addressinp	varchar
dateentryusa	datetime

TABLE NAME : ITINSPOUSE

FIELS NAME	DATATYPE
taxid	varchar
fname	varchar
mname	varchar
lname	varchar

phone	int
pedate	datetime
vnumber	int
vdate	datetime
pbirth	varchar
addressinp	varchar
dateentryusa	datetime

TABLE NAME : DOCUMENT

FIELS NAME	DATATYPE
taxid	varchar
dtype	varchar
dife	varchar
status	varchar

TABLE NAME : SPOUSE

FIELS NAME	DATATYPE
taxid	varchar
fname	varchar
mname	varchar
lname	varchar
dob	datetime
occupation	varchar
ssnitin	varchar

TABLE NAME : DEPENDENT

FIELS NAME	DATATYPE
taxid	varchar
fname	varchar
mname	varchar
lname	varchar
relation	varchar
other	varchar
dob	datetime
ssnitin	varchar

noofyearsinusa	int

TABLE NAME : INTERVIEW

FIELS NAME	DATATYPE
userid	varchar
pwd	varchar

TABLE NAME : VEHICLE

FIELS NAME	DATATYPE
make	varchar
model	varchar
pdate	varchar
bmileage	varchar
taxid	varchar

TABLE NAME : BANK

FIELS NAME	DATATYPE
bname	varchar
anumber	int
rnumber	int
atype	varchar
taxid	varchar

TABLE NAME : COMPOSE

FIELS NAME	DATATYPE
email	varchar
subject	varchar
attachments	varchar
date	datetime
body	varchar

TABLE NAME : PAYMENT

FIELS NAME	DATATYPE
taxid	varchar
bank	varchar
number	int
expdate	datetime

TABLE NAME : SCHEDULE

FIELS NAME	DATATYPE
taxid	varchar
sdate	datetime
statusconfirmation	varchar
assign	varchar

TABLE NAME : MYPROFILE

FIELS NAME	DATATYPE
taxid	varchar
fname	varchar
mname	varchar
lname	varchar
gendr	varcha
dob	datetime
ssninit	int
address	varchar
city	varchar
state	varchar
zip	int
email	varchar

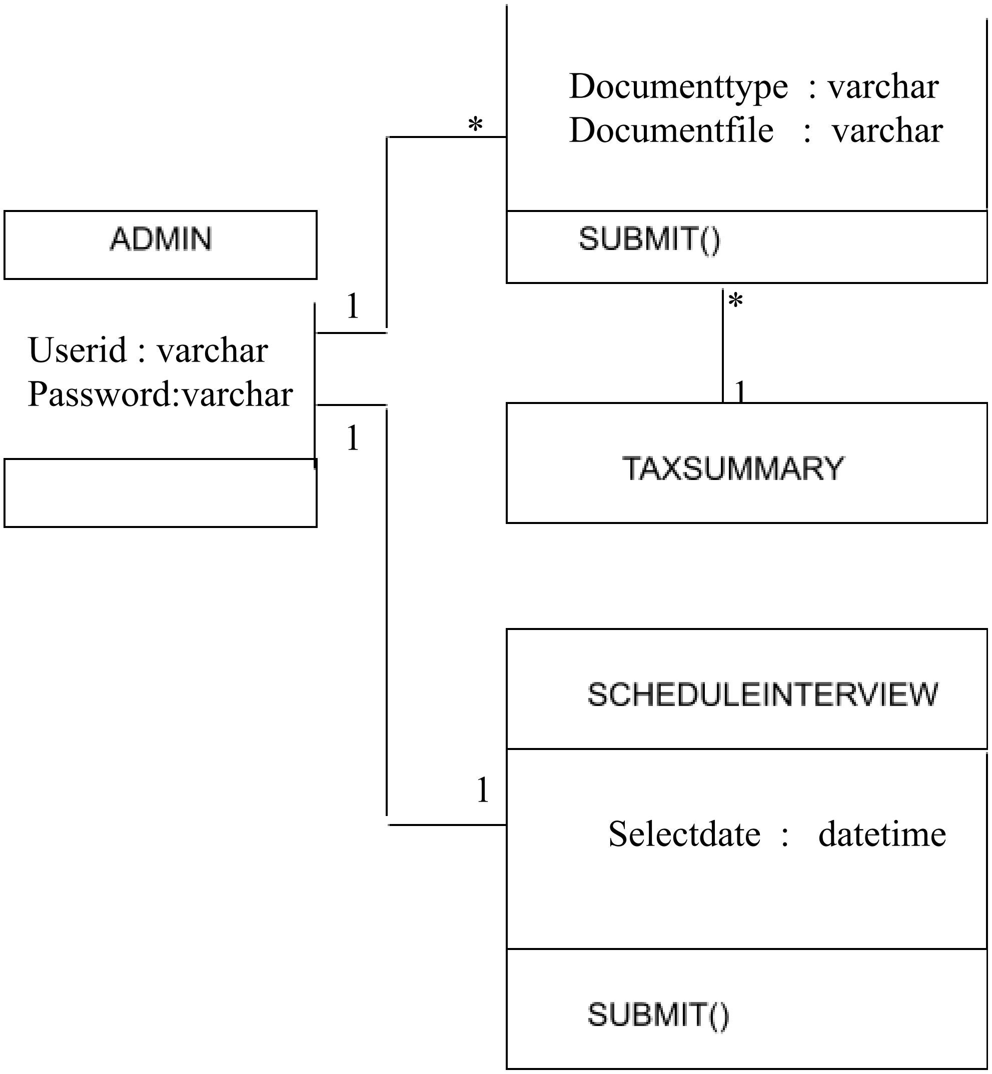
TABLE NAME : ADMINMAIL

FIELS NAME	DATATYPE
sender	varchar

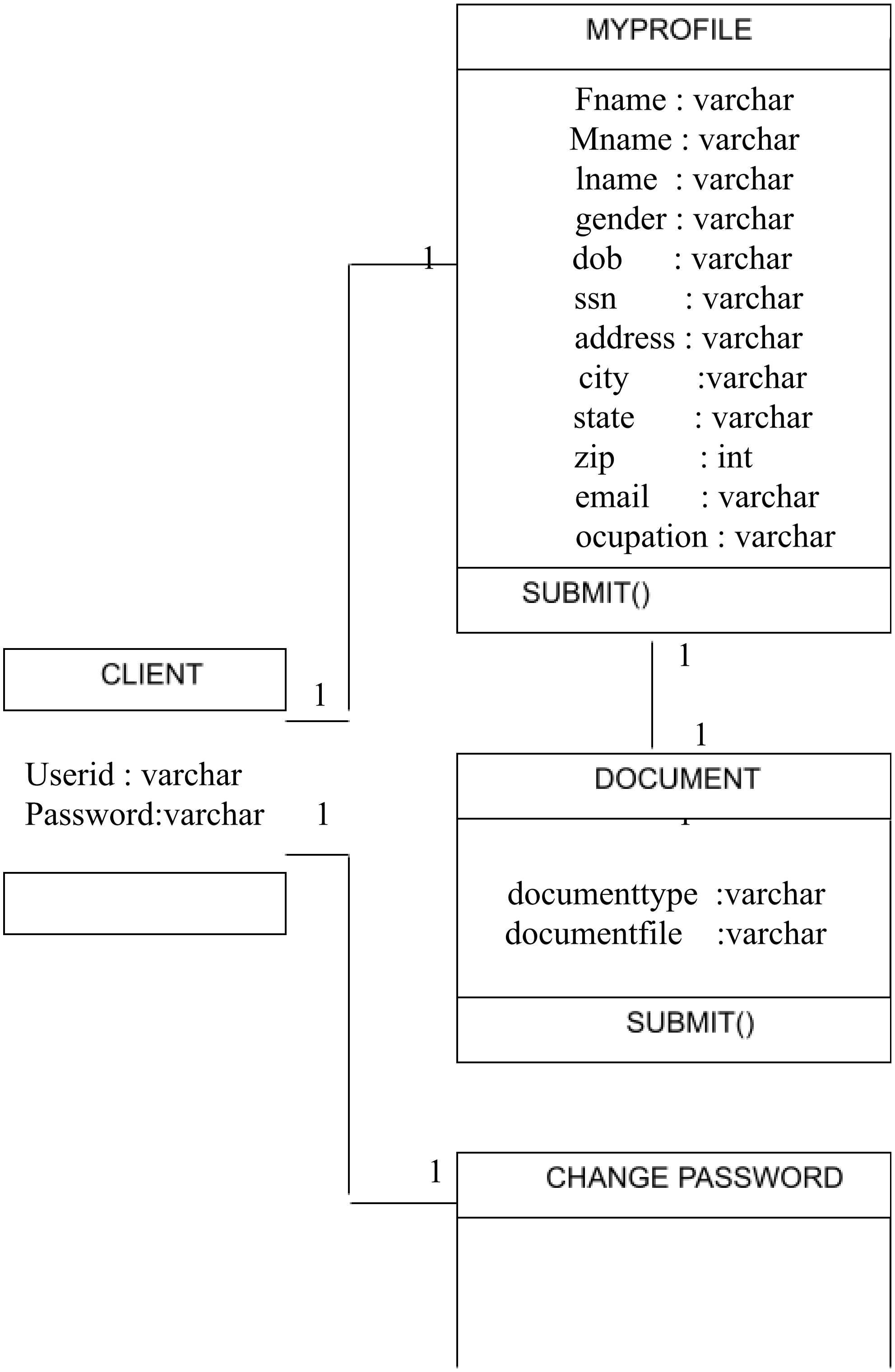
receiver	varchar
subject	varchar
attachments	varchar
body	varchar
senddate	datetime

3.5.2 CLASS DIAGRAMS

CLASS DIAGRAM FOR ADMIN



CLASS DIAGRAM FOR CLIENT

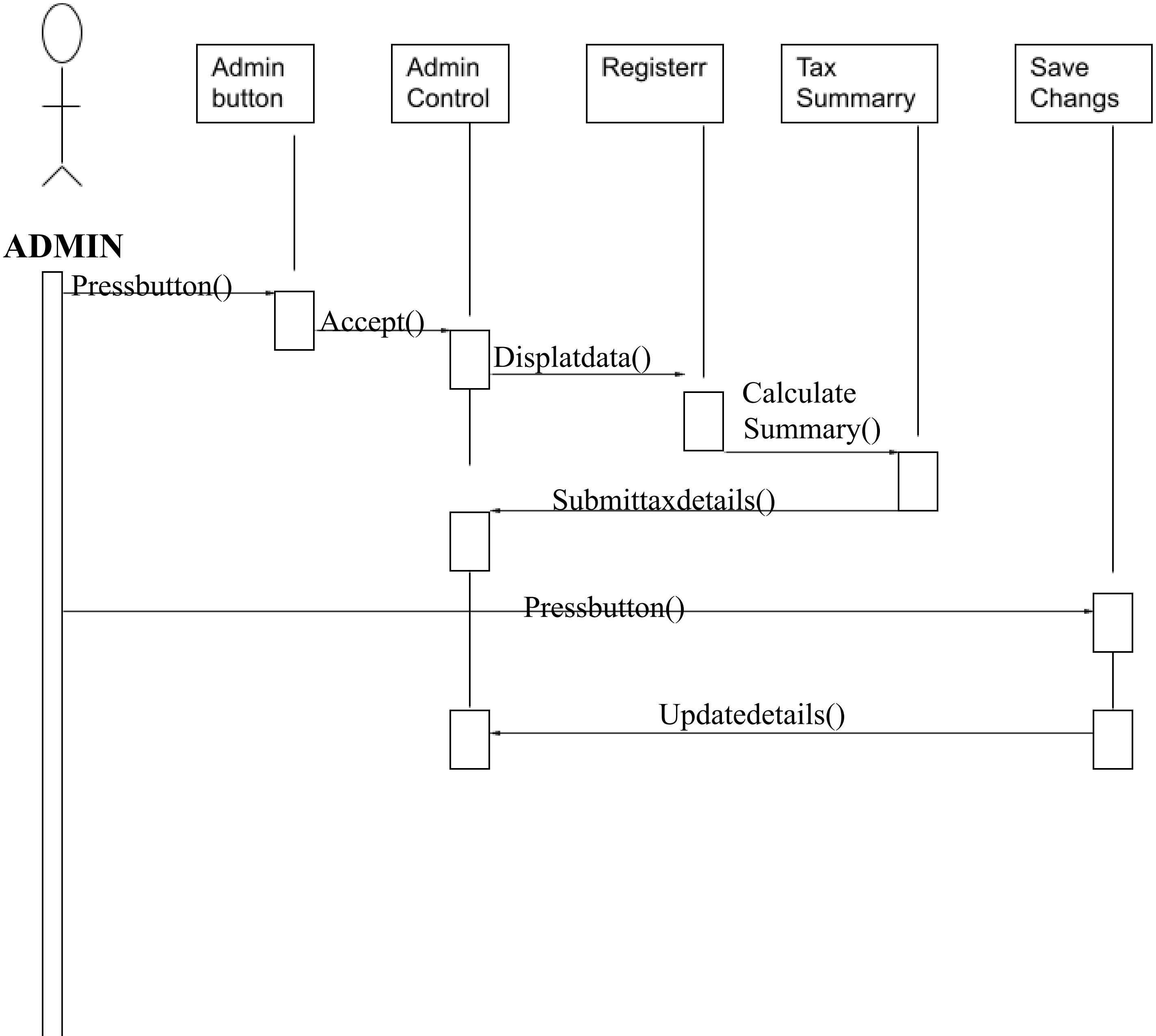


oldpwd : varchar
newpwd : varchar
confirmpwd : varchar

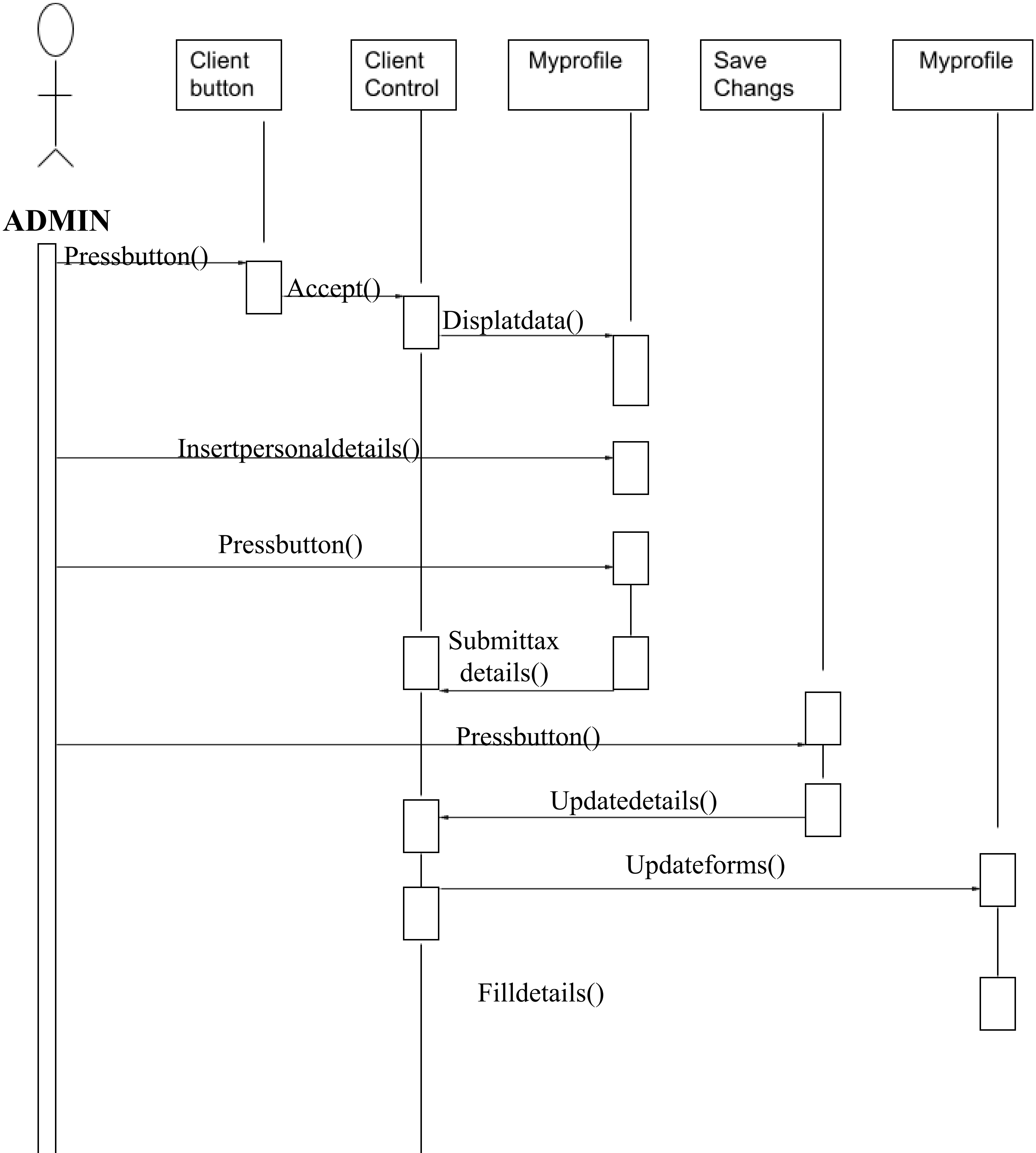
SUBMIT()

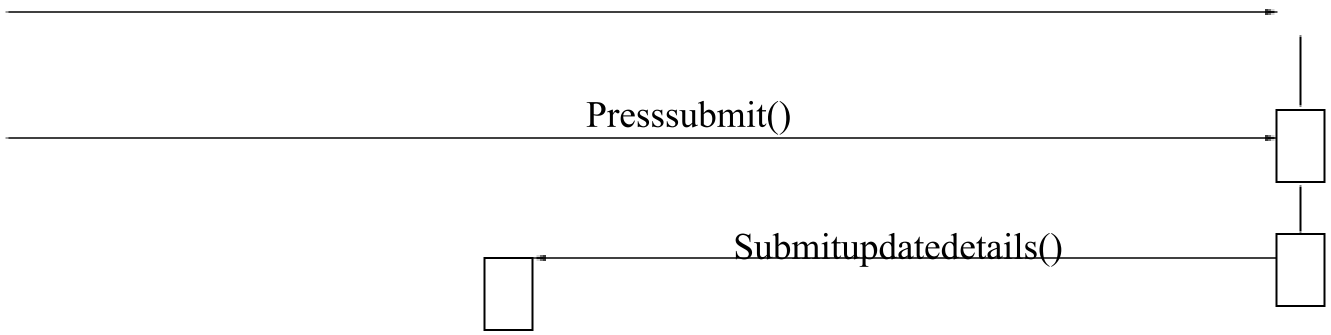
3.6 DYNAMIC MODEL

SEQUENCE DIGRAM FOR ADMIN

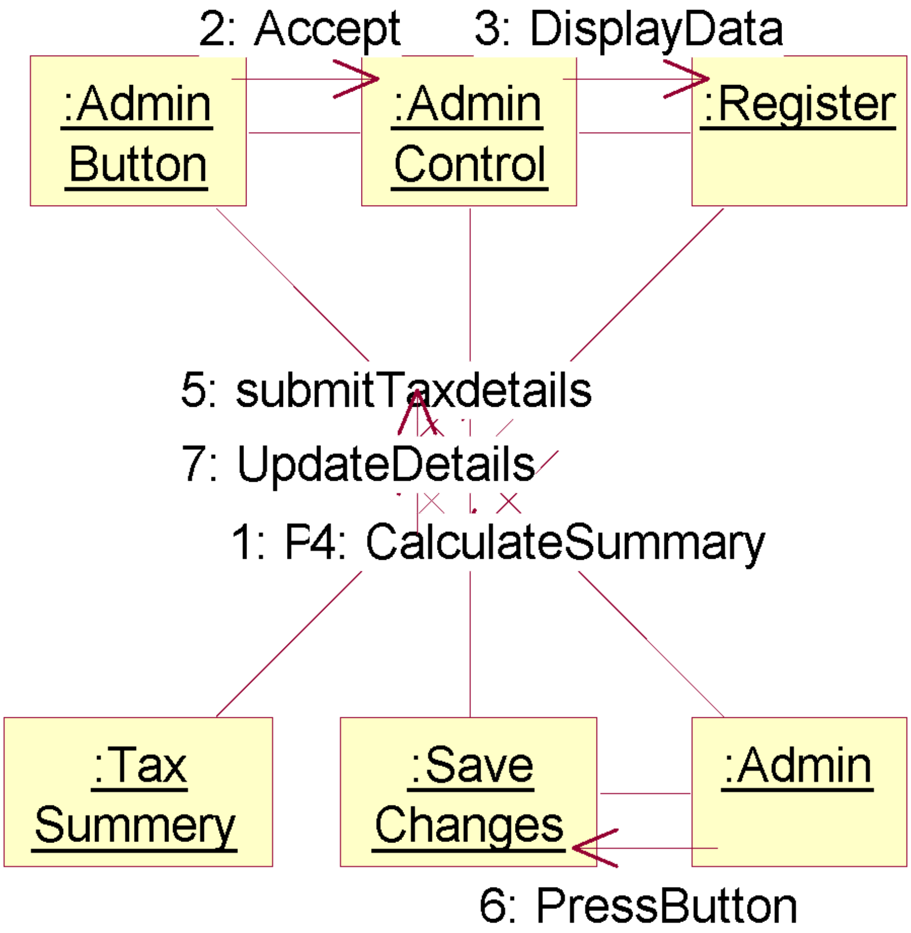


SEQUENCE DIGRAM FOR CLIENT

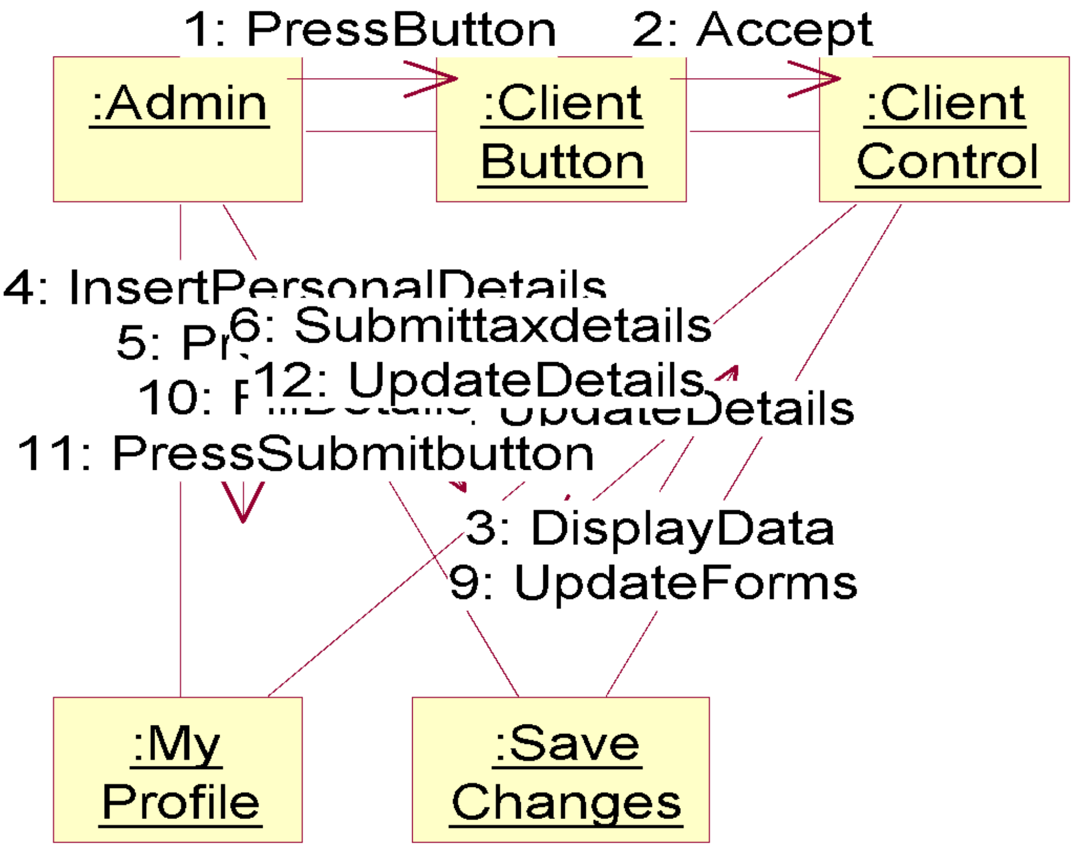



Presssubmit()
Submitupdatedetails()

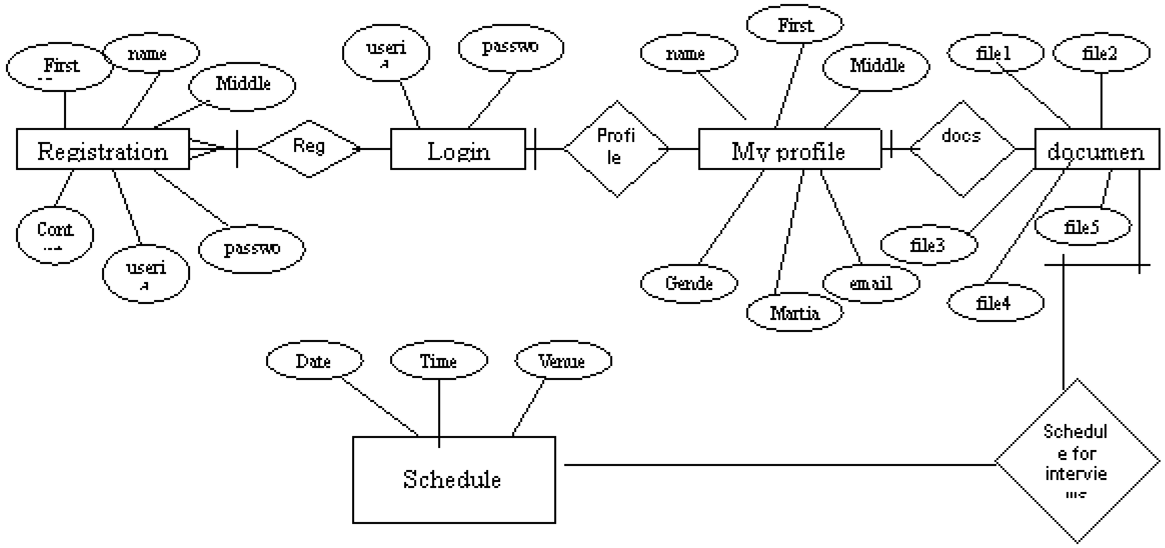
Collabration Diagrams
Admin

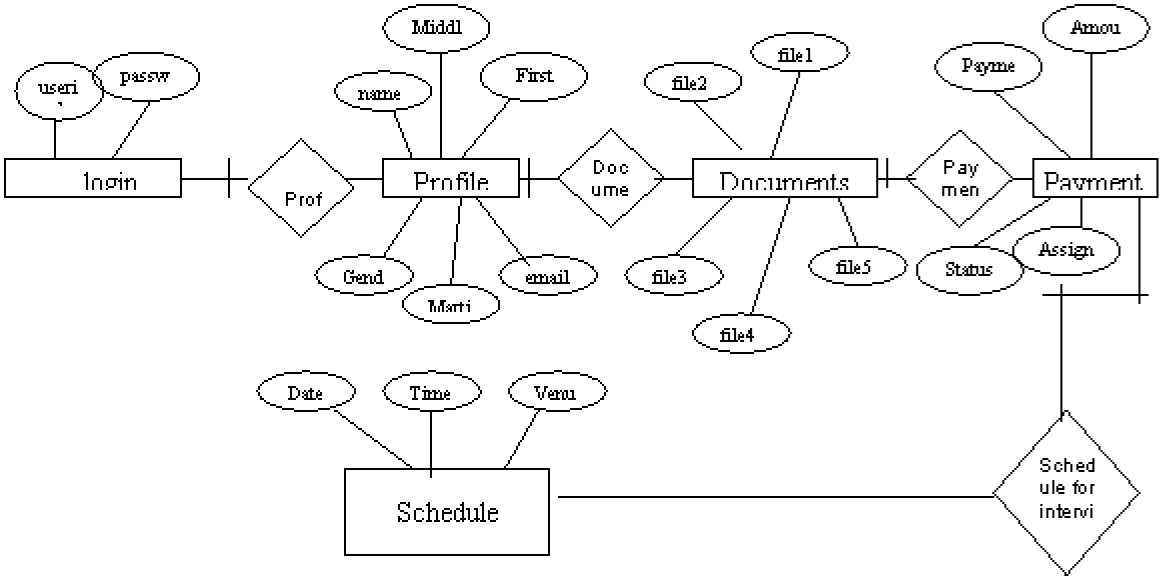


Client

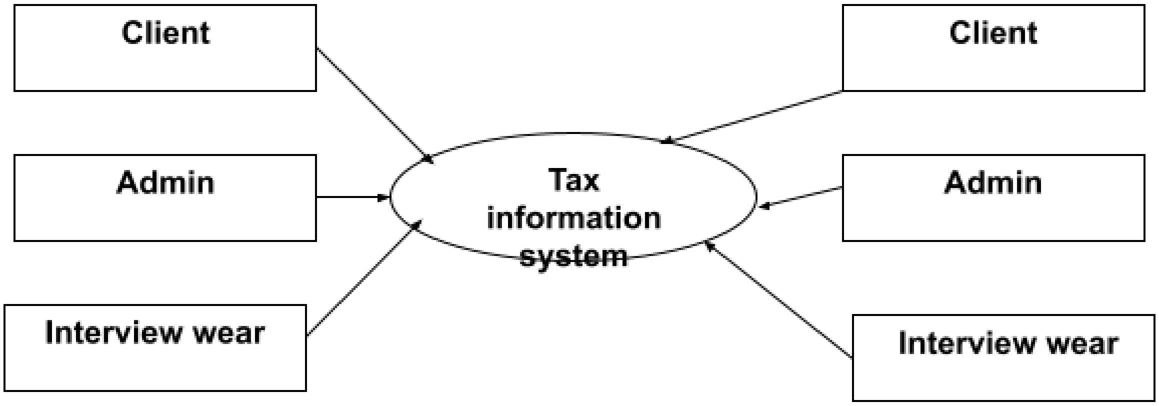


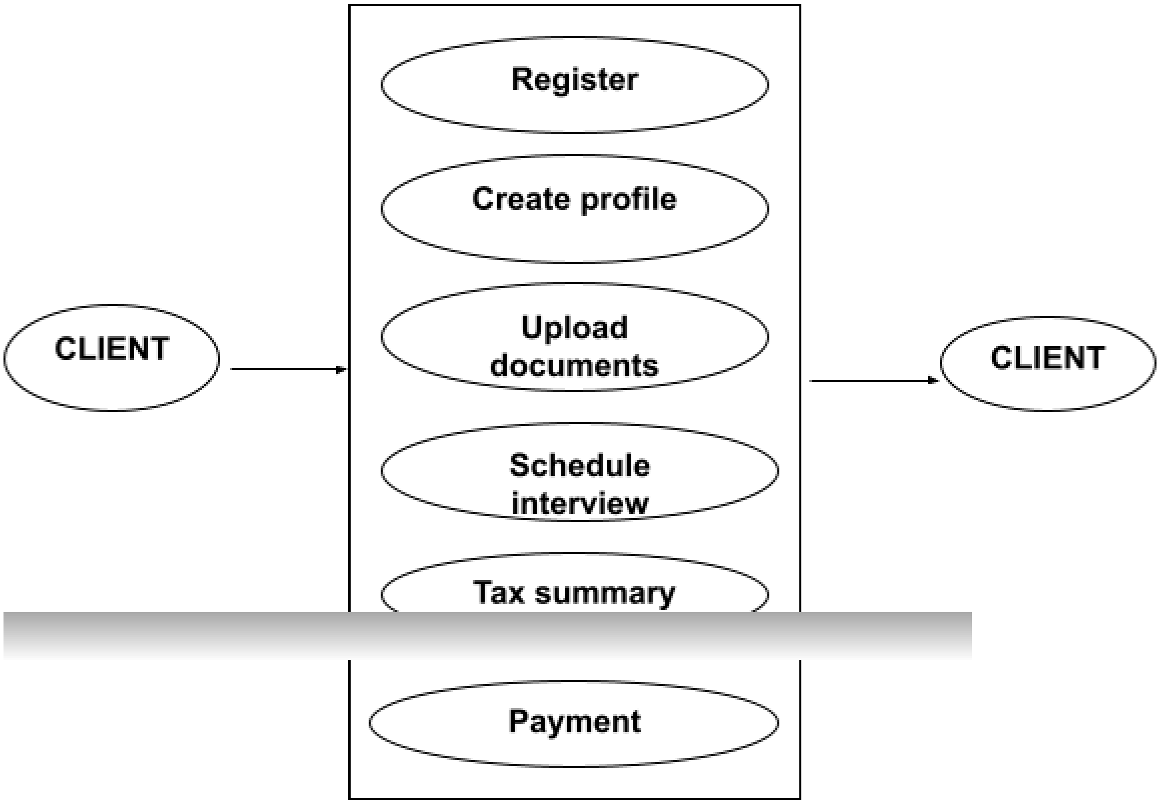
ER-Diagram

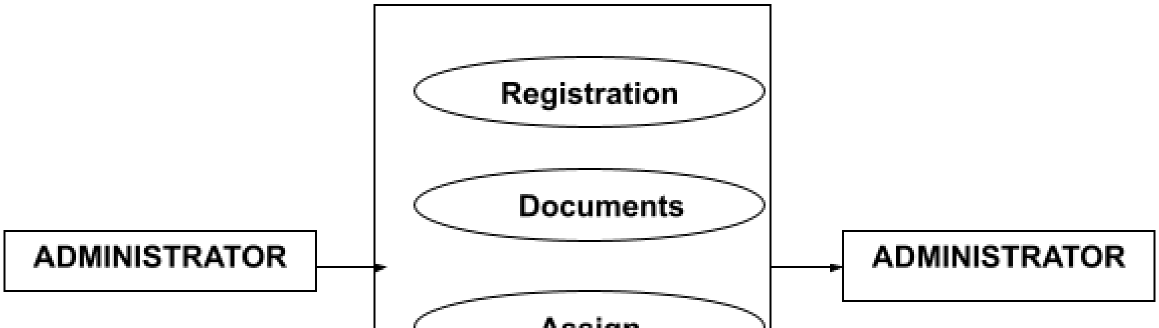


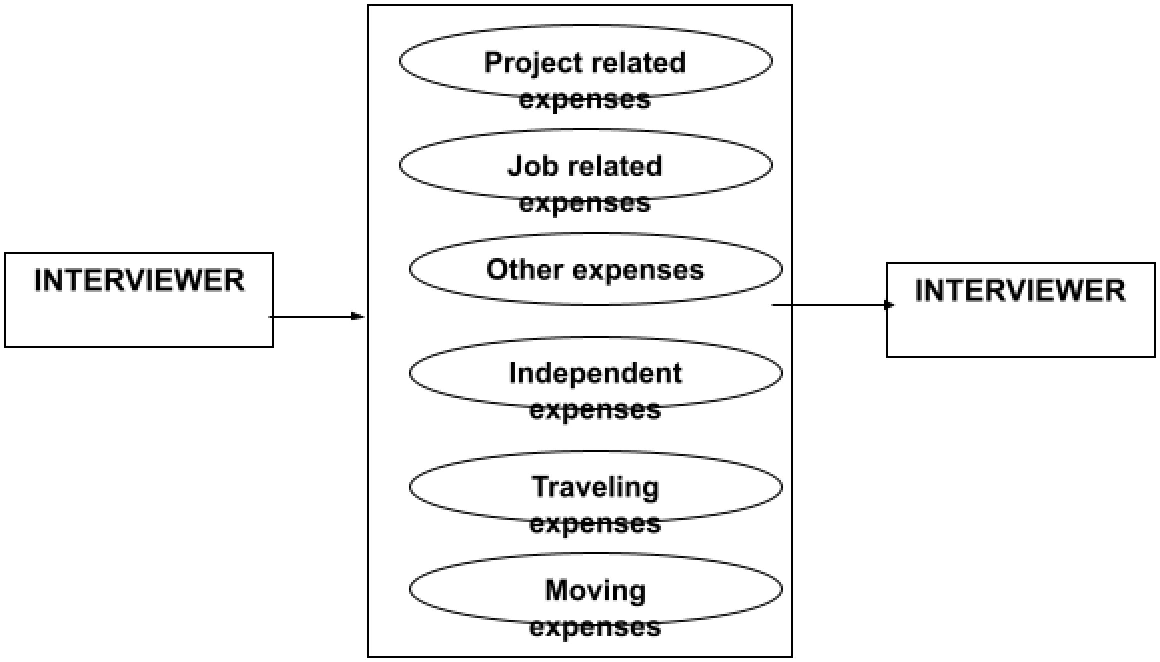

useri
passw
login
Prof
Middl
name
First
Profile
Gend
Marti
email
Doc ume
file2
file1
Documents
file3
file4
file5
Pay men
Payme
Amou
Payment
Status
Assign
Date
Time
Venu
Schedule
Sched ule for intervi

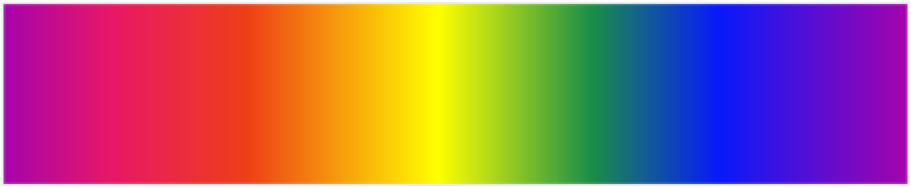
DFDs

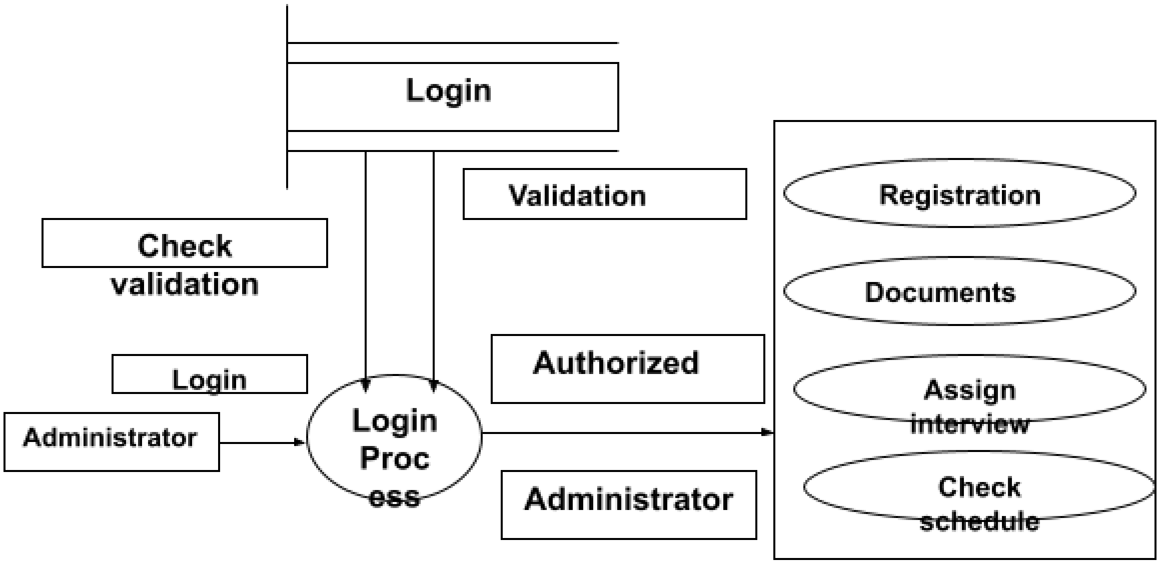

Client
Admin
Interview wear
Tax information system
Client
Admin
Interview wear

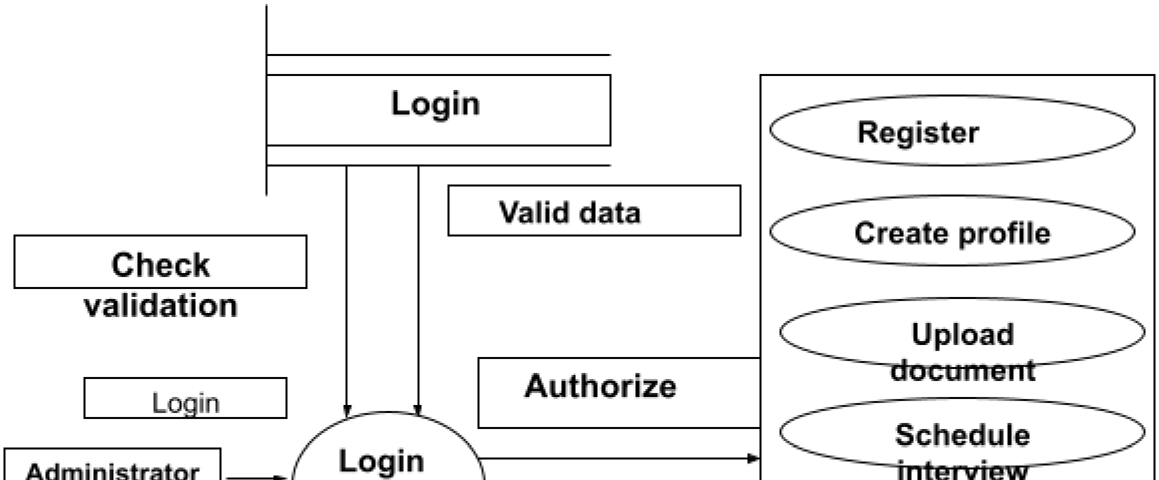

CLIENT
Register
Create profile
Upload documents
Schedule interview
Tax summary
Payment
CLIENT

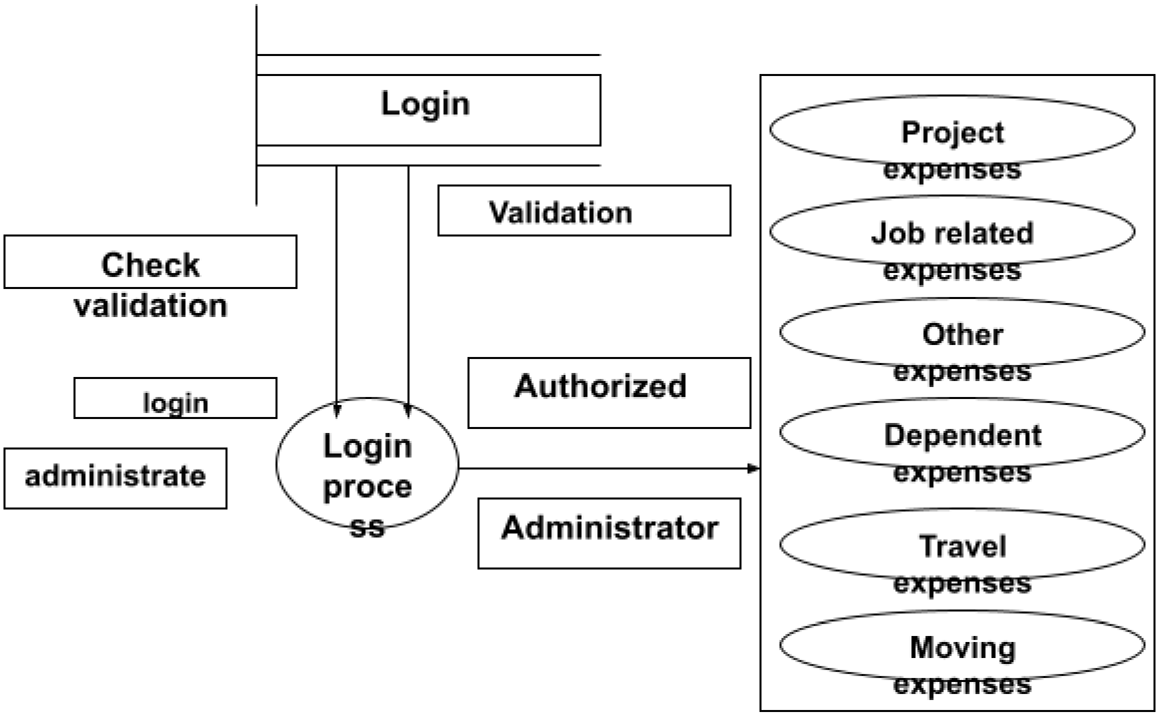

ADMINISTRATOR
Registration
Documents
Assign
ADMINISTRATOR

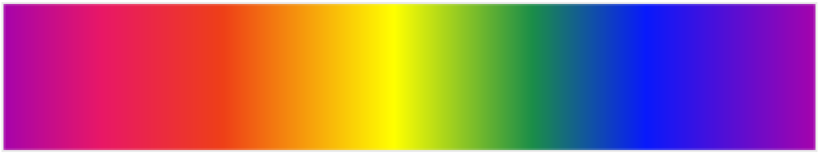

INTERVIEWER
Project related expenses
Job related expenses
Other expenses
Independent expenses
Traveling expenses
Moving expenses
INTERVIEWER

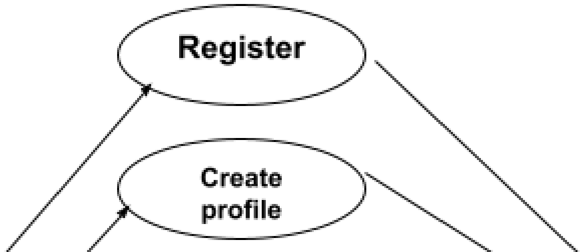


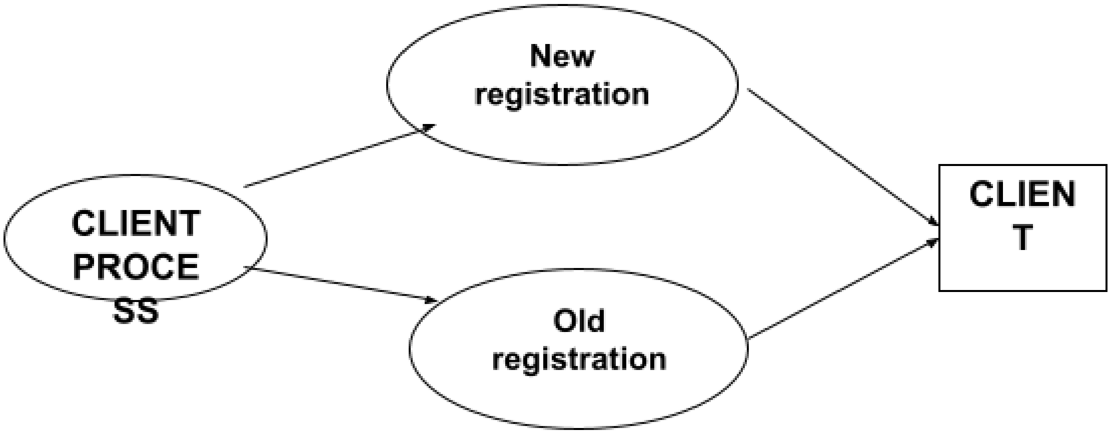

Login
Validation
Check validation
Login
Administrator
Login Proc ess
Authorized
Administrator
Registration
Documents
Assign interview
Check schedule

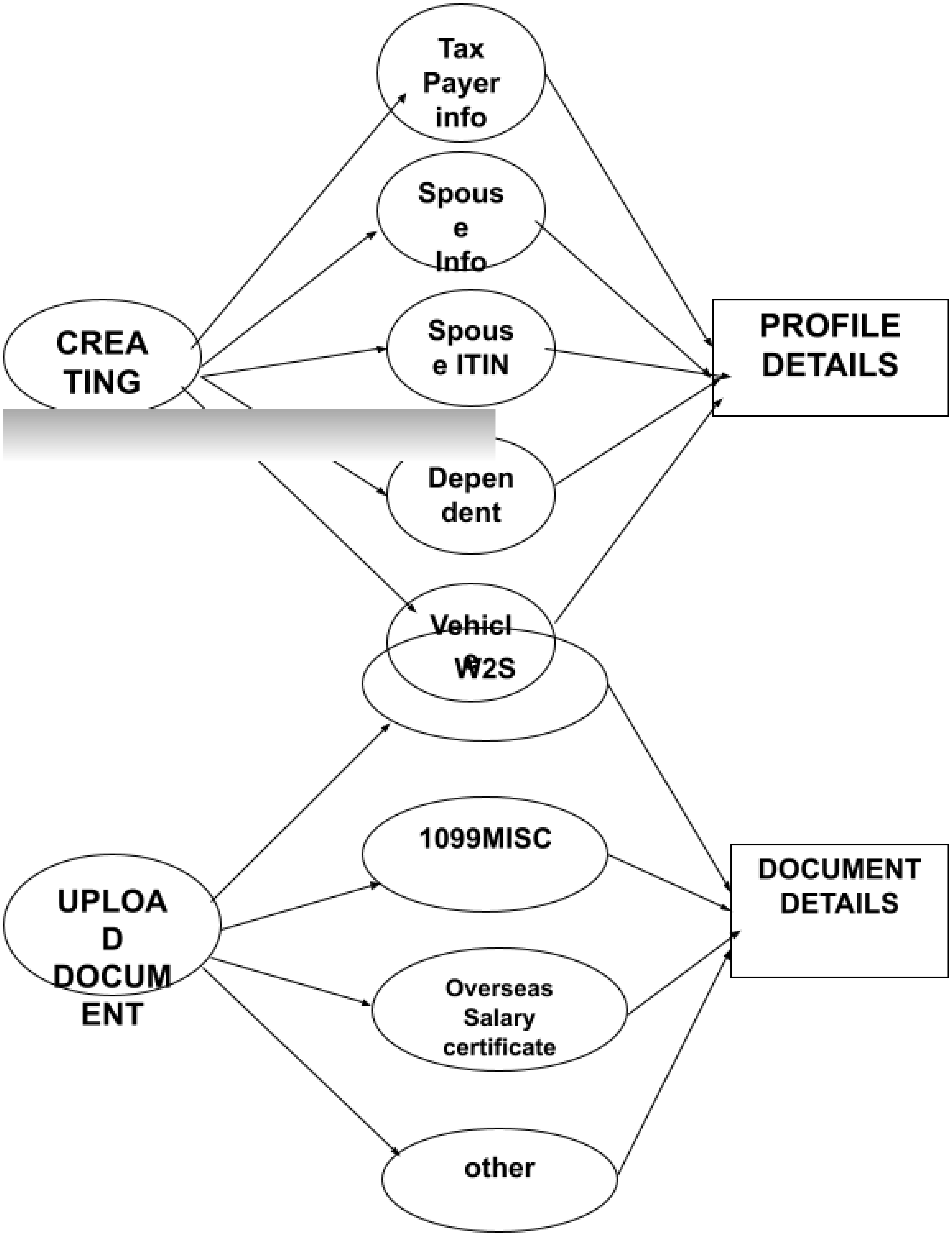

Login
Valid data
Check validation
Login
Administrator
Login
Authorize
Register
Create profile
Upload document
Schedule interview

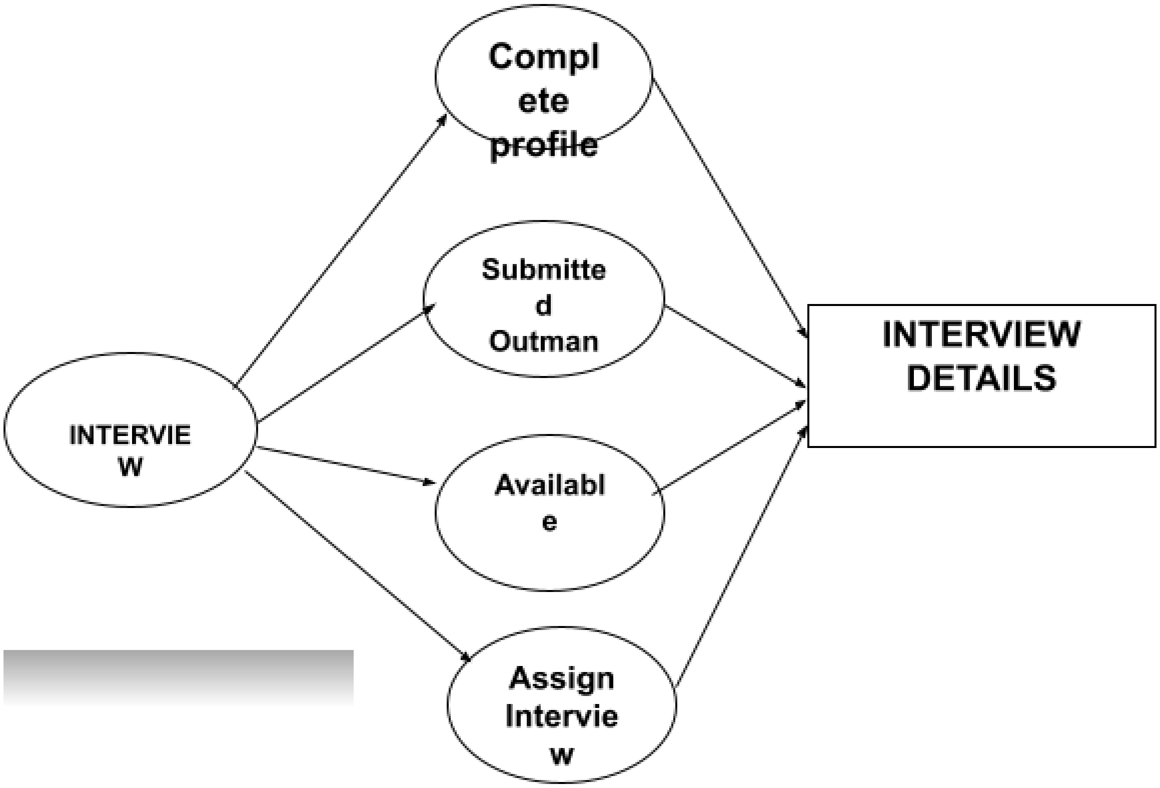

Login
Validation
Check validation
login
administrate
Login proce ss
Authorized
Administrator
Project expenses
Job related expenses
Other expenses
Dependent expenses
Travel expenses
Moving expenses

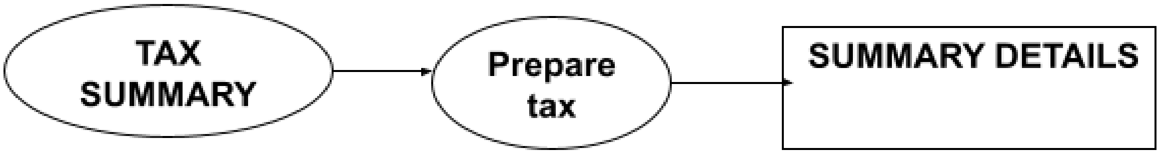


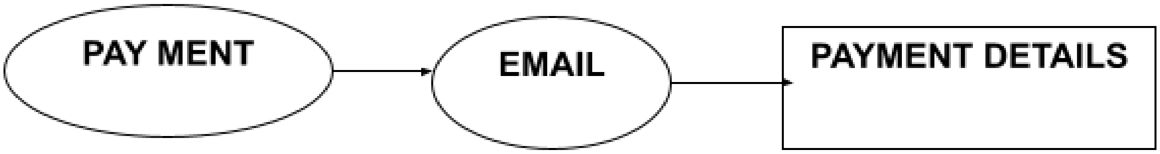

Register
Create profile

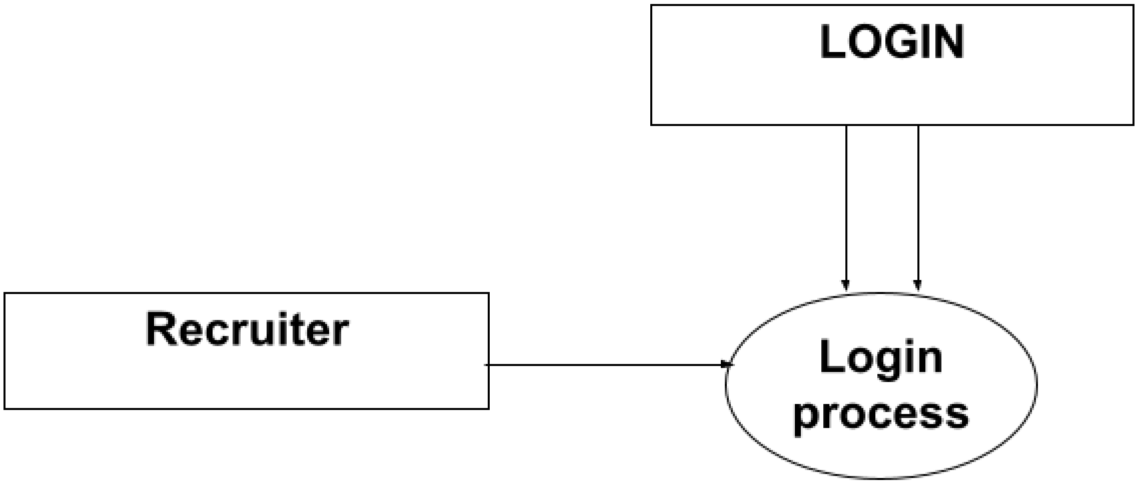

CLIENT PROCE SS
New registration
Old registration
CLIEN T

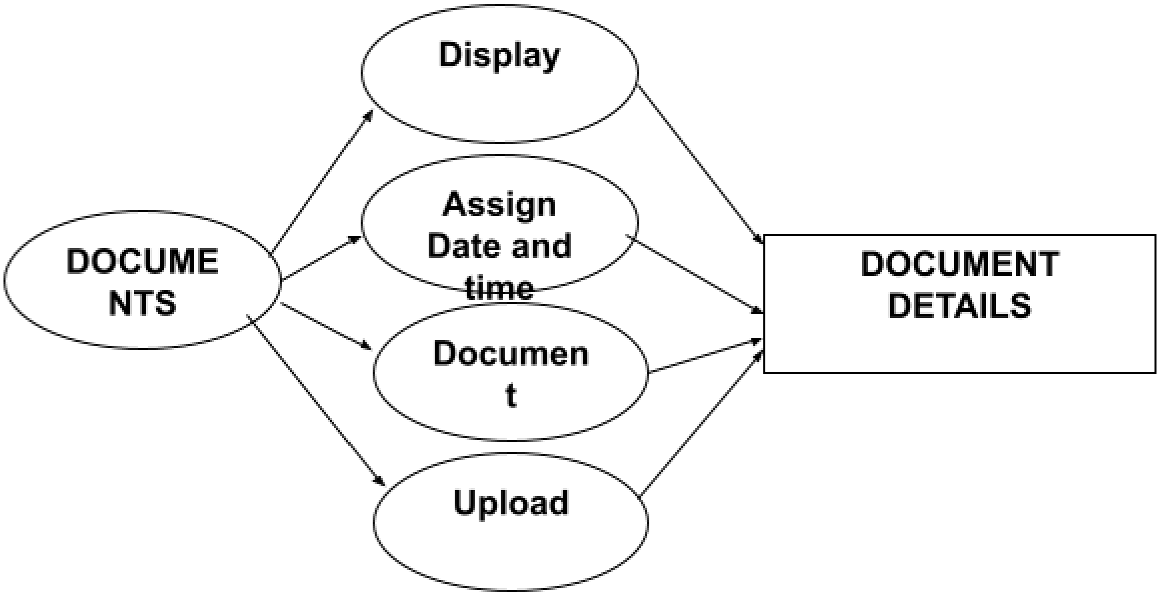

CREATING
Tax Payer info
Spouse Info
Spouse ITIN
Dependent
Vehicle
PROFILE DETAILS
UPLOAD DOCUMENT
W2S
1099MISC
Overseas Salary certificate
other
DOCUMENT DETAILS

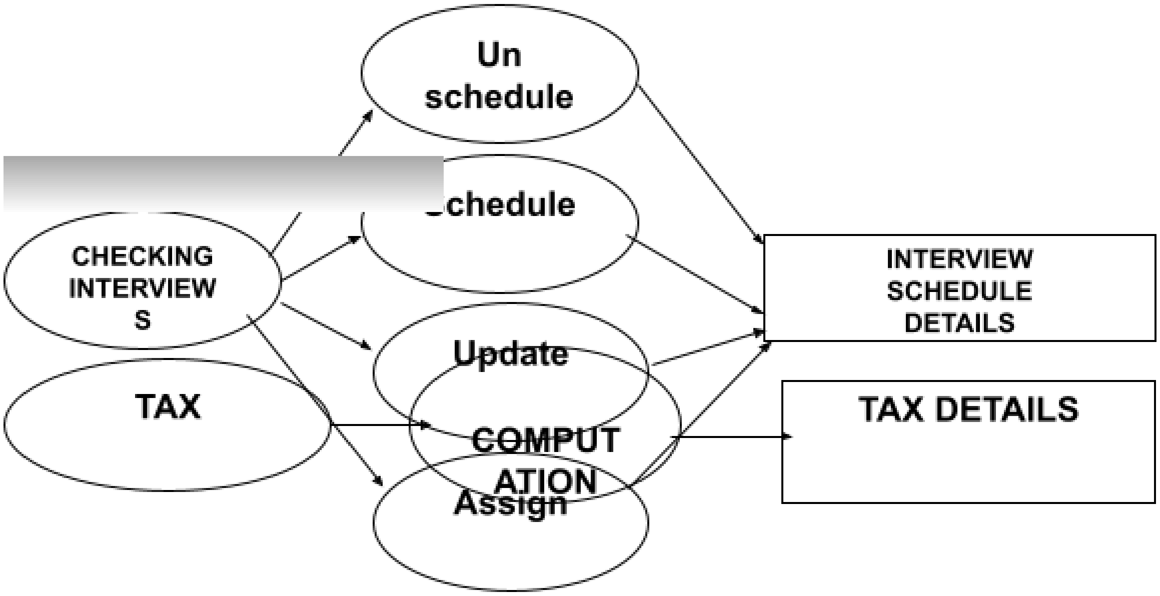

INTERVIE W
Compl ete profile
Submitte d Outman
Availabl e
Assign Intervie w
INTERVIEW DETAILS


TAX SUMMARY
Prepare tax
SUMMARY DETAILS


PAY MENT
EMAIL
PAYMENT DETAILS


LOGIN
Recruiter
Login process


DOCUMENTS
Display
Assign Date and time
Document
Upload
DOCUMENT DETAILS


CHECKING INTERVIEWS
Un schedule
Schedule
Update
Assign
INTERVIEW SCHEDULE DETAILS
TAX
COMPUTATION
TAX DETAILS

System Design

4.SYSTEM DESIGN

DEFINITION:

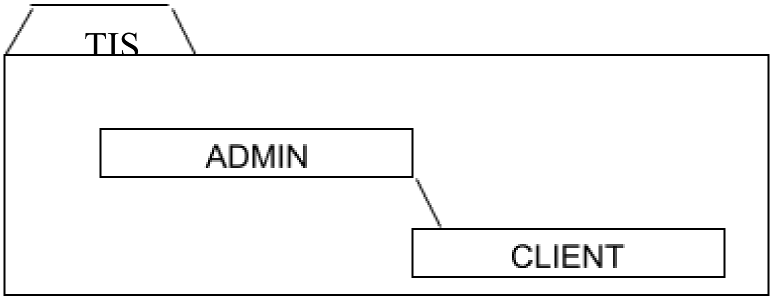
Design is the first step in the development phase for any engineered product/system. It may be defined as “The process of applying various techniques and principles for the purpose of defining a device, a process/a system in sufficient detail to permit its physical realization.
Once the software requirements have been analyzed and specified, the software design involves three technical activities—Design, Code generation and testing, that are required to build and verify the software.
The design activities are of main importance in this phase, because in this activity decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software/
a system.

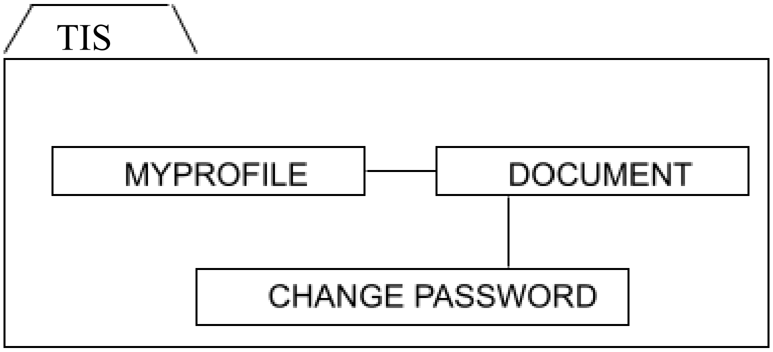
4.1SUBSYSTEM DECOMPOSITION

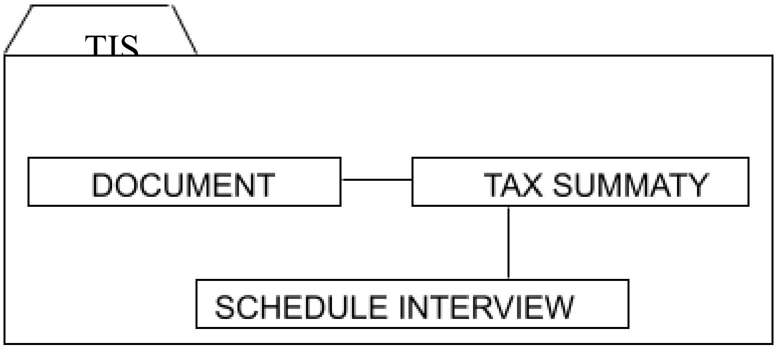
Subsystem decomposition is to find subsystem and to keep track of related objects together. Subsystems are merged into one subsystem, a complex subsystem are added to take care of new functionality. The first iterations over the subsystem decomposition can introduce drastic changes in the system design changes in the system design model.

The subsystems identified are :

TIS
Admin
Client

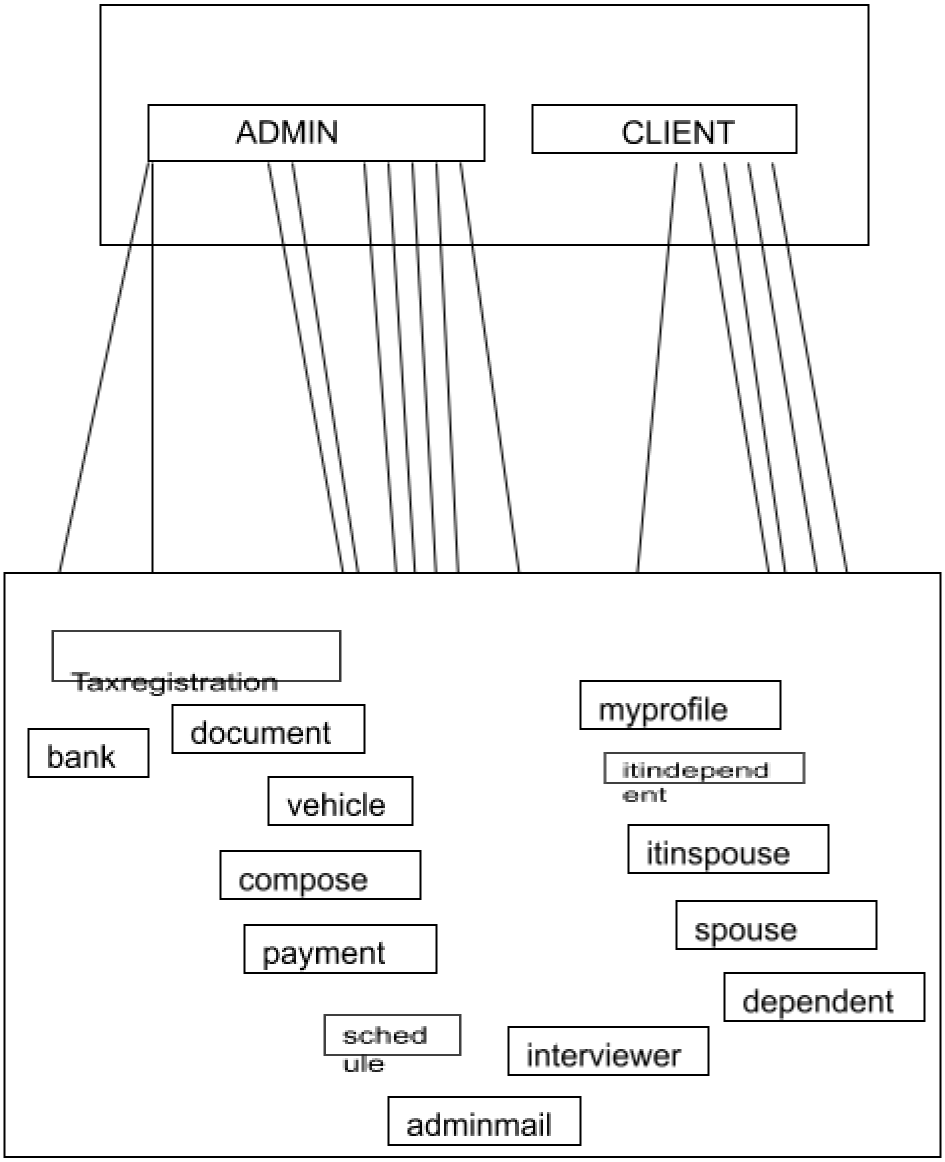




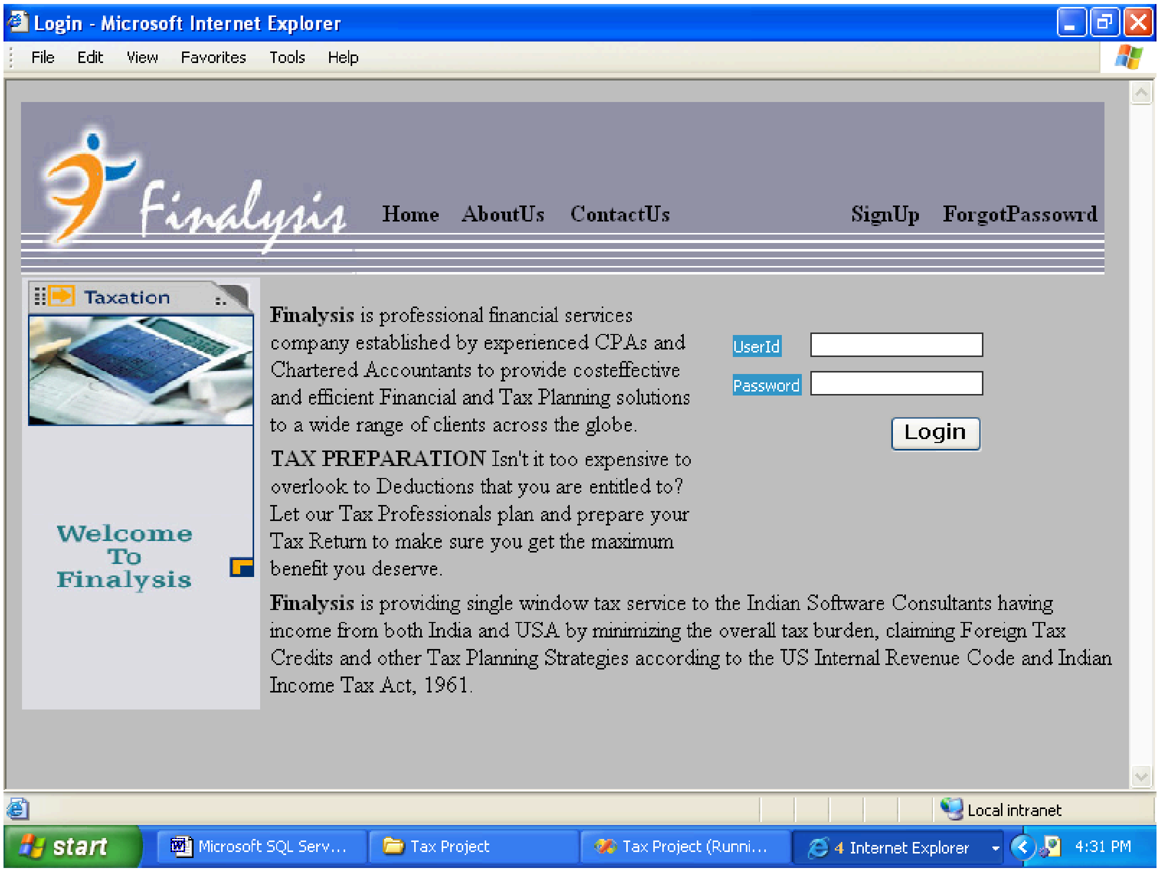


4.1.2 PRESISTANT DATA MANAGEMENT

Persistent data outlive a single execution of the system. It describes the persistent data stored by the system and data management infrastructure required for it . It includes the description if the encapsulation of database.



Input forms



Finalysis is professional financial services company established by experienced CPAs and Chartered Accountants to provide costeffective and efficient Financial and Tax Planning solutions to a wide range of clients across the globe.

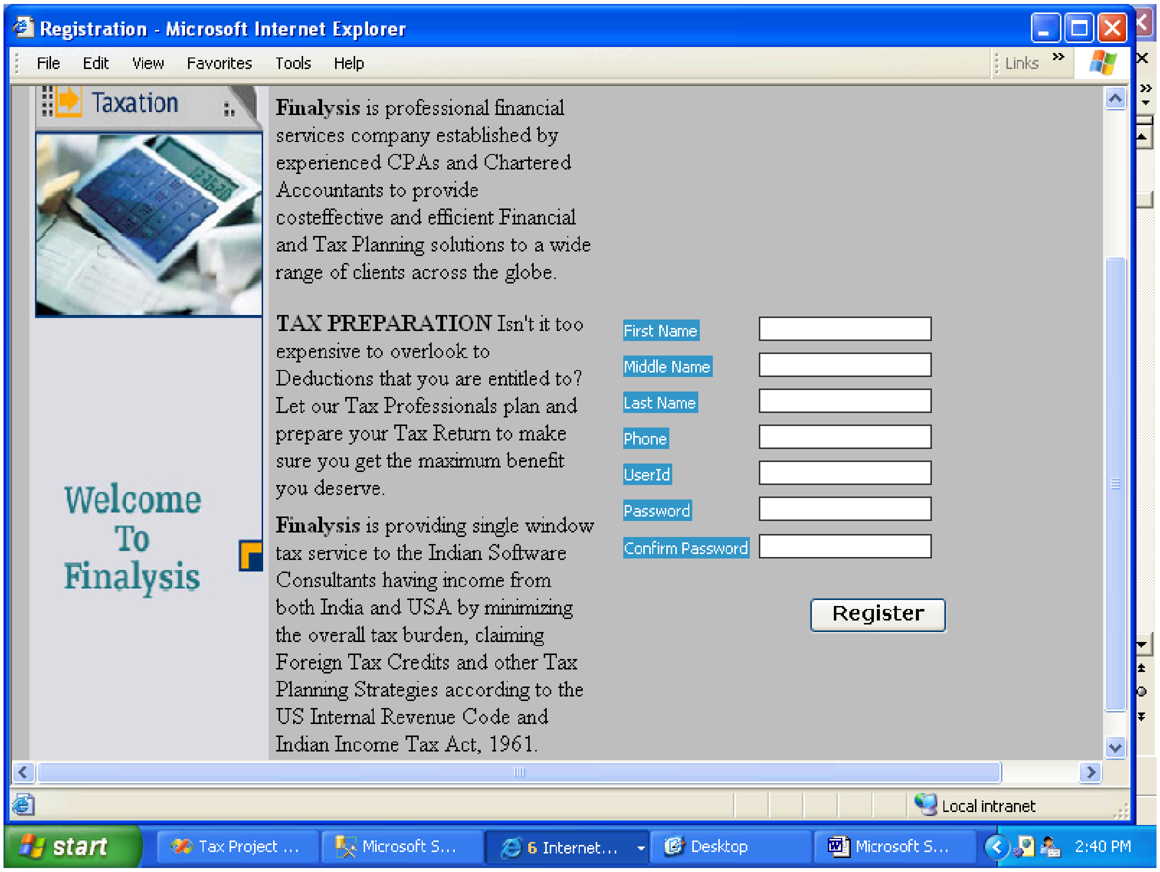
TAX PREPARATION Isn't it too expensive to overlook to Deductions that you are entitled to? Let our Tax Professionals plan and prepare your Tax Return to make sure you get the maximum benefit you deserve.

Finalysis is providing single window tax service to the Indian Software Consultants having income from both India and USA by minimizing the overall tax burden, claiming Foreign Tax Credits and other Tax Planning Strategies according to the US Internal Revenue Code and Indian Income Tax Act, 1961.

UserId

Password

Login



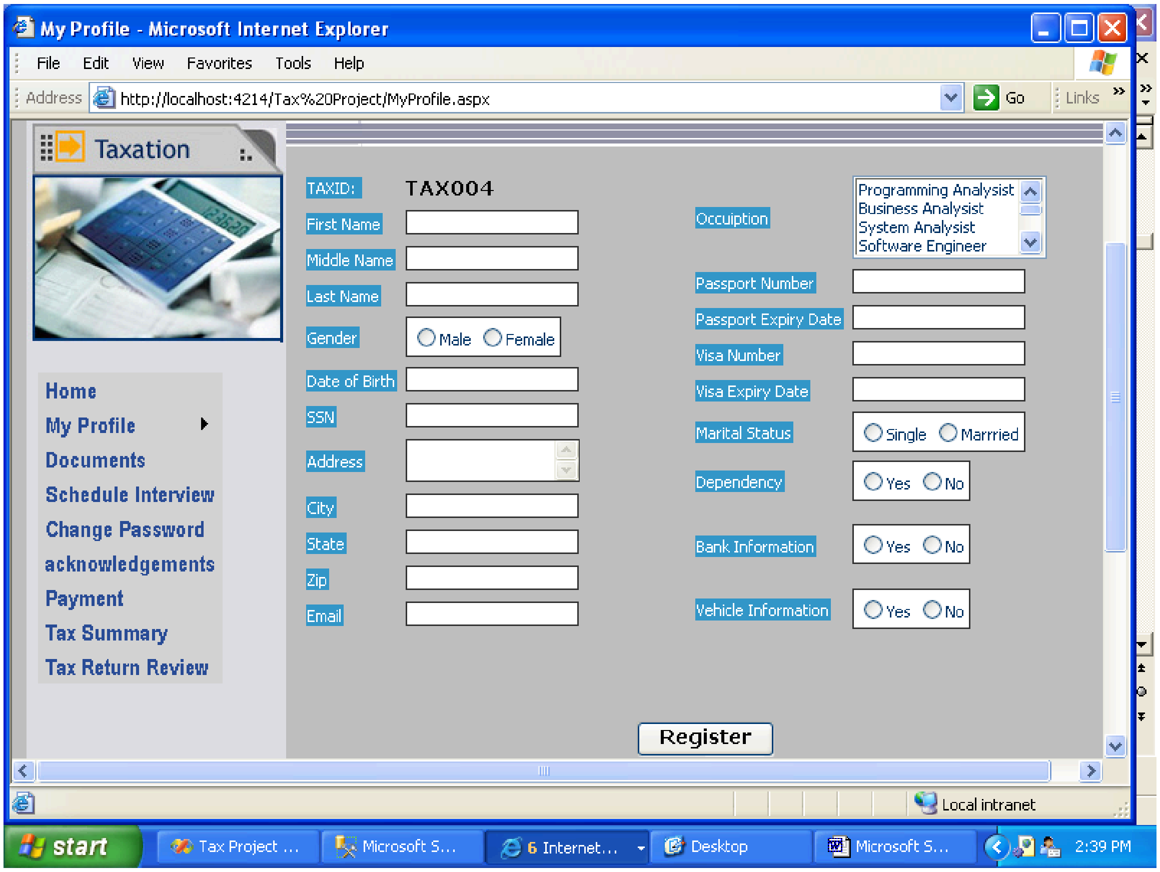
Registration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Taxation
Welcome To Finalysis

Finalysis is professional financial services company established by experienced CPAs and Chartered Accountants to provide costeffective and efficient Financial and Tax Planning solutions to a wide range of clients across the globe.

TAX PREPARATION Isn't it too expensive to overlook to Deductions that you are entitled to? Let our Tax Professionals plan and prepare your Tax Return to make sure you get the maximum benefit you deserve.

Finalysis is providing single window tax service to the Indian Software Consultants having income from both India and USA by minimizing the overall tax burden, claiming Foreign Tax Credits and other Tax Planning Strategies according to the US Internal Revenue Code and Indian Income Tax Act, 1961.

First Name	
Middle Name	
Last Name	
Phone	
UserId	
Password	
Confirm Password	



http://localhost:4214/Tax%20Project/MyProfile.aspx

TAXID: TAX004

First Name

Middle Name

Last Name

Gender ○ Male ○ Female

Date of Birth

SSN

Address

City

State

Zip

Email

Occuiption: Programming Analysist, Business Analysist, System Analysist, Software Engineer

Passport Number

Passport Expiry Date

Visa Number

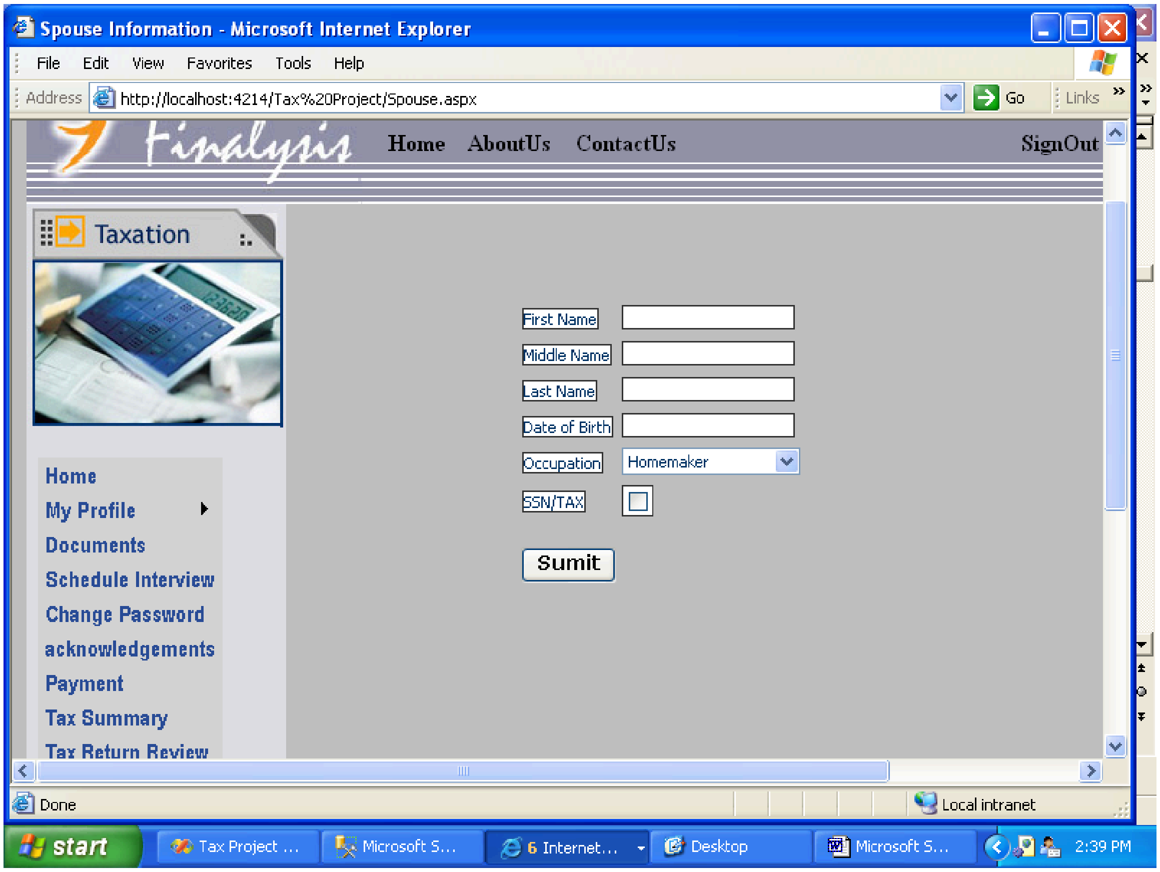
Visa Expiry Date

Marital Status ○ Single ○ Marrried

Dependency ○ Yes ○ No

Bank Information ○ Yes ○ No

Vehicle Information ○ Yes ○ No



First Name

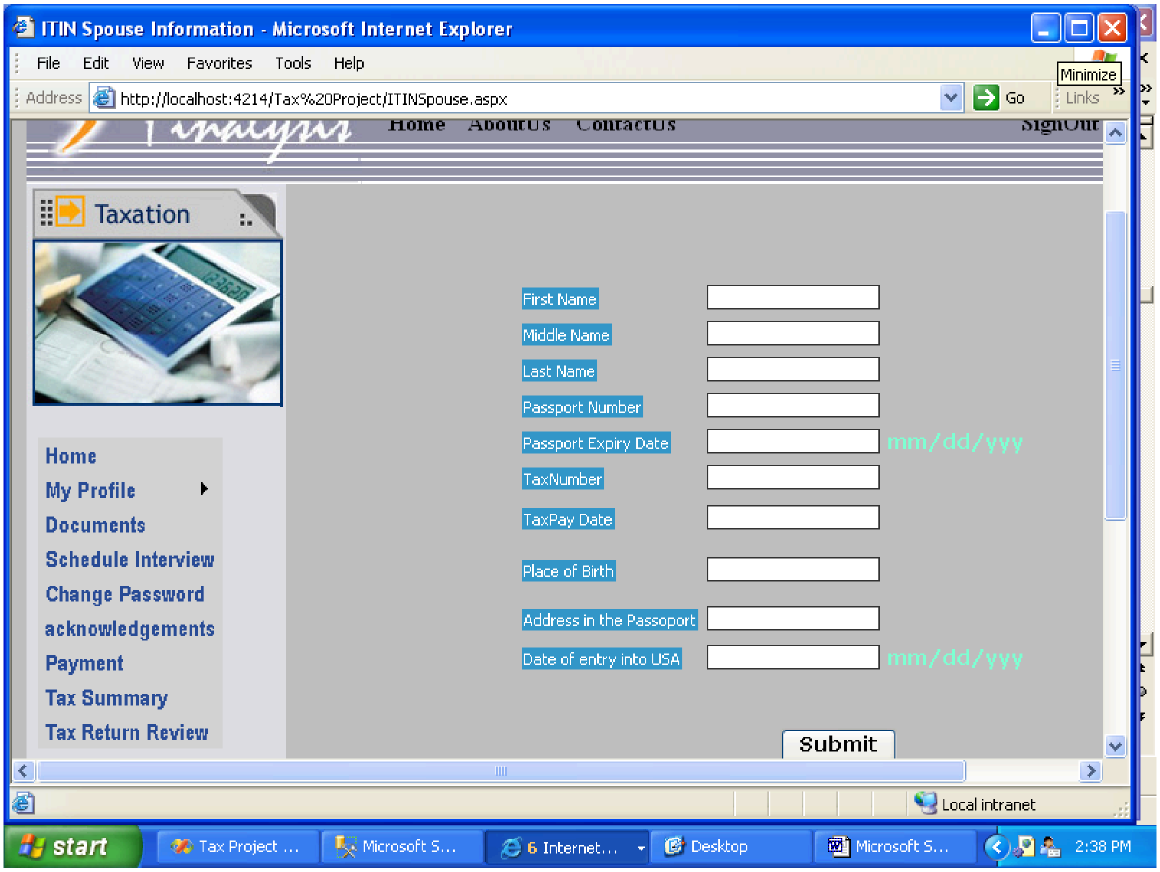
Middle Name

Last Name

Date of Birth

Occupation Homemaker

SSN/TAX □



ITIN Spouse Information - Microsoft Internet Explorer

Address http://localhost:4214/Tax%20Project/ITINSpouse.aspx

First Name

Middle Name

Last Name

Passport Number

Passport Expiry Date mm/dd/yyy

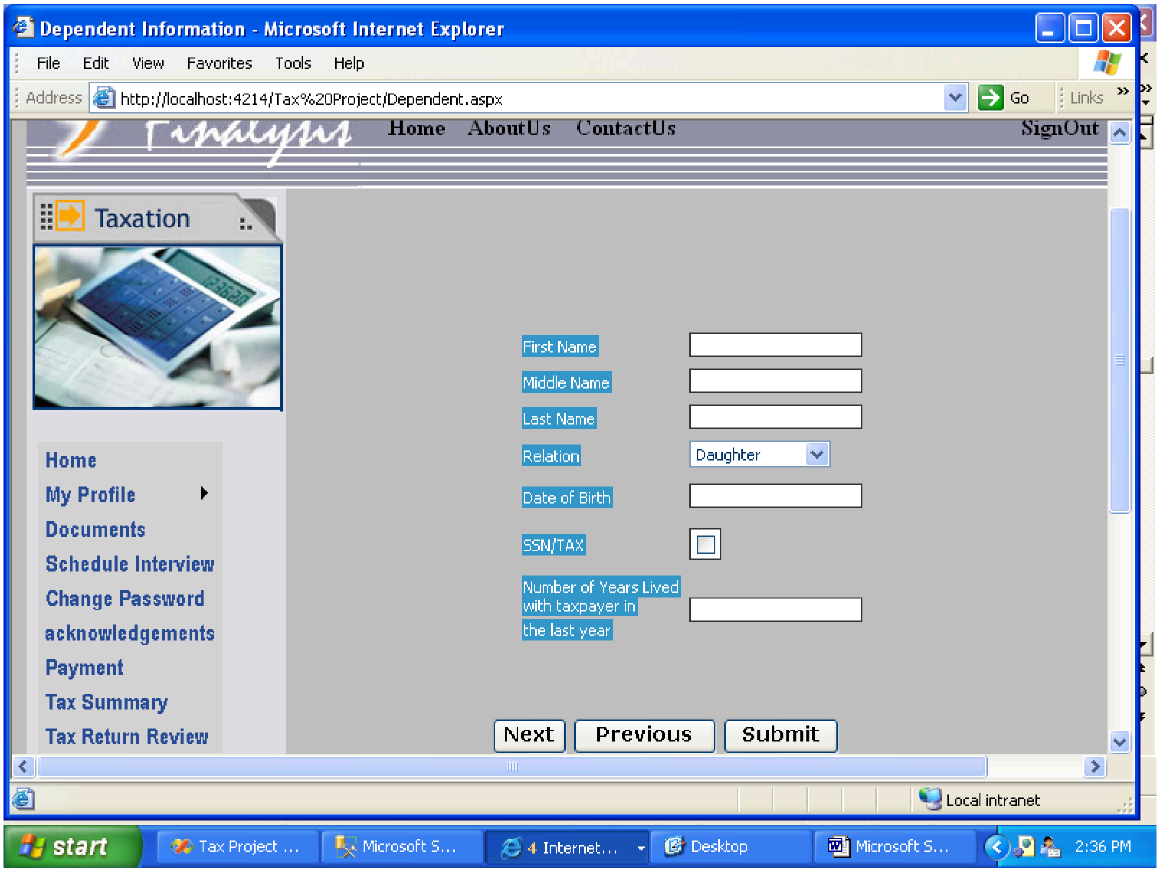
TaxNumber

TaxPay Date

Place of Birth

Address in the Passoport

Date of entry into USA mm/dd/yyy



First Name

Middle Name

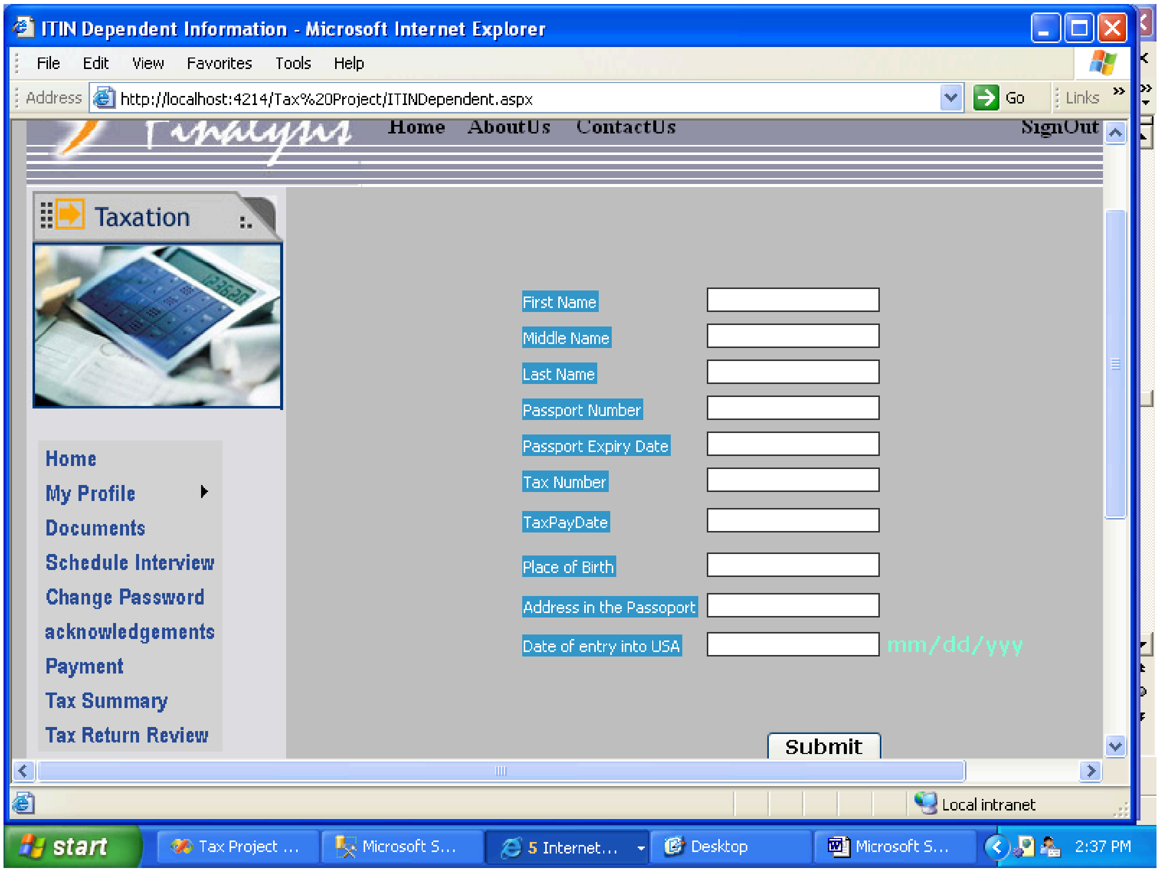
Last Name

Relation Daughter

Date of Birth

SSN/TAX □

Number of Years Lived with taxpayer in the last year



First Name

Middle Name

Last Name

Passport Number

Passport Expiry Date

Tax Number

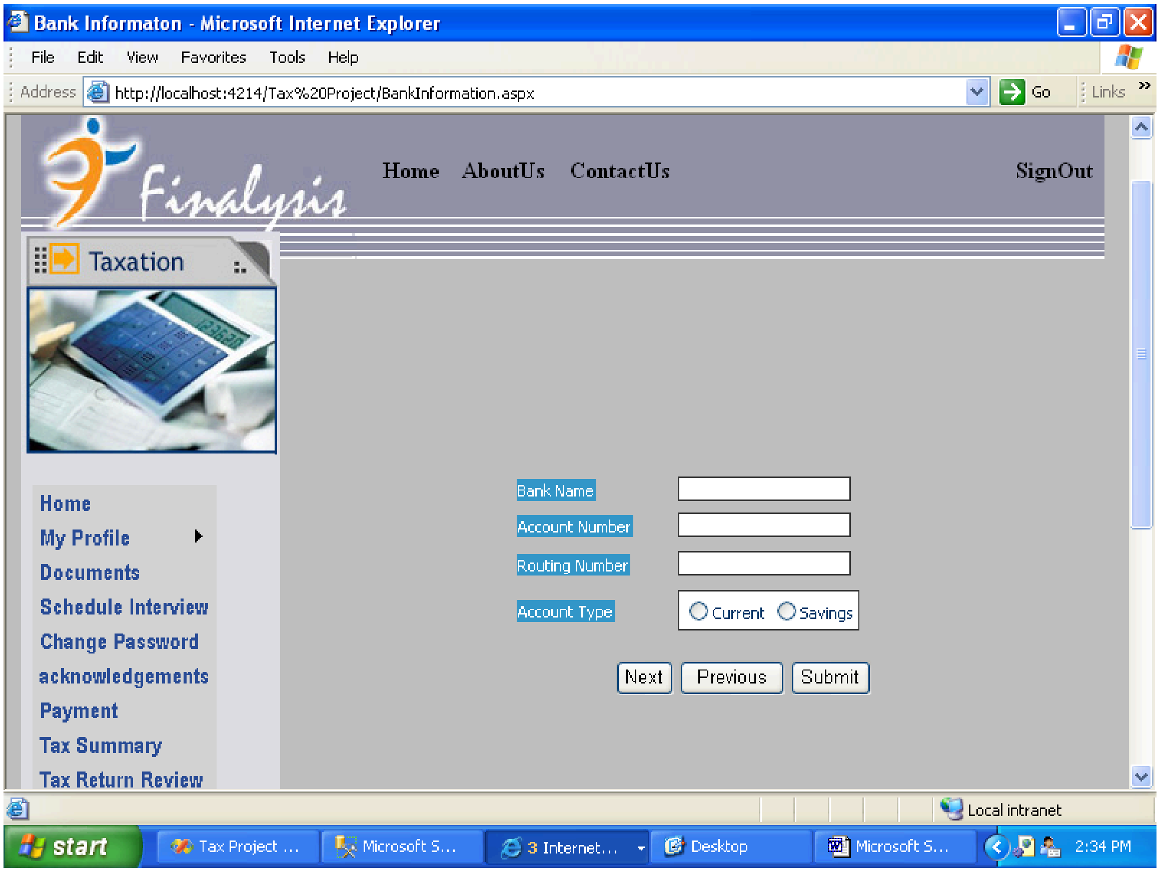
TaxPayDate

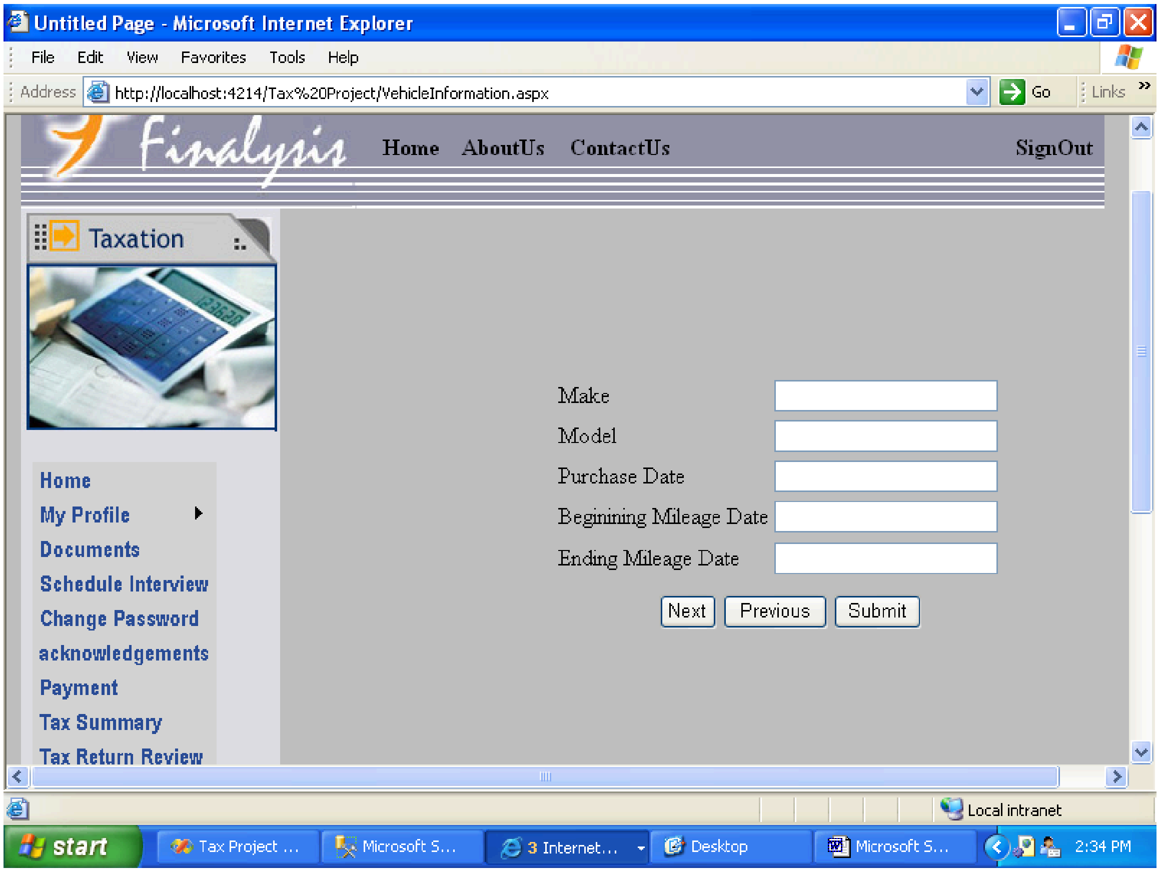
Place of Birth

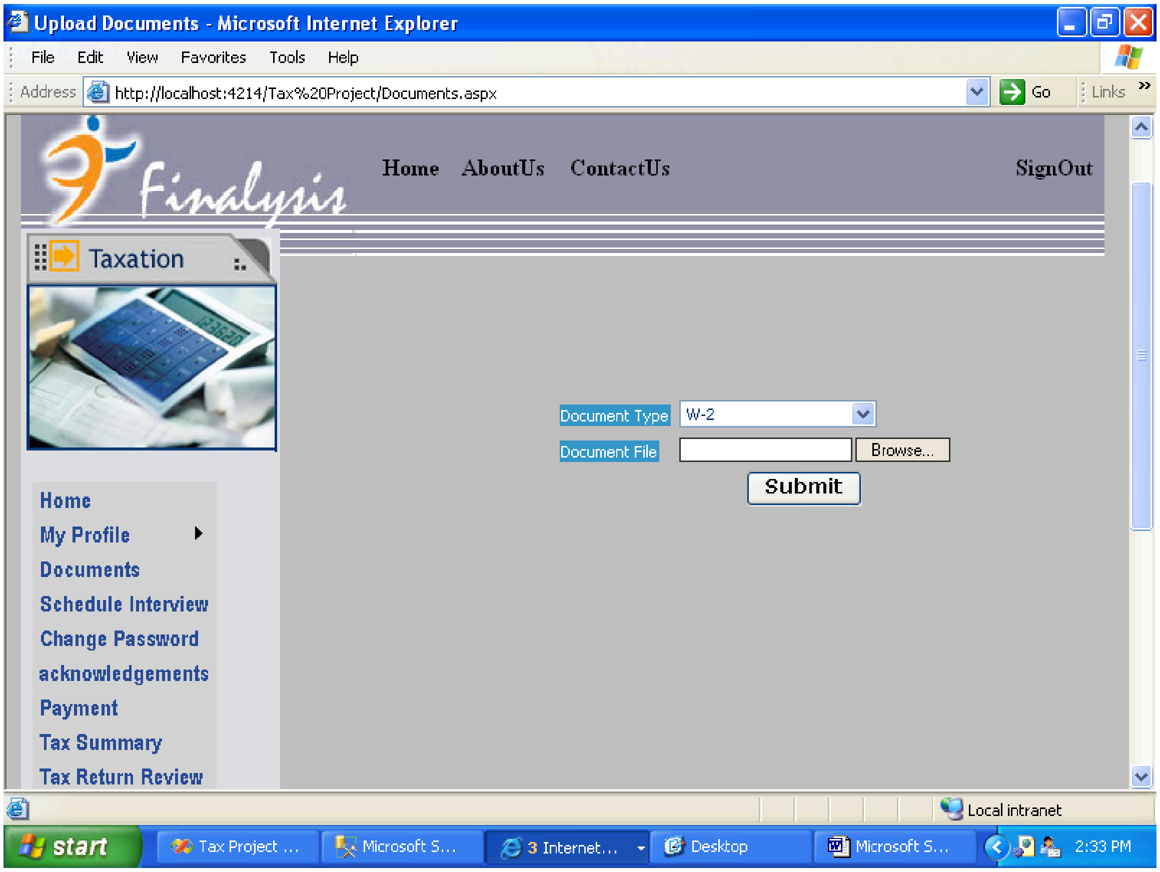
Address in the Passoport

Date of entry into USA mm/dd/yyy

Submit

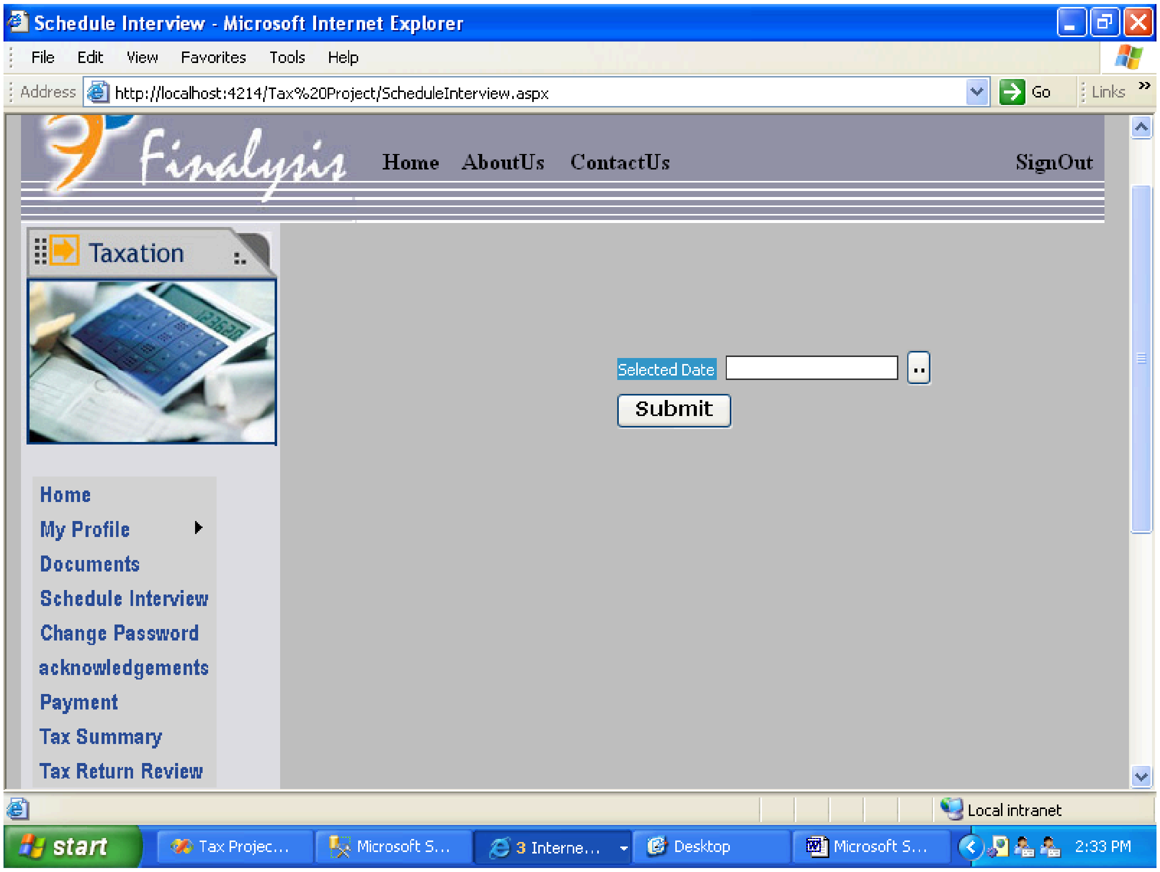

Bank Informaton - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:4214/Tax%20Project/BankInformation.aspx
Go
Links
Finalysis
Home
AboutUs
ContactUs
SignOut
Taxation
Home
My Profile
Documents
Schedule Interview
Change Password
acknowledgements
Payment
Tax Summary
Tax Return Review
Bank Name
Account Number
Routing Number
Account Type
○ Current ○ Savings
Next
Previous
Submit
Local intranet
start
Tax Project ...
Microsoft S...
3 Internet...
Desktop
Microsoft S...
2:34 PM

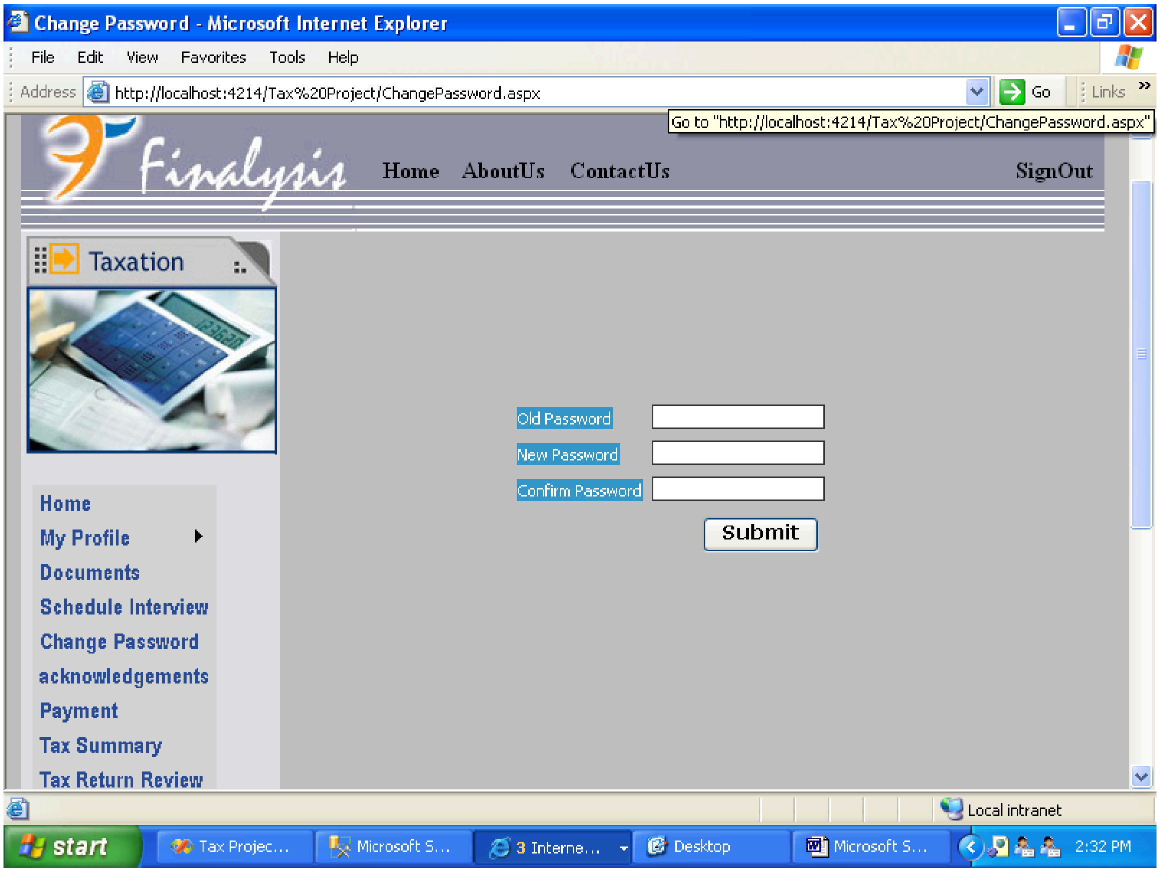




Document Type W-2

Document File


Schedule Interview - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://localhost:4214/Tax%20Project/ScheduleInterview.aspx
Go
Links
Finalysis
Home
AboutUs
ContactUs
SignOut
Taxation
Home
My Profile
Documents
Schedule Interview
Change Password
acknowledgements
Payment
Tax Summary
Tax Return Review
Selected Date
..
Submit
Local intranet
start
Tax Projec...
Microsoft S...
3 Interne...
Desktop
Microsoft S...
2:33 PM



Change Password - Microsoft Internet Explorer

Address http://localhost:4214/Tax%20Project/ChangePassword.aspx

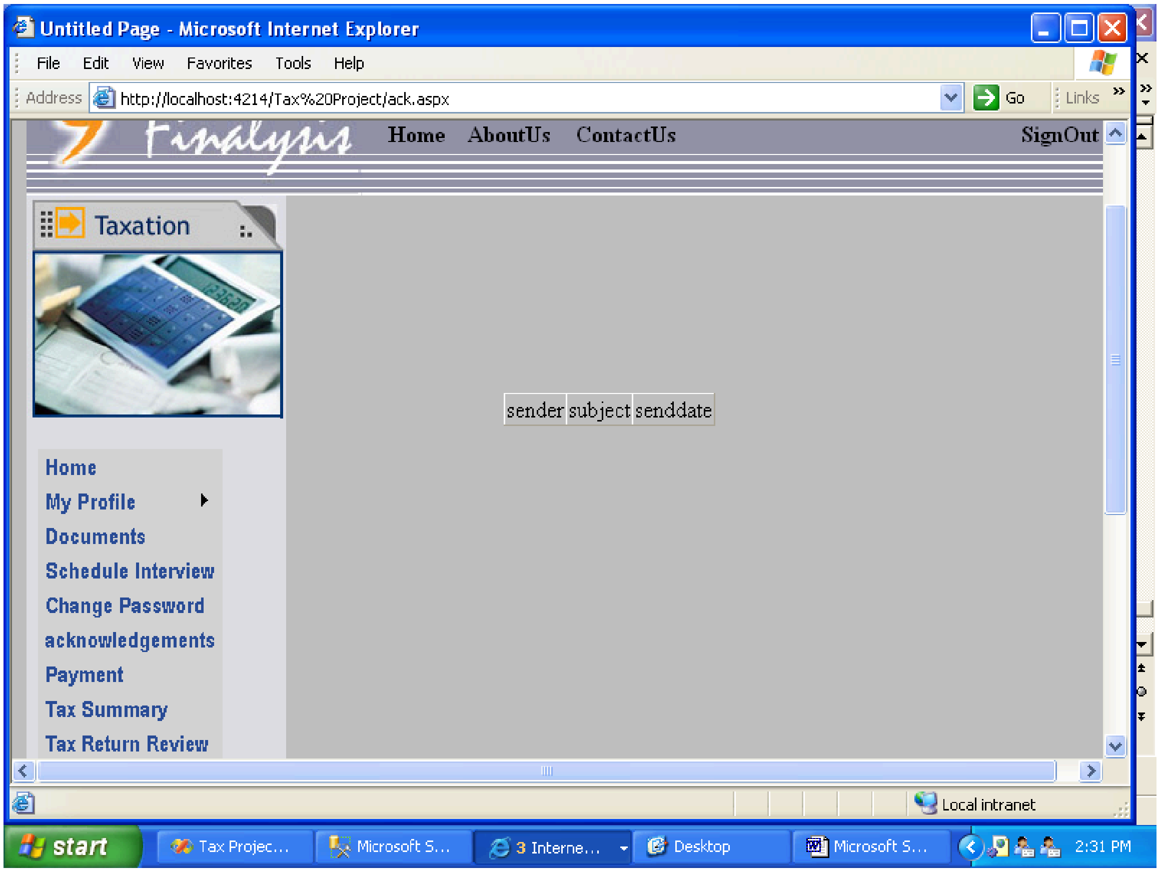
Go to "http://localhost:4214/Tax%20Project/ChangePassword.aspx"

Old Password

New Password

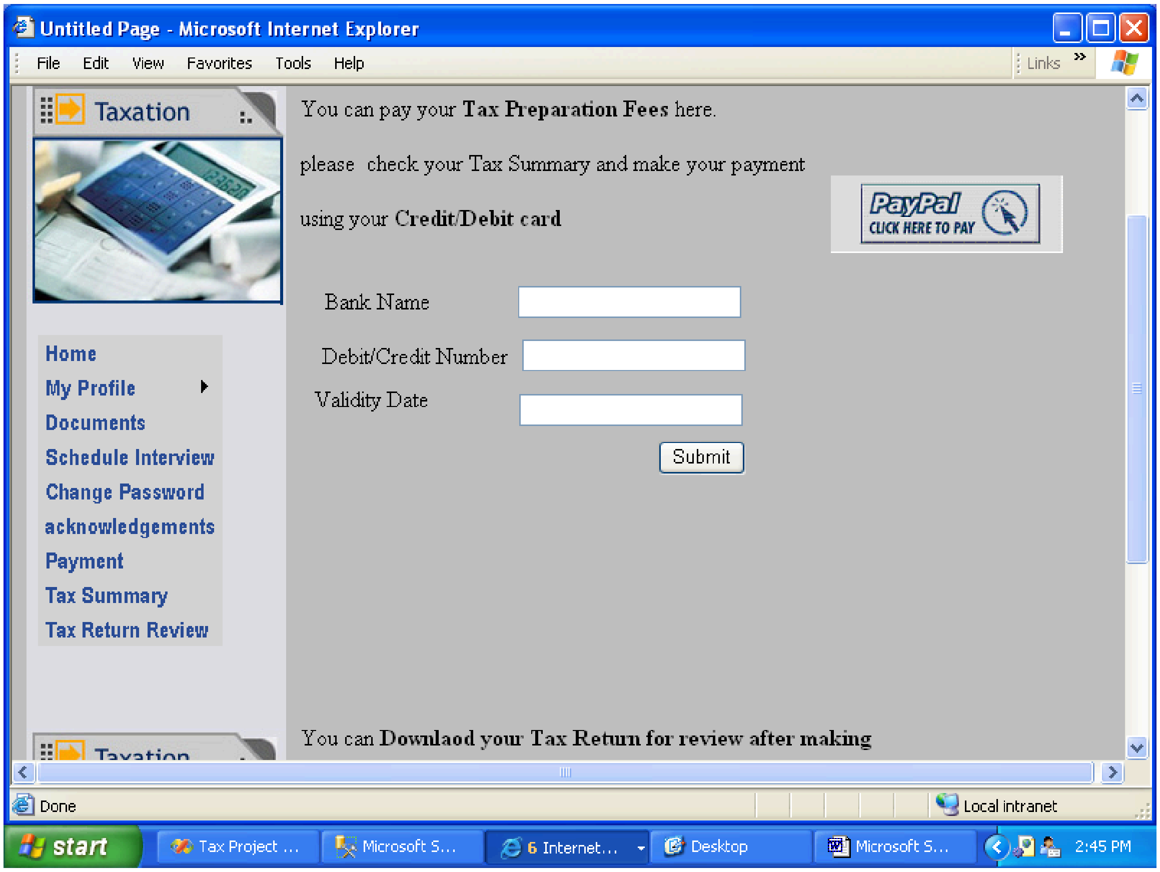
Confirm Password

Submit



Address http://localhost:4214/Tax%20Project/ack.aspx

sender	subject	senddate



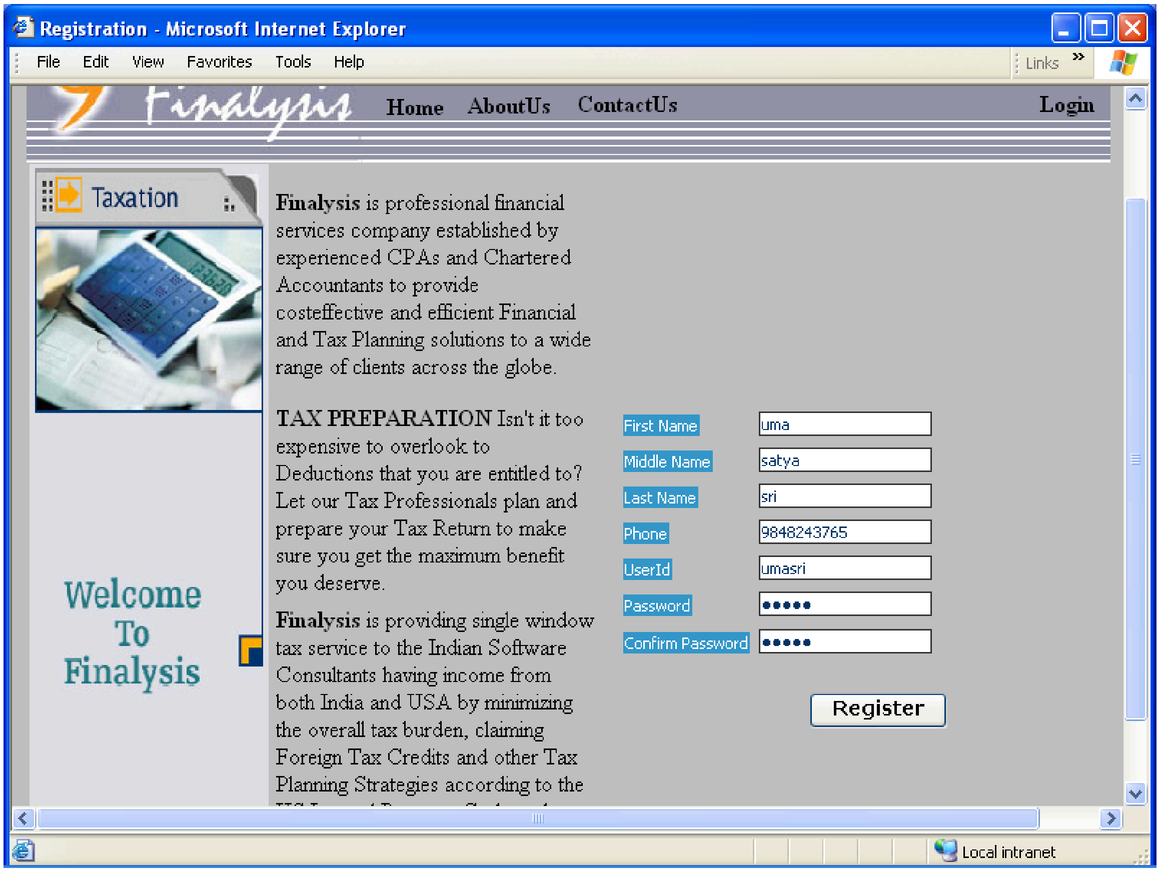
You can pay your **Tax Preparation Fees** here.

please check your Tax Summary and make your payment

using your **Credit/Debit card**

Bank Name	
Debit/Credit Number	
Validity Date	

You can **Downlaod your Tax Return for review after making**

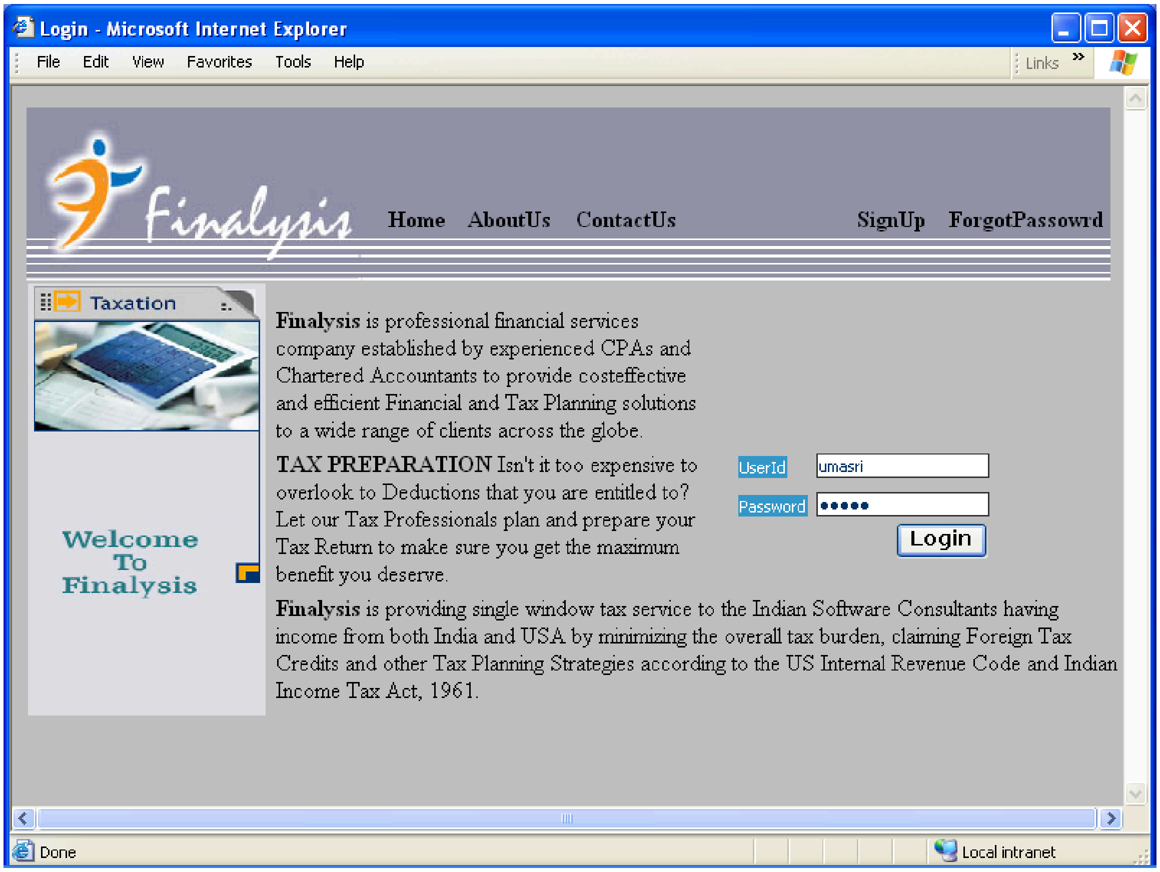


Finalysis is professional financial services company established by experienced CPAs and Chartered Accountants to provide costeffective and efficient Financial and Tax Planning solutions to a wide range of clients across the globe.

TAX PREPARATION Isn't it too expensive to overlook to Deductions that you are entitled to? Let our Tax Professionals plan and prepare your Tax Return to make sure you get the maximum benefit you deserve.

Finalysis is providing single window tax service to the Indian Software Consultants having income from both India and USA by minimizing the overall tax burden, claiming Foreign Tax Credits and other Tax Planning Strategies according to the

First Name	uma
Middle Name	satya
Last Name	sri
Phone	9848243765
UserId	umasri
Password	●●●●●
Confirm Password	●●●●●



Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Finalysis
Home AboutUs ContactUs SignUp ForgotPassowrd
Taxation
Welcome To Finalysis
Done
Local intranet

Finalysis is professional financial services company established by experienced CPAs and Chartered Accountants to provide costeffective and efficient Financial and Tax Planning solutions to a wide range of clients across the globe.

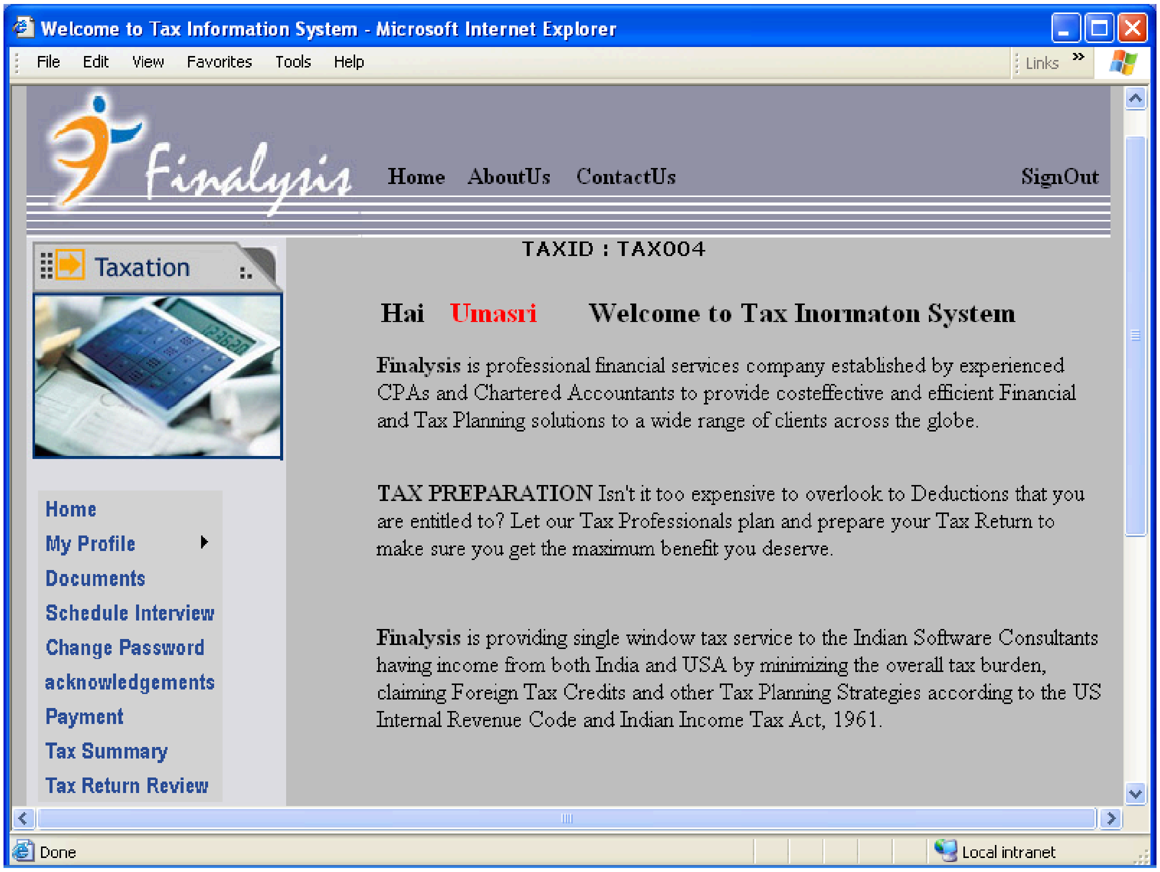
TAX PREPARATION Isn't it too expensive to overlook to Deductions that you are entitled to? Let our Tax Professionals plan and prepare your Tax Return to make sure you get the maximum benefit you deserve.

Finalysis is providing single window tax service to the Indian Software Consultants having income from both India and USA by minimizing the overall tax burden, claiming Foreign Tax Credits and other Tax Planning Strategies according to the US Internal Revenue Code and Indian Income Tax Act, 1961.

UserId umasri

Password •••••

Login



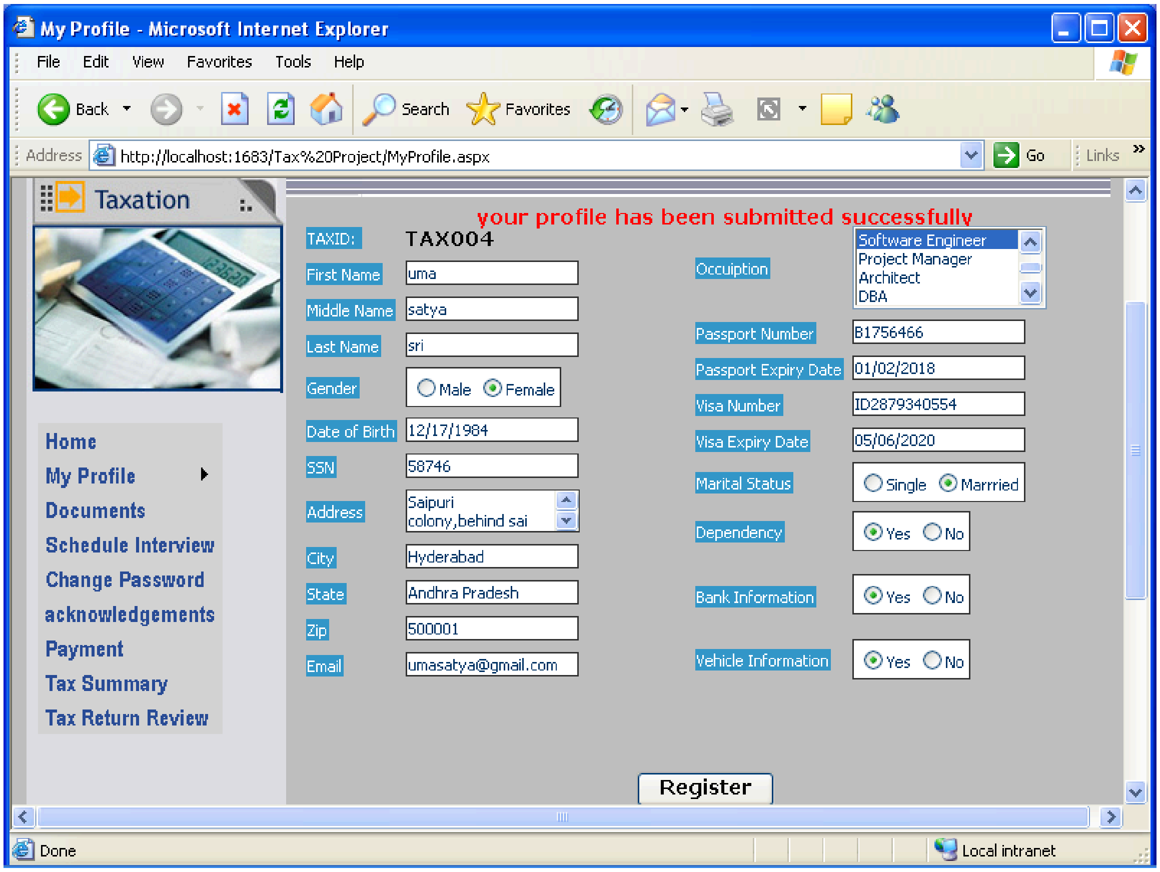
TAXID : TAX004

Hai Umasri Welcome to Tax Inormaton System

Finalysis is professional financial services company established by experienced CPAs and Chartered Accountants to provide costeffective and efficient Financial and Tax Planning solutions to a wide range of clients across the globe.

TAX PREPARATION Isn't it too expensive to overlook to Deductions that you are entitled to? Let our Tax Professionals plan and prepare your Tax Return to make sure you get the maximum benefit you deserve.

Finalysis is providing single window tax service to the Indian Software Consultants having income from both India and USA by minimizing the overall tax burden, claiming Foreign Tax Credits and other Tax Planning Strategies according to the US Internal Revenue Code and Indian Income Tax Act, 1961.



your profile has been submitted successfully

TAXID: TAX004

First Name: uma

Middle Name: satya

Last Name: sri

Gender: ○ Male ● Female

Date of Birth: 12/17/1984

SSN: 58746

Address: Saipuri colony,behind sai

City: Hyderabad

State: Andhra Pradesh

Zip: 500001

Email: umasatya@gmail.com

Occuption: [✓ Software Engineer] Project Manager Architect DBA

Passport Number: B1756466

Passport Expiry Date: 01/02/2018

Visa Number: ID2879340554

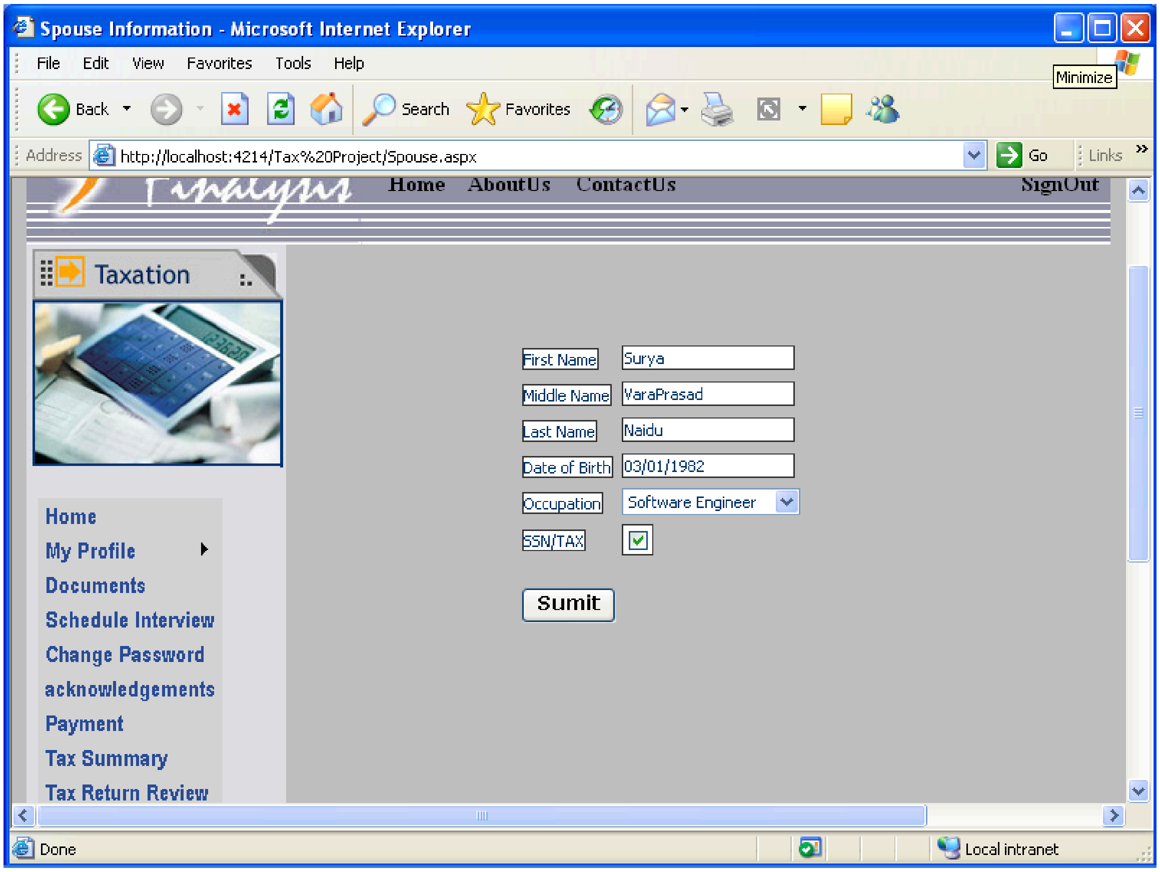
Visa Expiry Date: 05/06/2020

Marital Status: ○ Single ● Marrried

Dependency: ● Yes ○ No

Bank Information: ● Yes ○ No

Vehicle Information: ● Yes ○ No



First Name: Surya

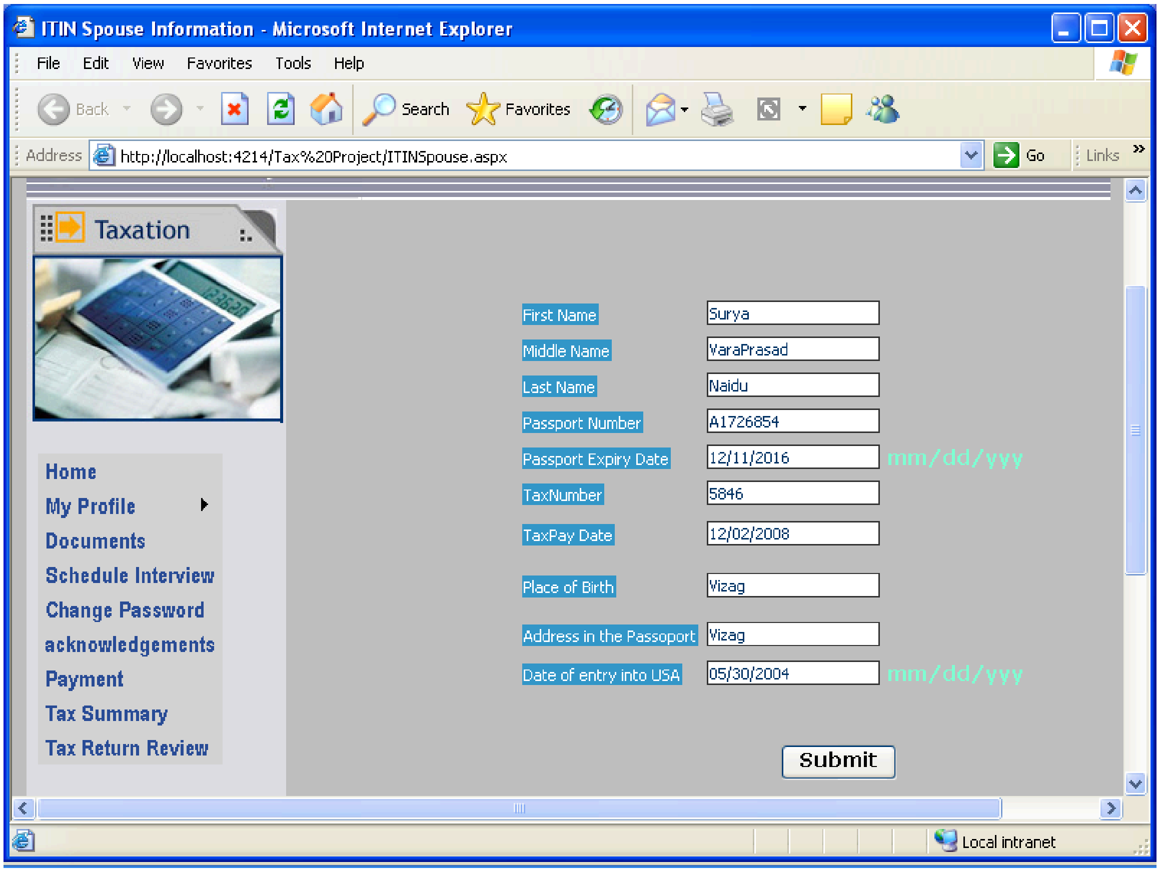
Middle Name: VaraPrasad

Last Name: Naidu

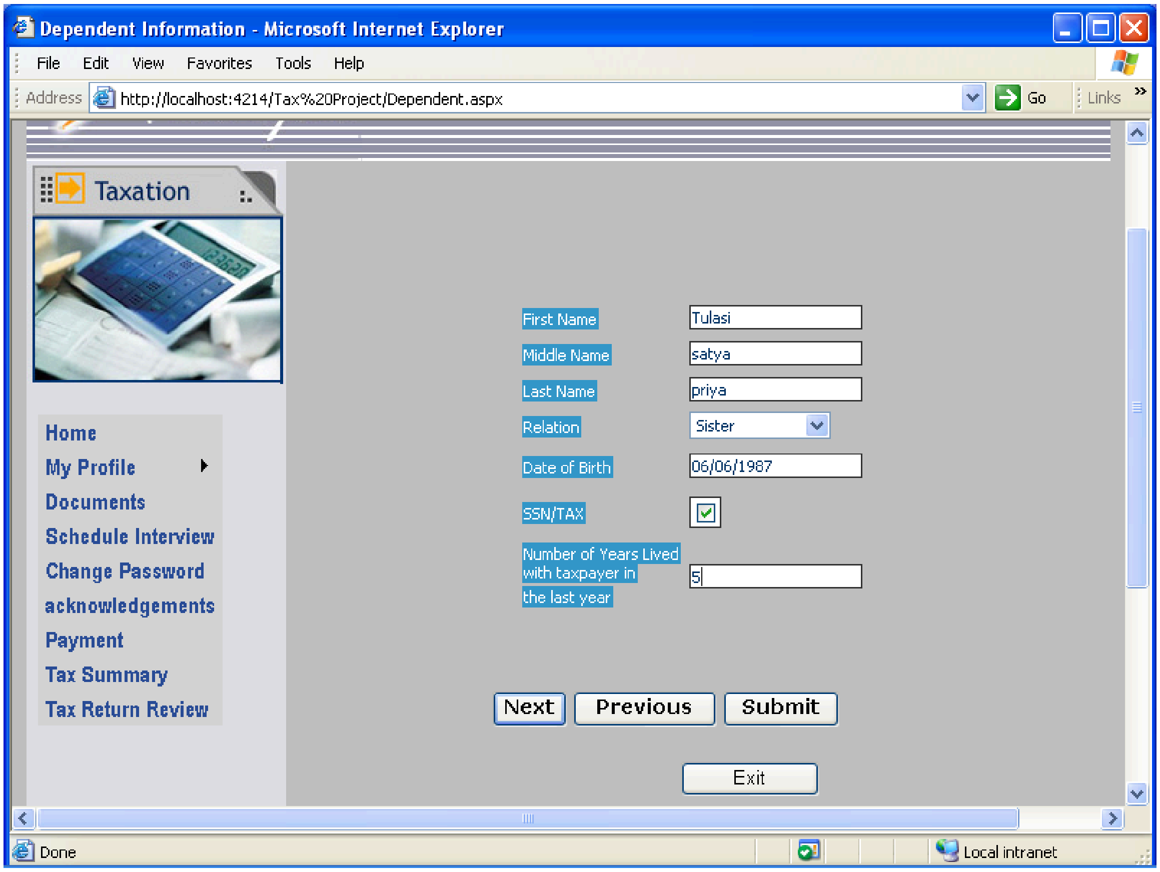
Date of Birth: 03/01/1982

Occupation: Software Engineer

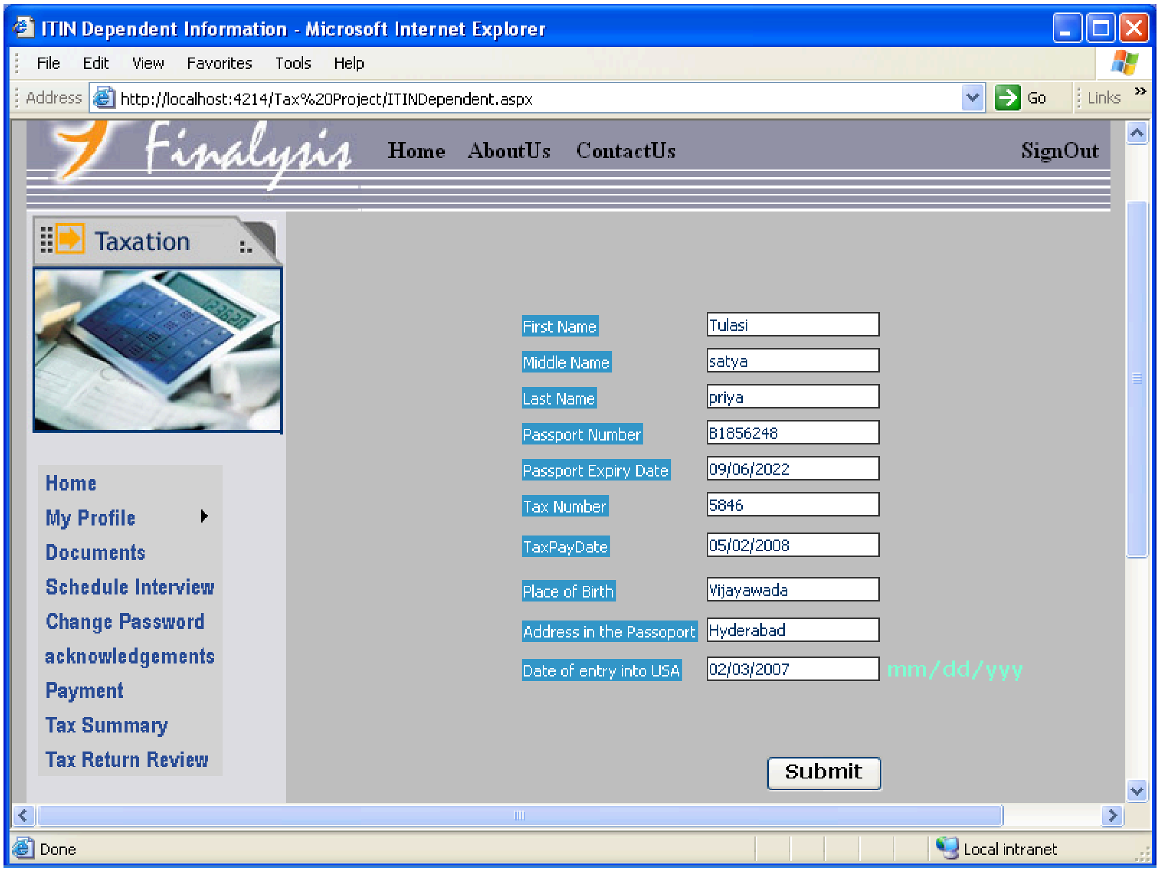
SSN/TAX ☑



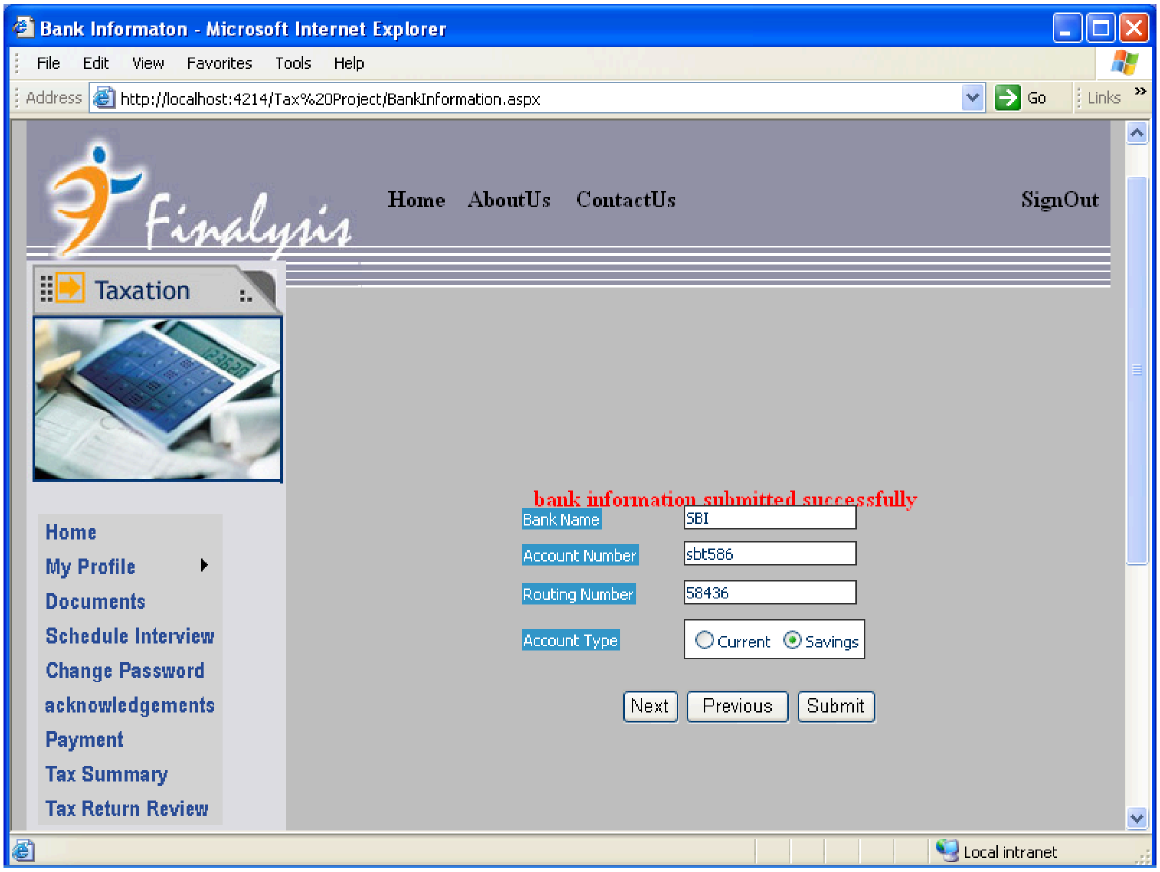
First Name	Surya	
Middle Name	VaraPrasad	
Last Name	Naidu	
Passport Number	A1726854	
Passport Expiry Date	12/11/2016	mm/dd/yyy
TaxNumber	5846	
TaxPay Date	12/02/2008	
Place of Birth	Vizag	
Address in the Passoport	Vizag	
Date of entry into USA	05/30/2004	mm/dd/yyy



First Name	Tulasi
Middle Name	satya
Last Name	priya
Relation	Sister
Date of Birth	06/06/1987
SSN/TAX	☑
Number of Years Lived with taxpayer in the last year	5

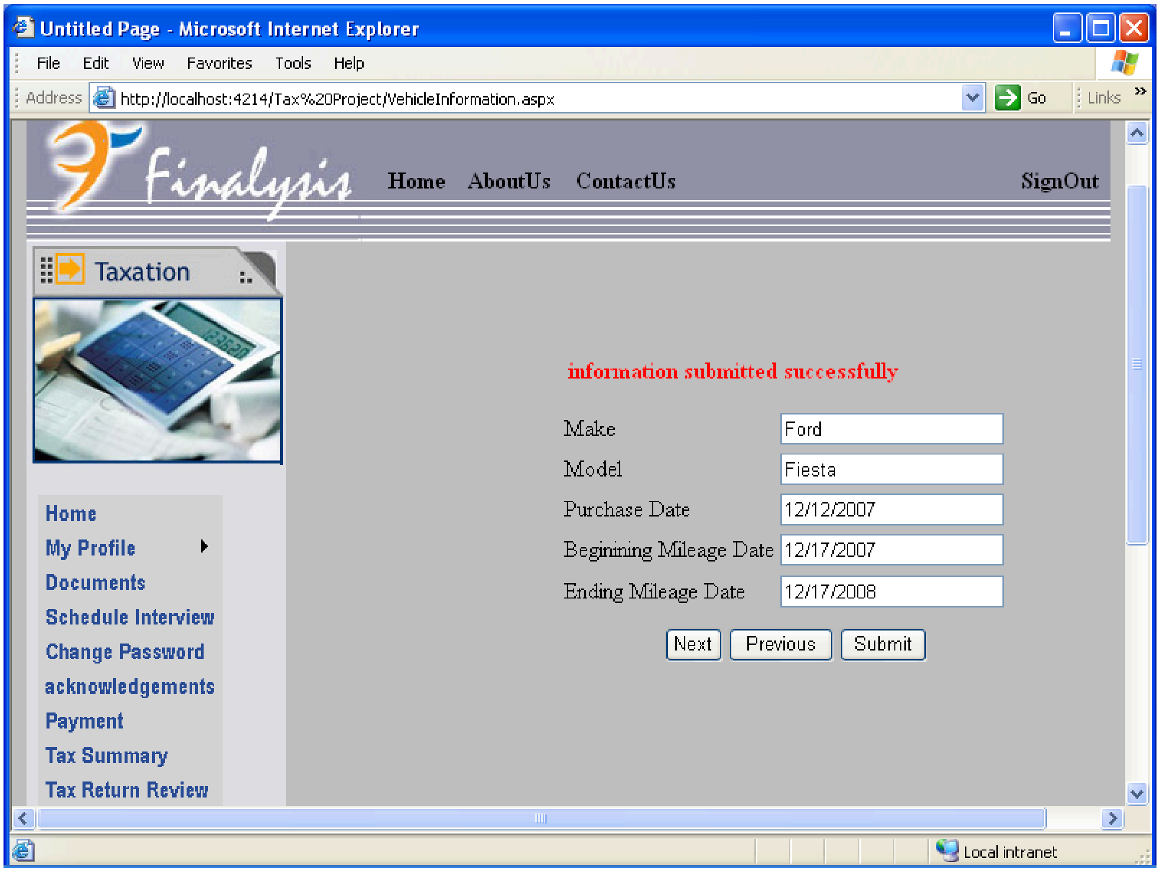


First Name	Tulasi
Middle Name	satya
Last Name	priya
Passport Number	B1856248
Passport Expiry Date	09/06/2022
Tax Number	5846
TaxPayDate	05/02/2008
Place of Birth	Vijayawada
Address in the Passoport	Hyderabad
Date of entry into USA	02/03/2007 mm/dd/yyy



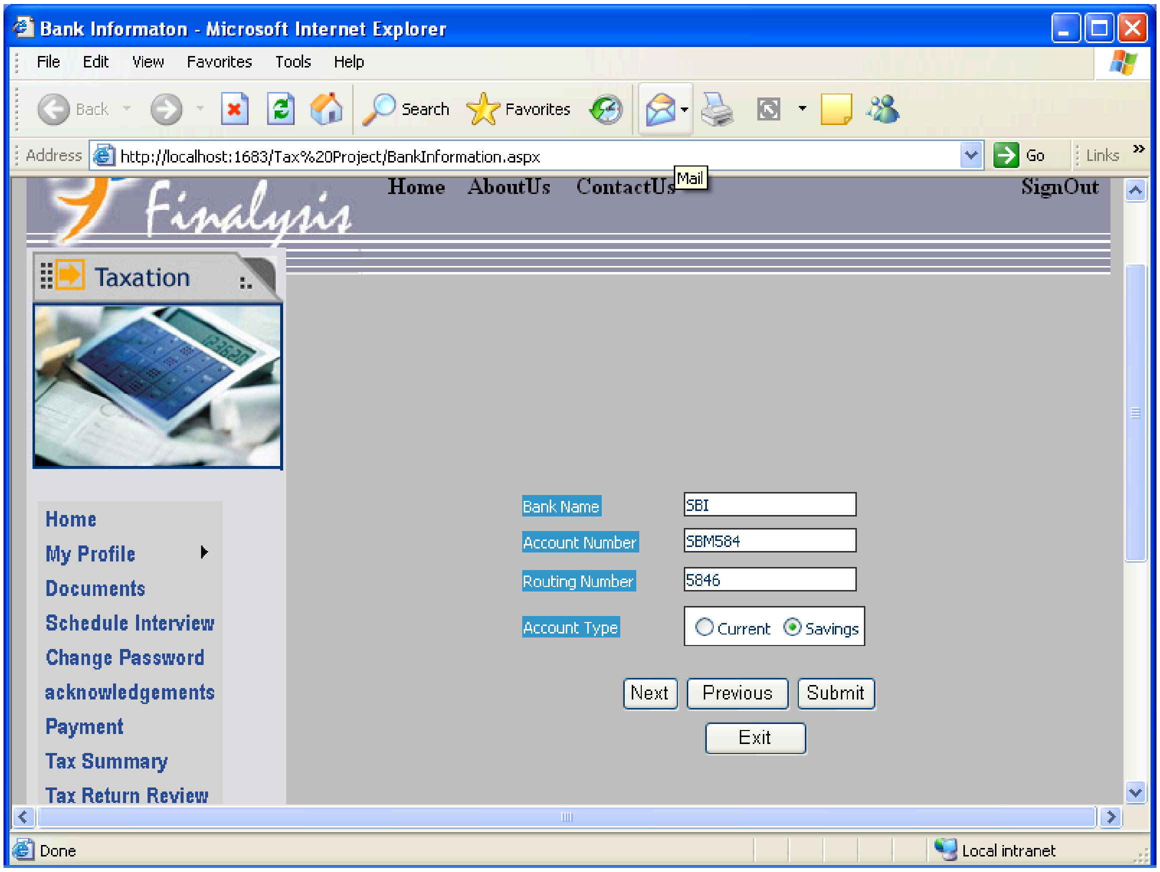
bank information submitted successfully

Bank Name	SBI
Account Number	sbt586
Routing Number	58436
Account Type	○ Current ● Savings



information submitted successfully

Make	Ford
Model	Fiesta
Purchase Date	12/12/2007
Begining Mileage Date	12/17/2007
Ending Mileage Date	12/17/2008

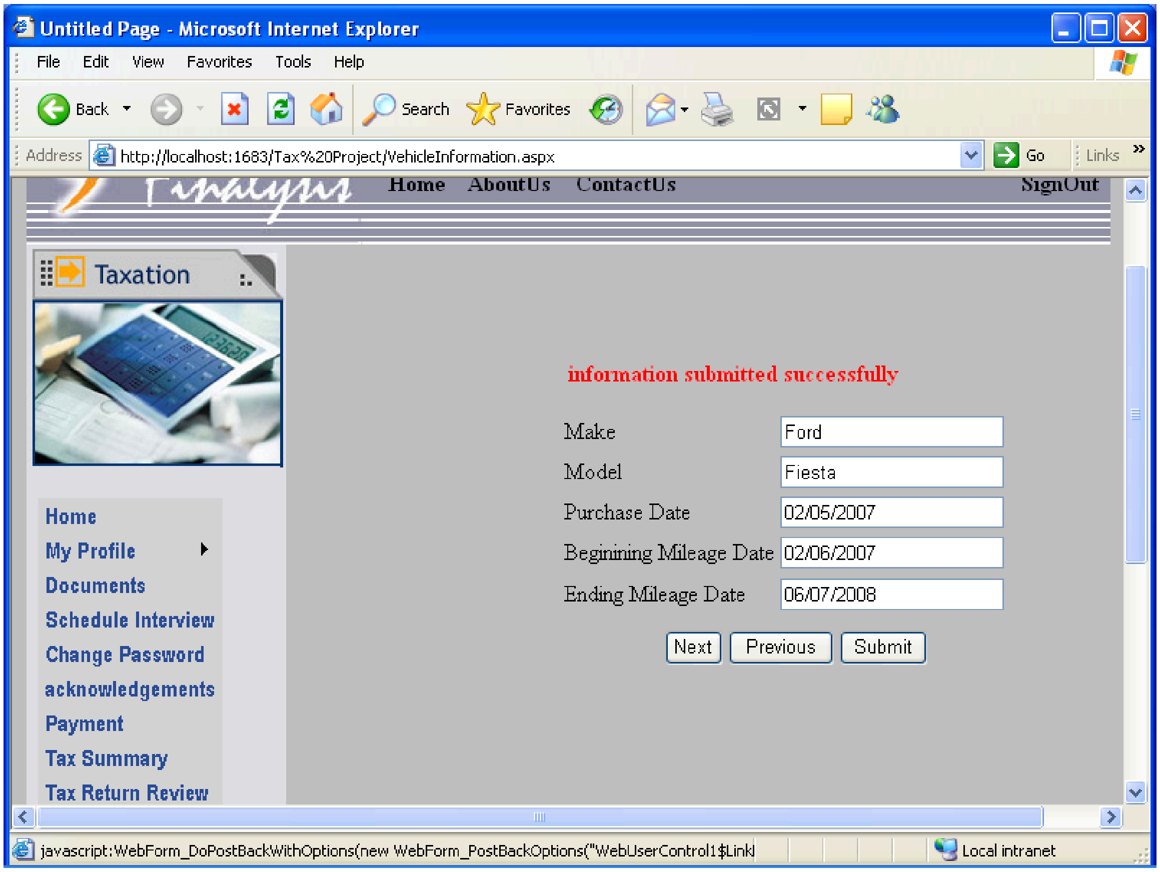


Bank Name: SBI

Account Number: SBM584

Routing Number: 5846

Account Type: ○ Current ● Savings

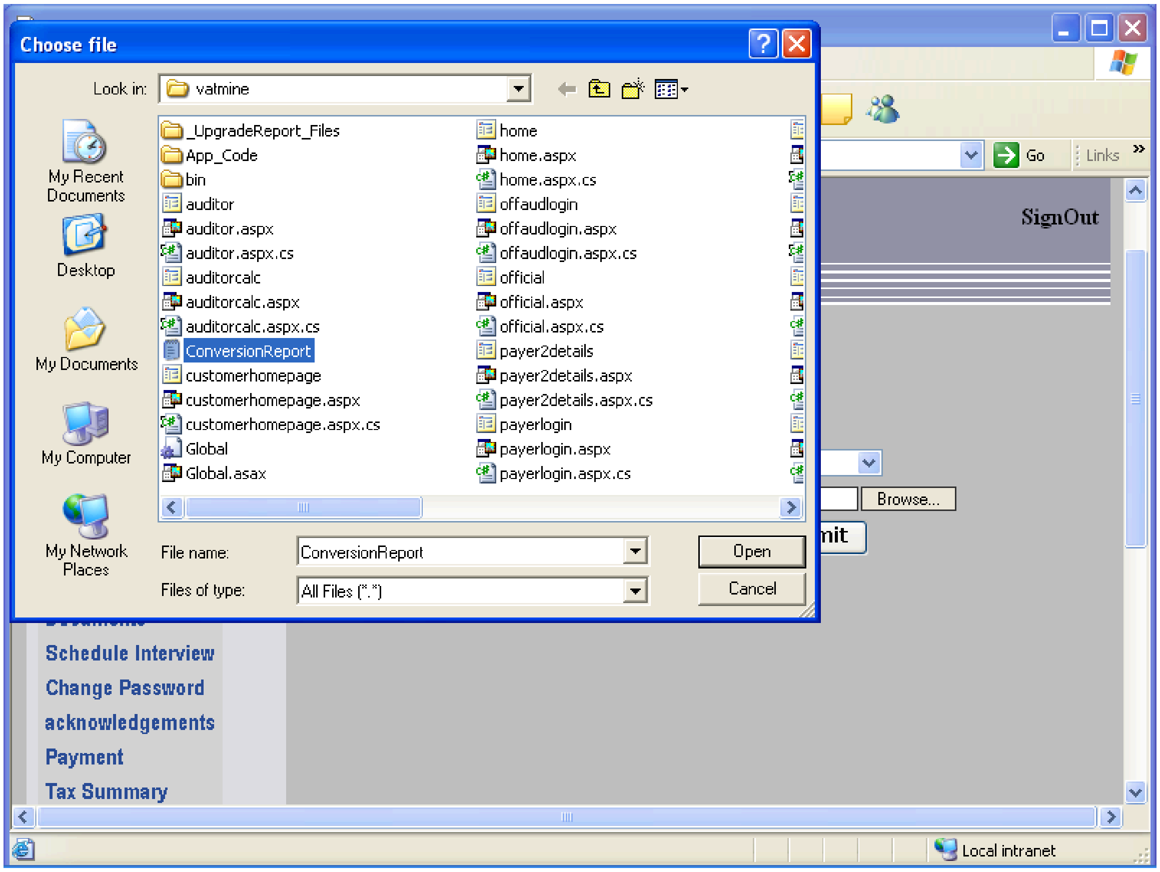


http://localhost:1683/Tax%20Project/VehicleInformation.aspx

information submitted successfully

Make	Ford
Model	Fiesta
Purchase Date	02/05/2007
Beginining Mileage Date	02/06/2007
Ending Mileage Date	06/07/2008

javascript:WebForm_DoPostBackWithOptions(new WebForm_PostBackOptions("WebUserControl1$Link



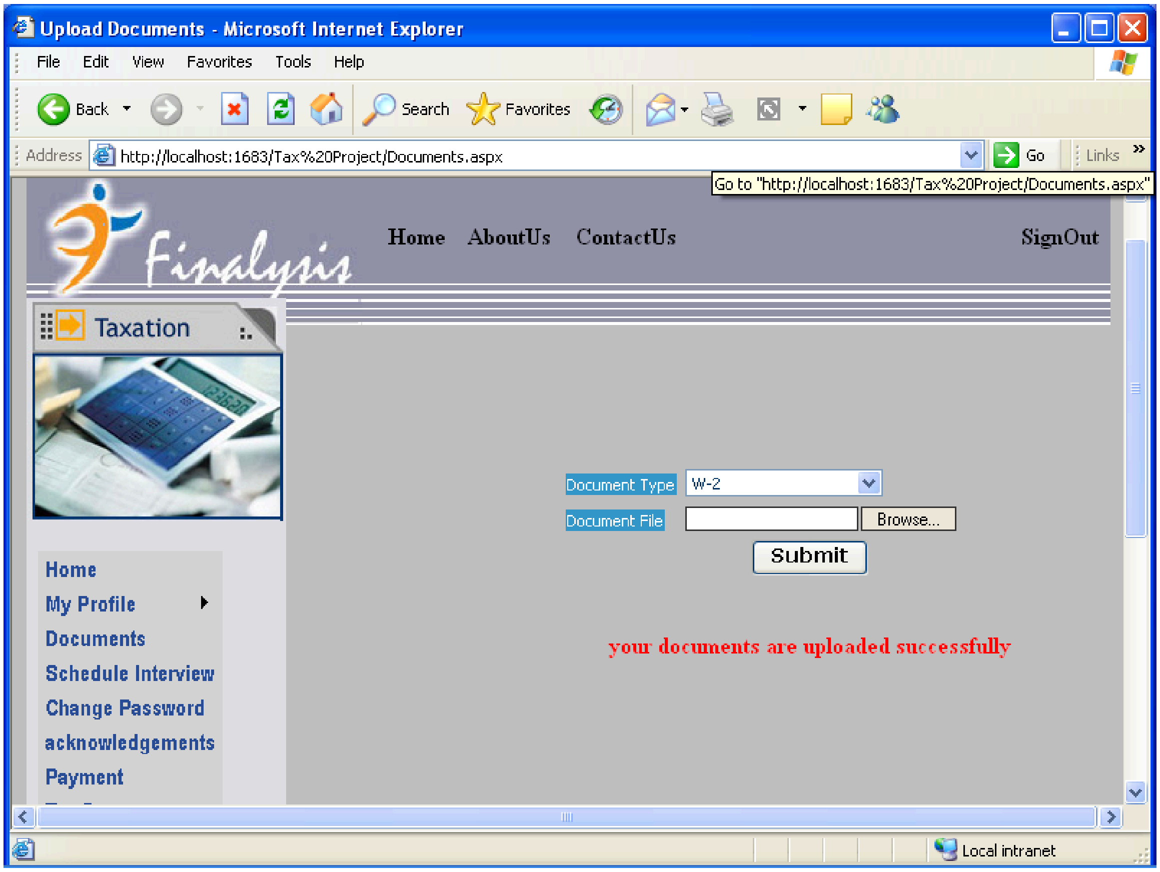
Choose file

Look in: vatmine

_UpgradeReport_Files
App_Code
bin
auditor
auditor.aspx
auditor.aspx.cs
auditorcalc
auditorcalc.aspx
auditorcalc.aspx.cs
ConversionReport
customerhomepage
customerhomepage.aspx
customerhomepage.aspx.cs
Global
Global.asax
home
home.aspx
home.aspx.cs
offaudlogin
offaudlogin.aspx
offaudlogin.aspx.cs
official
official.aspx
official.aspx.cs
payer2details
payer2details.aspx
payer2details.aspx.cs
payerlogin
payerlogin.aspx
payerlogin.aspx.cs

File name: ConversionReport

Files of type: All Files (*.*)

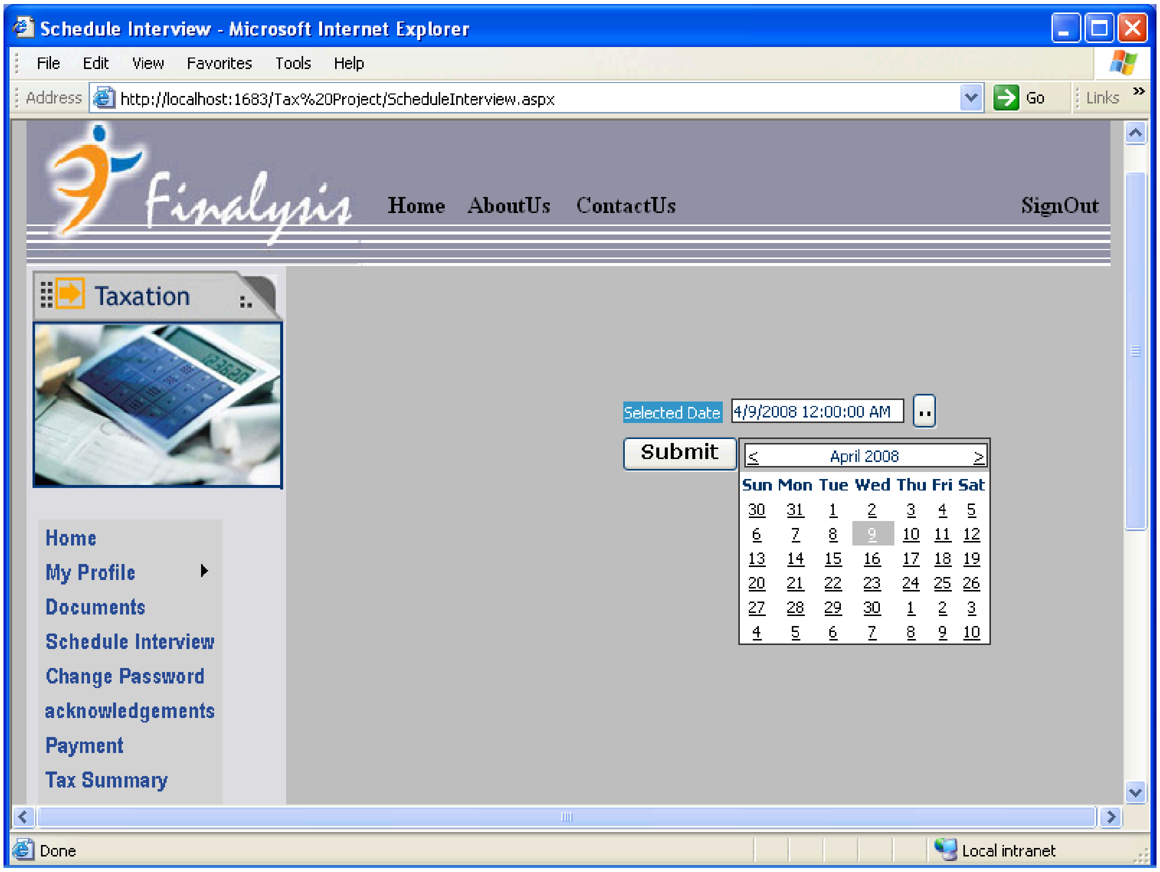


Address: http://localhost:1683/Tax%20Project/Documents.aspx

Document Type: W-2

Document File:

your documents are uploaded successfully

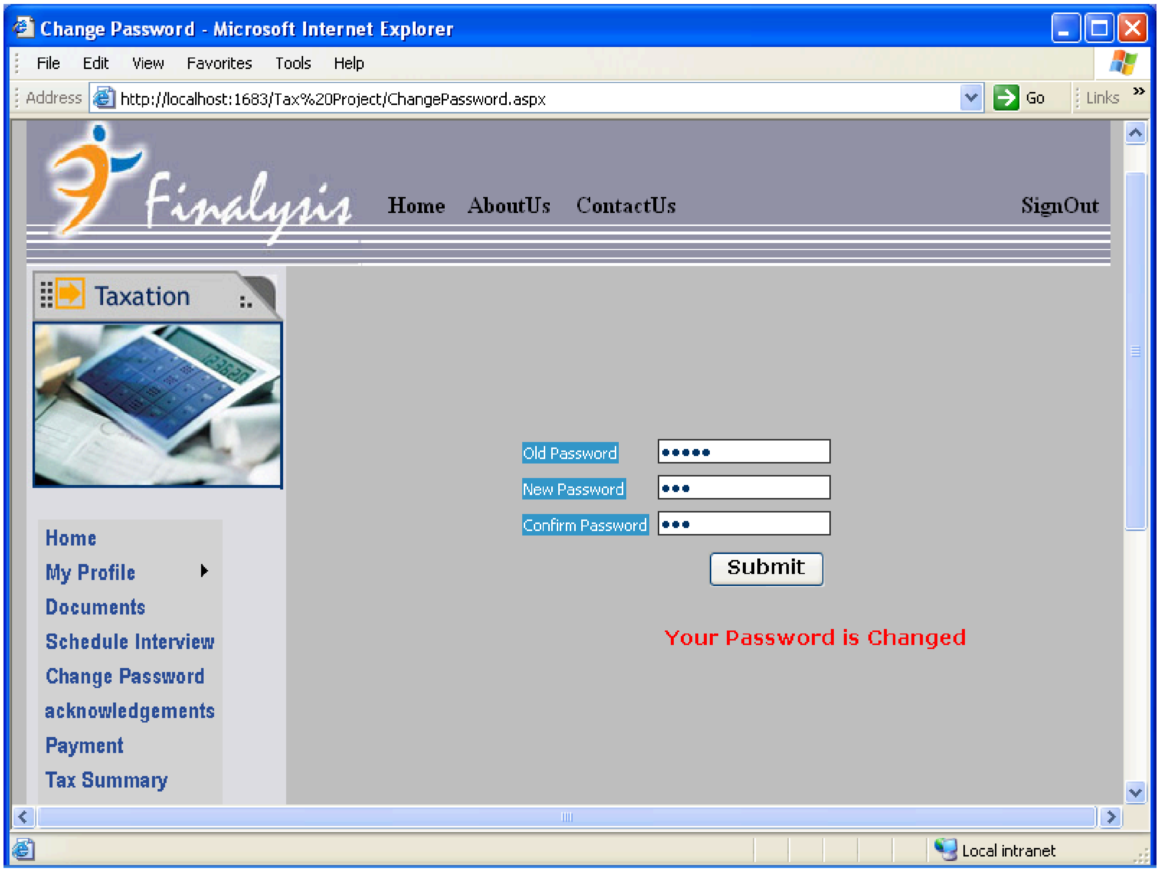


Address http://localhost:1683/Tax%20Project/ScheduleInterview.aspx

Selected Date 4/9/2008 12:00:00 AM

< April 2008 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10



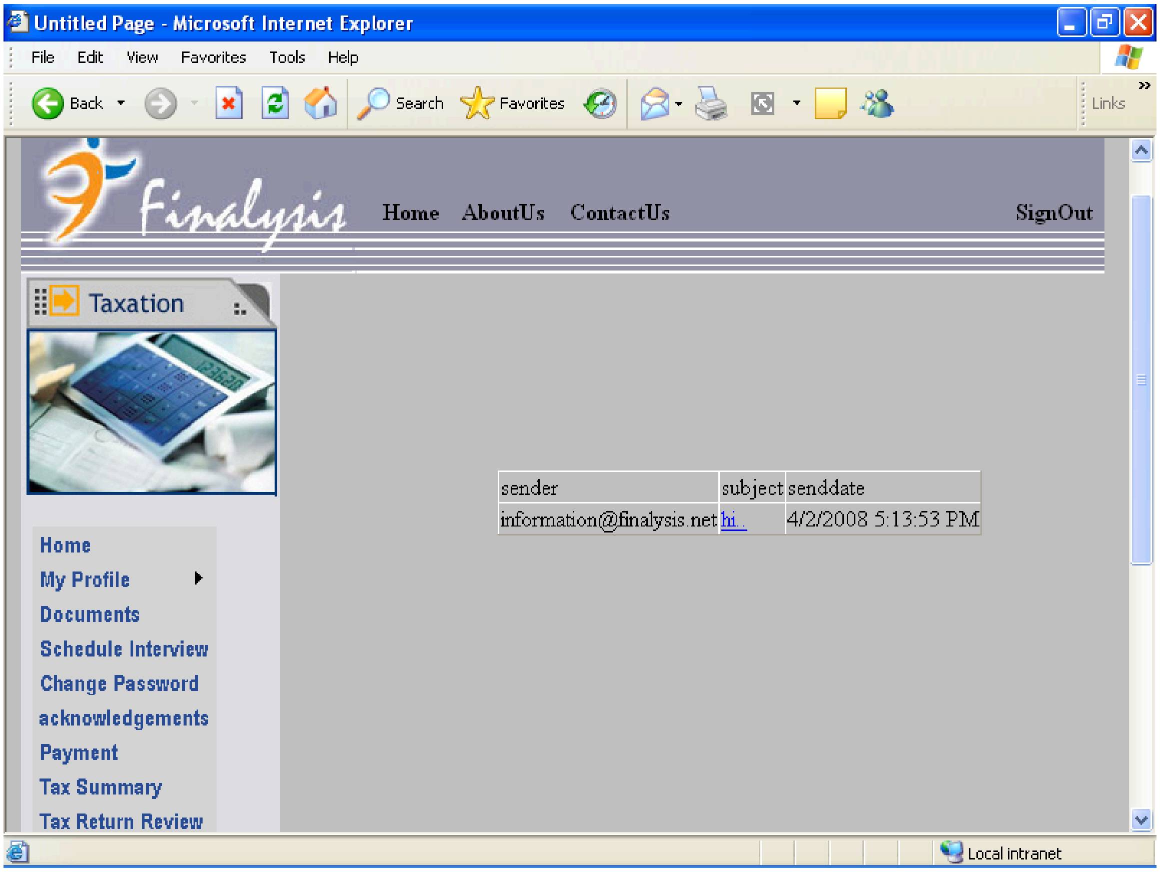
Old Password ●●●●●

New Password ●●●

Confirm Password ●●●

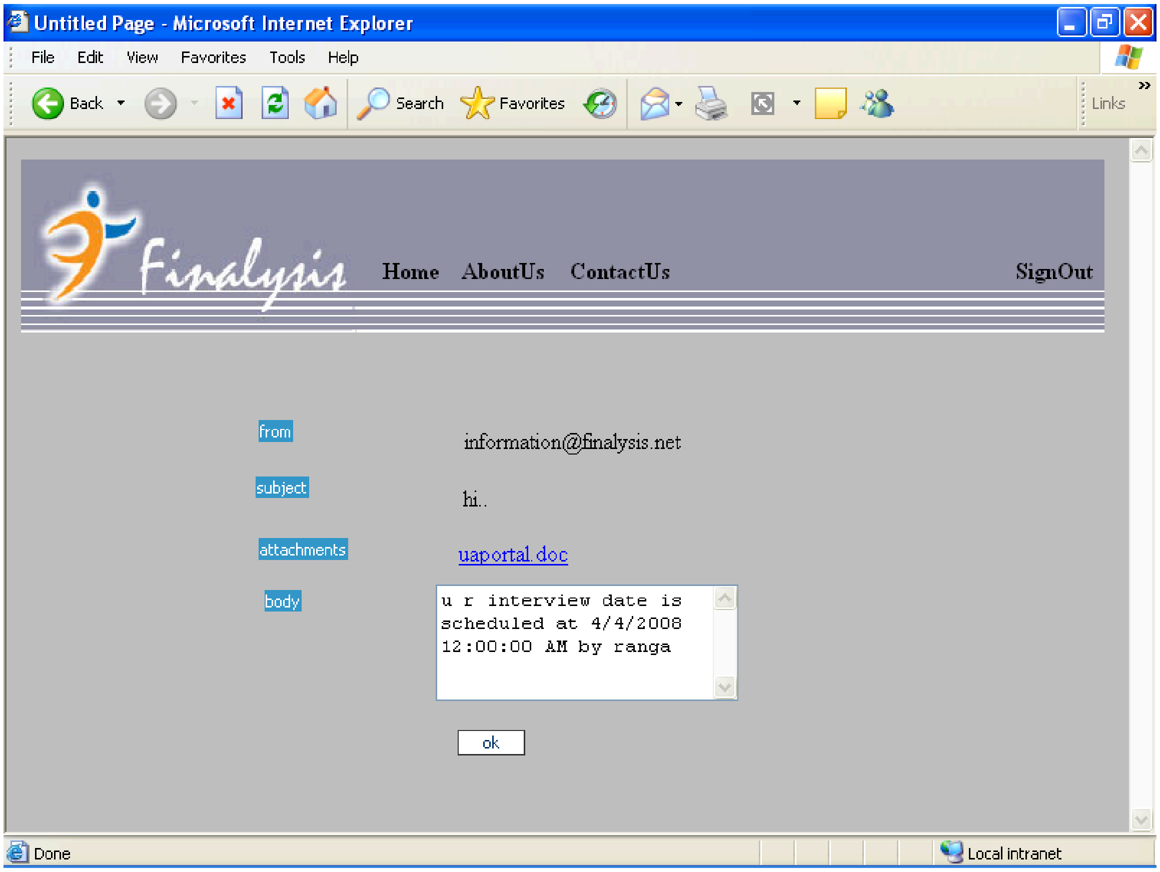
Your Password is Changed

acknowledgement



sender	subject	senddate
information@finalysis.net	hi..	4/2/2008 5:13:53 PM

Details of interview sent to mail


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Links
Finalysis
Home
AboutUs
ContactUs
SignOut
Done
Local intranet

from information@finalysis.net

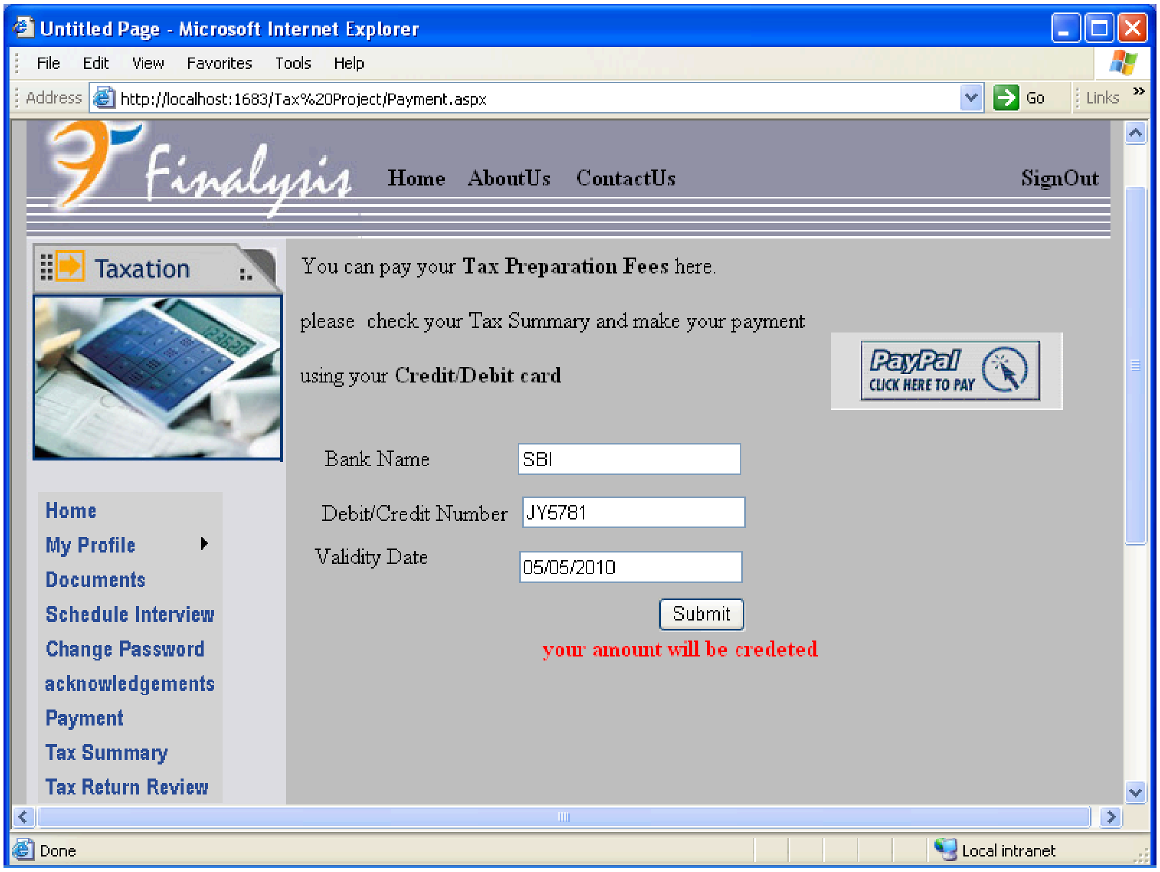
subject hi..

attachments uaportal.doc

body u r interview date is scheduled at 4/4/2008 12:00:00 AM by ranga

ok

payment



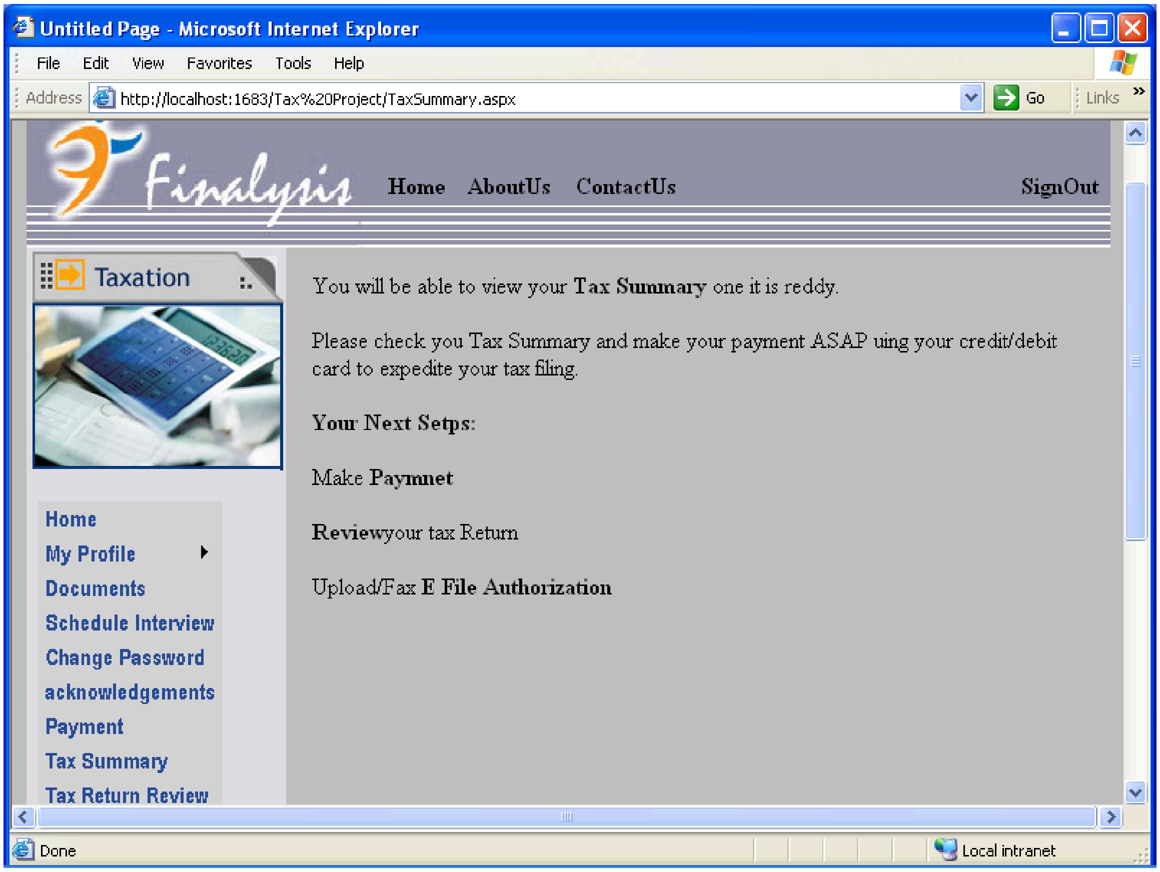
You can pay your **Tax Preparation Fees** here.

please check your Tax Summary and make your payment

using your **Credit/Debit card**

Bank Name	SBI
Debit/Credit Number	JY5781
Validity Date	05/05/2010

your amount will be credeted



You will be able to view your **Tax Summary** one it is reddy.

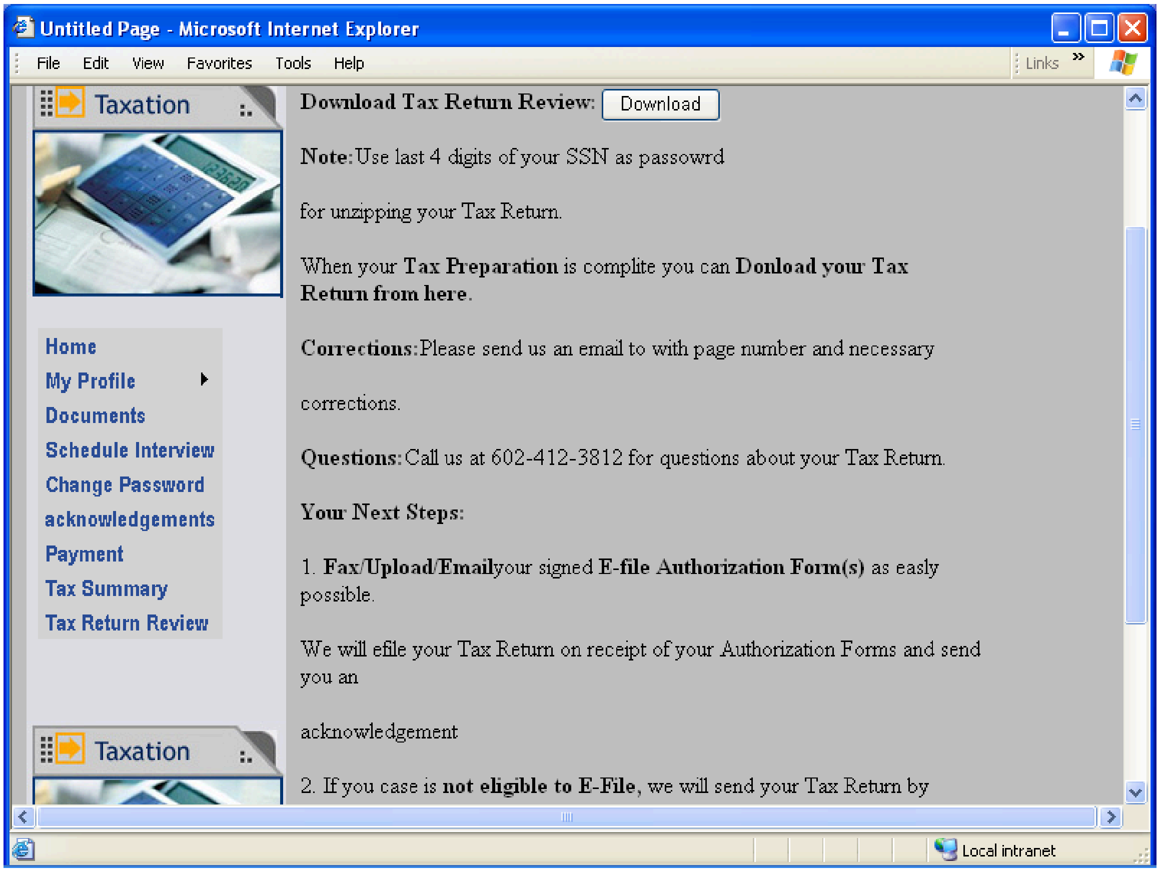
Please check you Tax Summary and make your payment ASAP uing your credit/debit card to expedite your tax filing.

Your Next Setps:

Make **Paymnet**

Reviewyour tax Return

Upload/Fax **E File Authorization**



Download Tax Return Review: Download

Note: Use last 4 digits of your SSN as passowrd

for unzipping your Tax Return.

When your **Tax Preparation** is complite you can **Donload your Tax Return from here**.

Corrections: Please send us an email to with page number and necessary

corrections.

Questions: Call us at 602-412-3812 for questions about your Tax Return.

Your Next Steps:

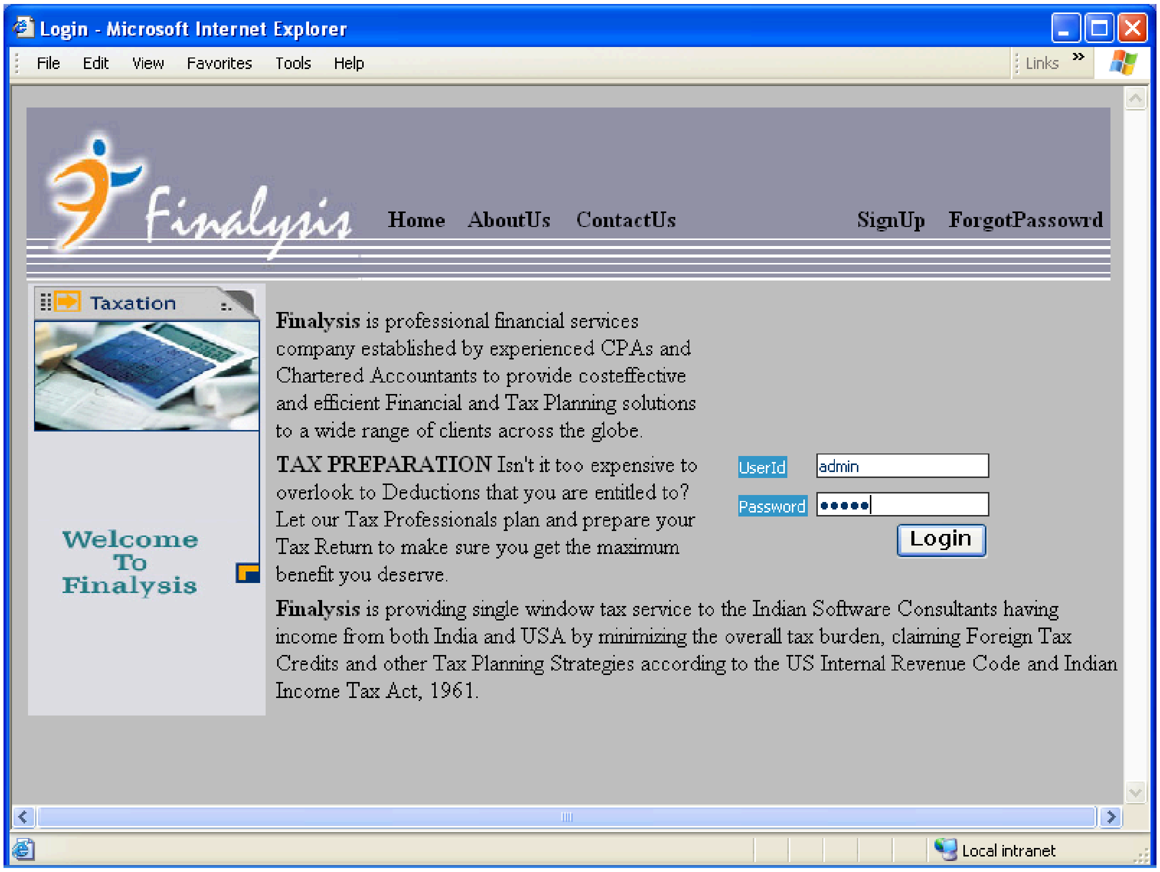
1. **Fax/Upload/Email**your signed **E-file Authorization Form(s)** as easly possible.

We will efile your Tax Return on receipt of your Authorization Forms and send you an

acknowledgement

2. If you case is **not eligible to E-File**, we will send your Tax Return by

ADMIN



Login - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Finalysis
Home AboutUs ContactUs SignUp ForgotPassowrd
Taxation
Welcome To Finalysis
Login
Local intranet

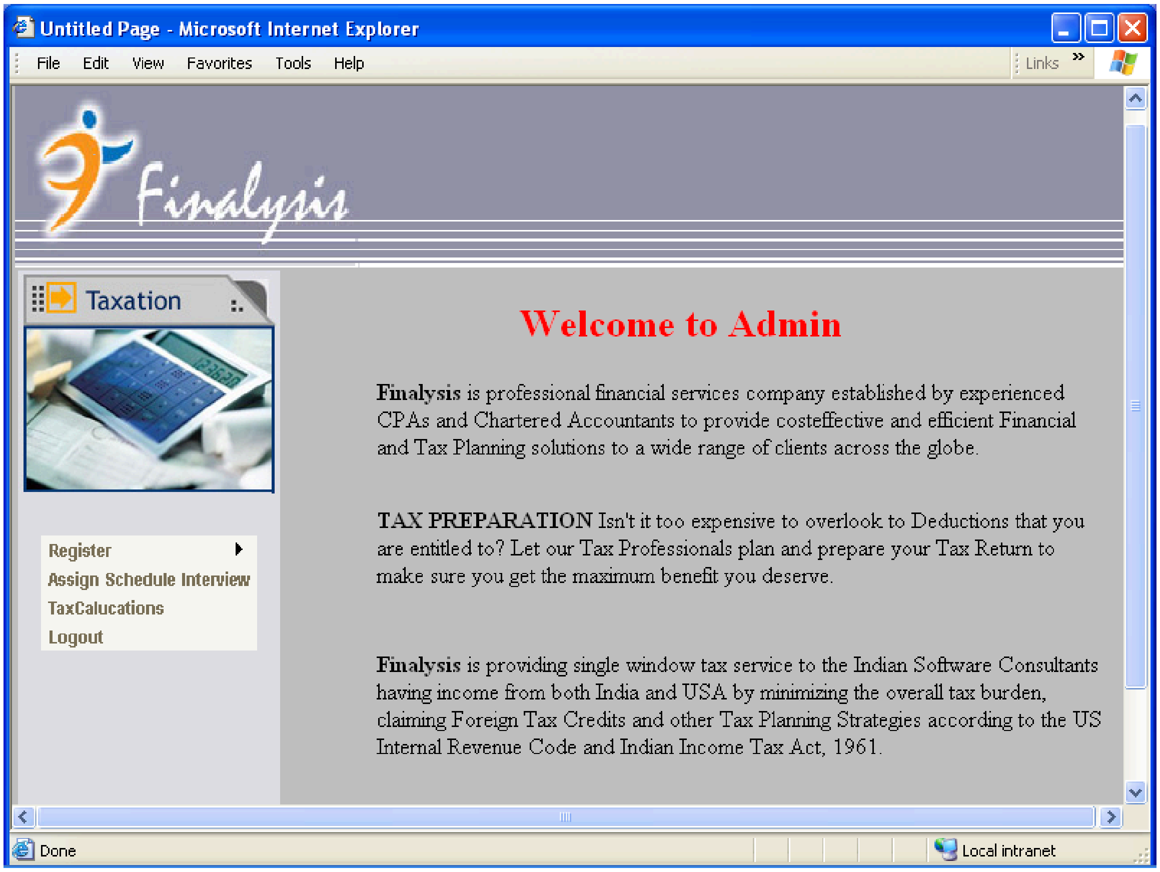
Finalysis is professional financial services company established by experienced CPAs and Chartered Accountants to provide costeffective and efficient Financial and Tax Planning solutions to a wide range of clients across the globe.

TAX PREPARATION Isn't it too expensive to overlook to Deductions that you are entitled to? Let our Tax Professionals plan and prepare your Tax Return to make sure you get the maximum benefit you deserve.

UserId: admin

Password: •••••

Finalysis is providing single window tax service to the Indian Software Consultants having income from both India and USA by minimizing the overall tax burden, claiming Foreign Tax Credits and other Tax Planning Strategies according to the US Internal Revenue Code and Indian Income Tax Act, 1961.

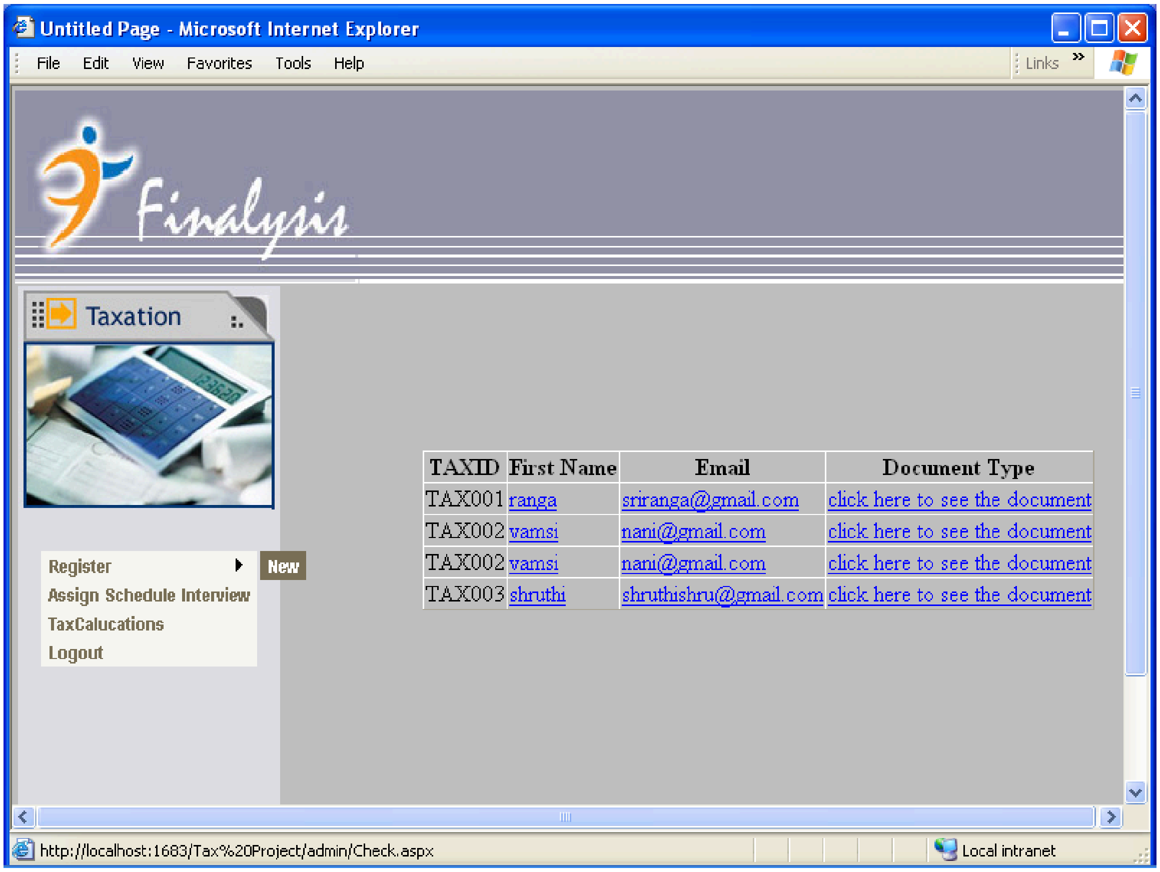


Welcome to Admin

Finalysis is professional financial services company established by experienced CPAs and Chartered Accountants to provide costeffective and efficient Financial and Tax Planning solutions to a wide range of clients across the globe.

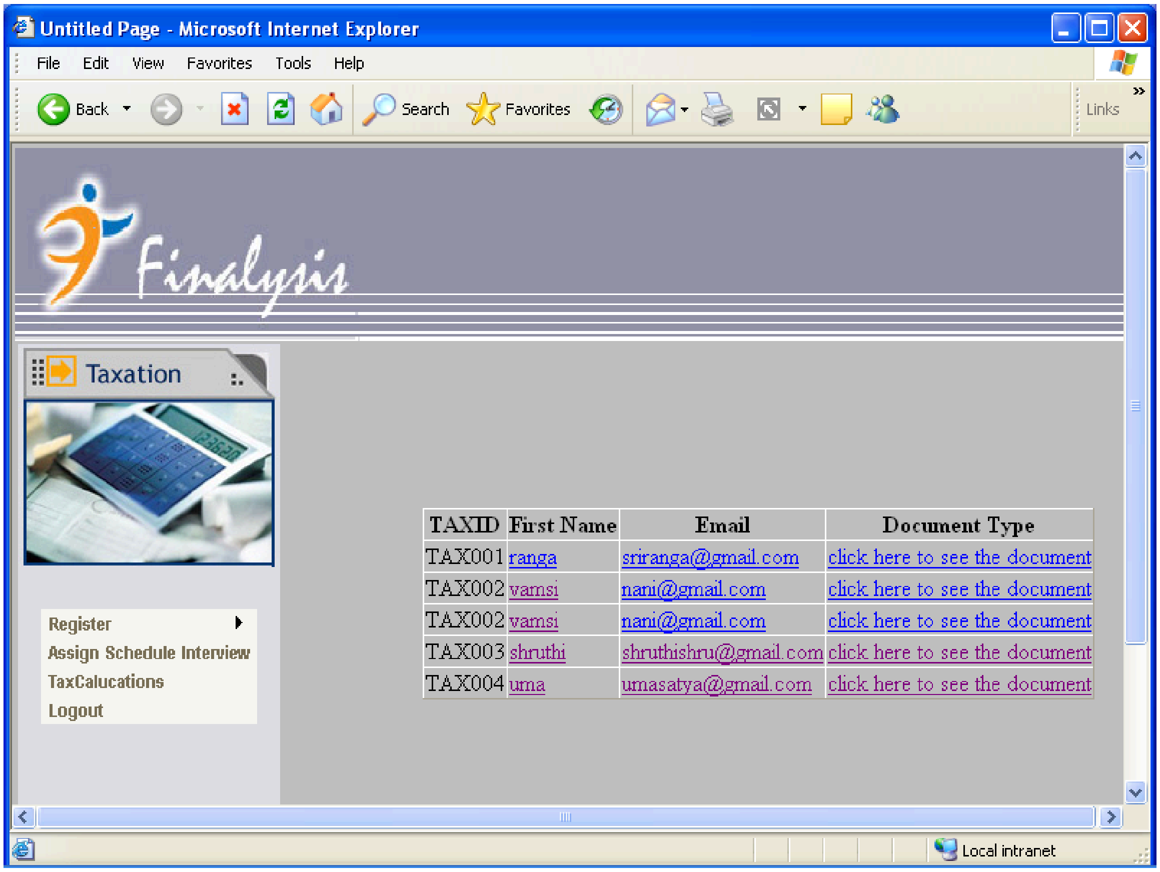
TAX PREPARATION Isn't it too expensive to overlook to Deductions that you are entitled to? Let our Tax Professionals plan and prepare your Tax Return to make sure you get the maximum benefit you deserve.

Finalysis is providing single window tax service to the Indian Software Consultants having income from both India and USA by minimizing the overall tax burden, claiming Foreign Tax Credits and other Tax Planning Strategies according to the US Internal Revenue Code and Indian Income Tax Act, 1961.



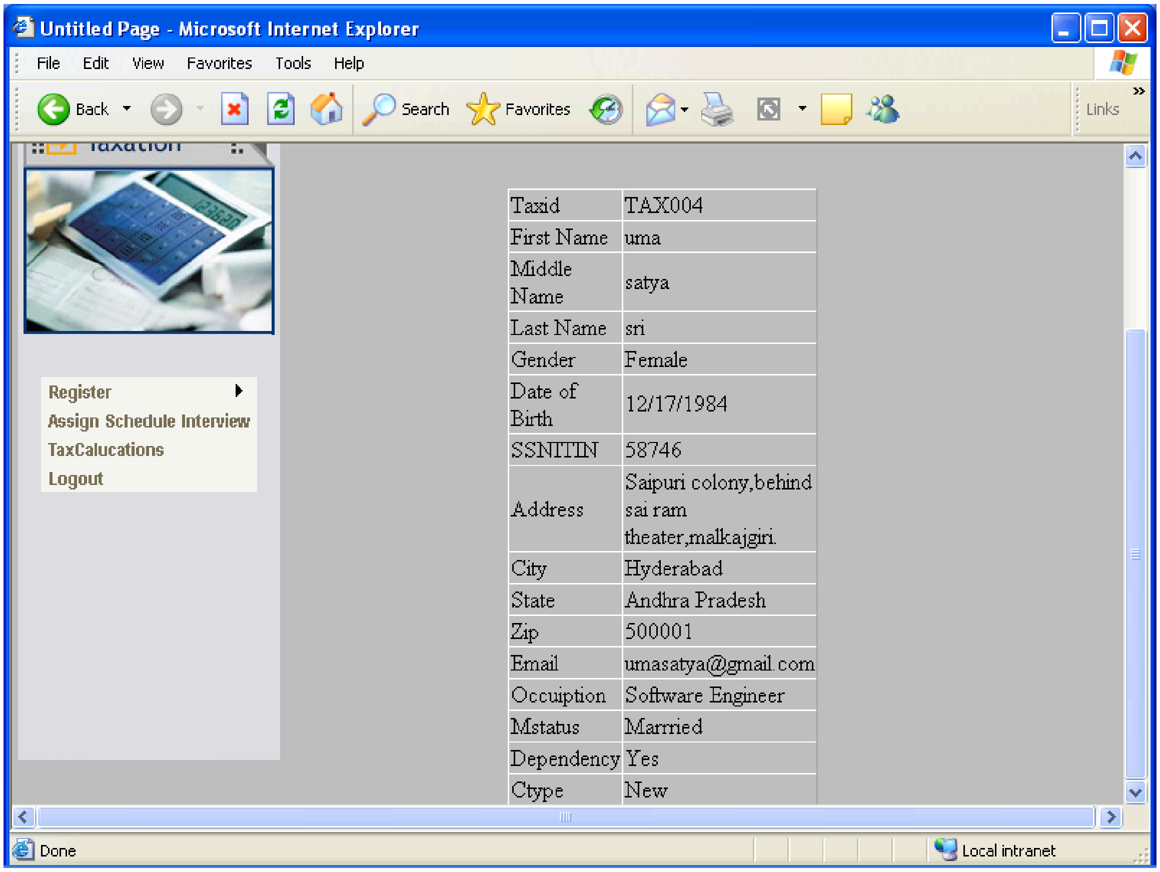
Untitled Page - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Links
Finalysis
Taxation
Register
New
Assign Schedule Interview
TaxCalucations
Logout
http://localhost:1683/Tax%20Project/admin/Check.aspx
Local intranet

TAXID	First Name	Email	Document Type
TAX001	ranga	sriranga@gmail.com	click here to see the document
TAX002	vamsi	nani@gmail.com	click here to see the document
TAX002	vamsi	nani@gmail.com	click here to see the document
TAX003	shruthi	shruthishru@gmail.com	click here to see the document

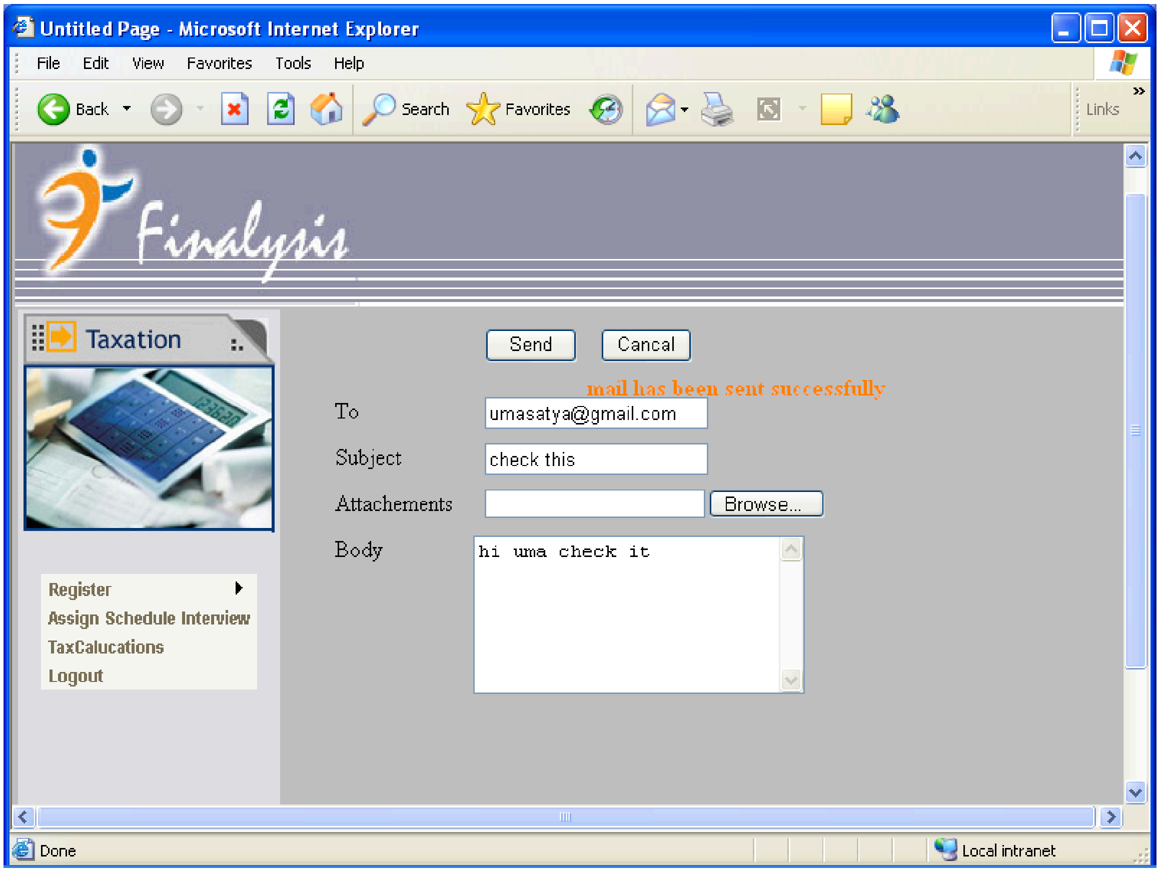


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Links
Finalysis
Taxation
Register
Assign Schedule Interview
TaxCalucations
Logout
Local intranet

TAXID	First Name	Email	Document Type
TAX001	ranga	sriranga@gmail.com	click here to see the document
TAX002	vamsi	nani@gmail.com	click here to see the document
TAX002	vamsi	nani@gmail.com	click here to see the document
TAX003	shruthi	shruthishru@gmail.com	click here to see the document
TAX004	uma	umasatya@gmail.com	click here to see the document



Taxid	TAX004
First Name	uma
Middle Name	satya
Last Name	sri
Gender	Female
Date of Birth	12/17/1984
SSNITIN	58746
Address	Saipuri colony,behind sai ram theater,malkajgiri.
City	Hyderabad
State	Andhra Pradesh
Zip	500001
Email	umasatya@gmail.com
Occuiption	Software Engineer
Mstatus	Married
Dependency	Yes
Ctype	New



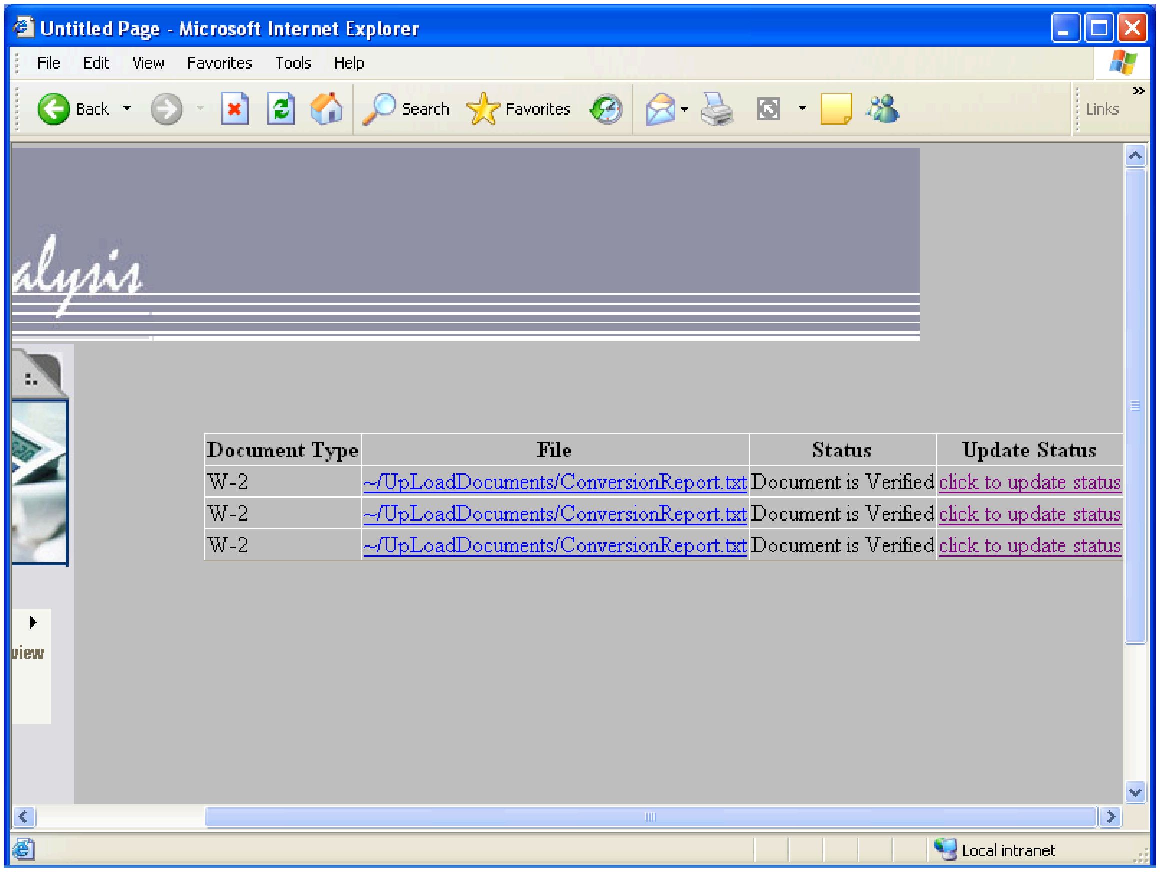
mail has been sent successfully

To umasatya@gmail.com

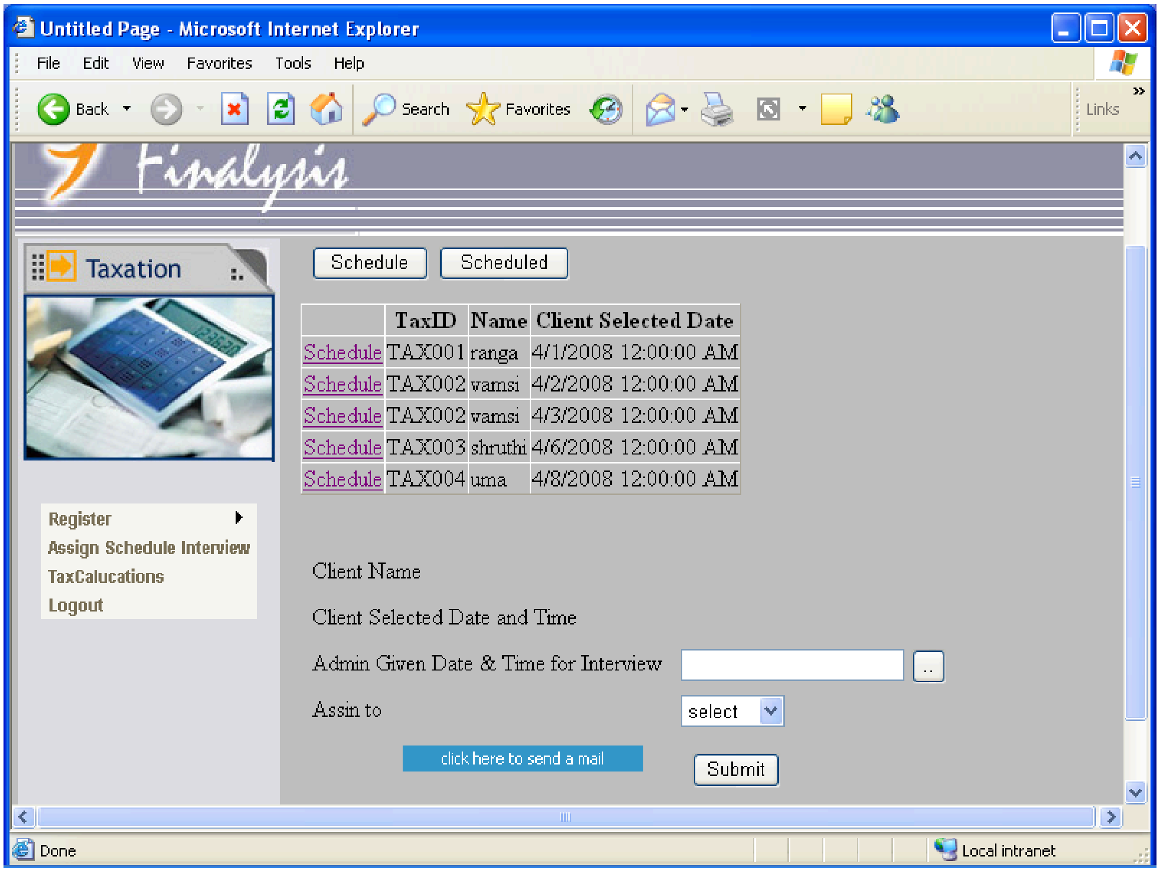
Subject check this

Attachements

Body hi uma check it



Document Type	File	Status	Update Status
W-2	~/UpLoadDocuments/ConversionReport.txt	Document is Verified	click to update status
W-2	~/UpLoadDocuments/ConversionReport.txt	Document is Verified	click to update status
W-2	~/UpLoadDocuments/ConversionReport.txt	Document is Verified	click to update status



	TaxID	Name	Client Selected Date
Schedule	TAX001	ranga	4/1/2008 12:00:00 AM
Schedule	TAX002	vamsi	4/2/2008 12:00:00 AM
Schedule	TAX002	vamsi	4/3/2008 12:00:00 AM
Schedule	TAX003	shruthi	4/6/2008 12:00:00 AM
Schedule	TAX004	uma	4/8/2008 12:00:00 AM

Client Name

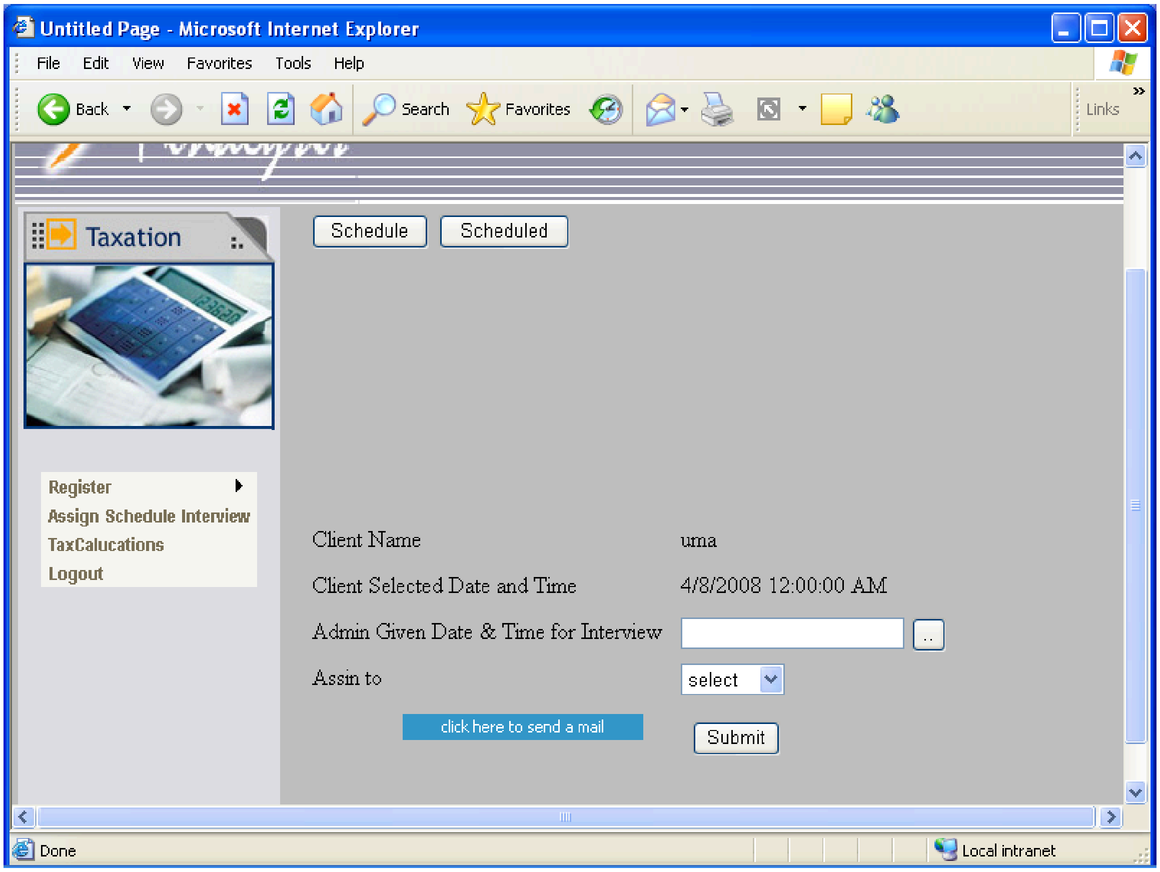
Client Selected Date and Time

Admin Given Date & Time for Interview

Assin to select

click here to send a mail

Submit

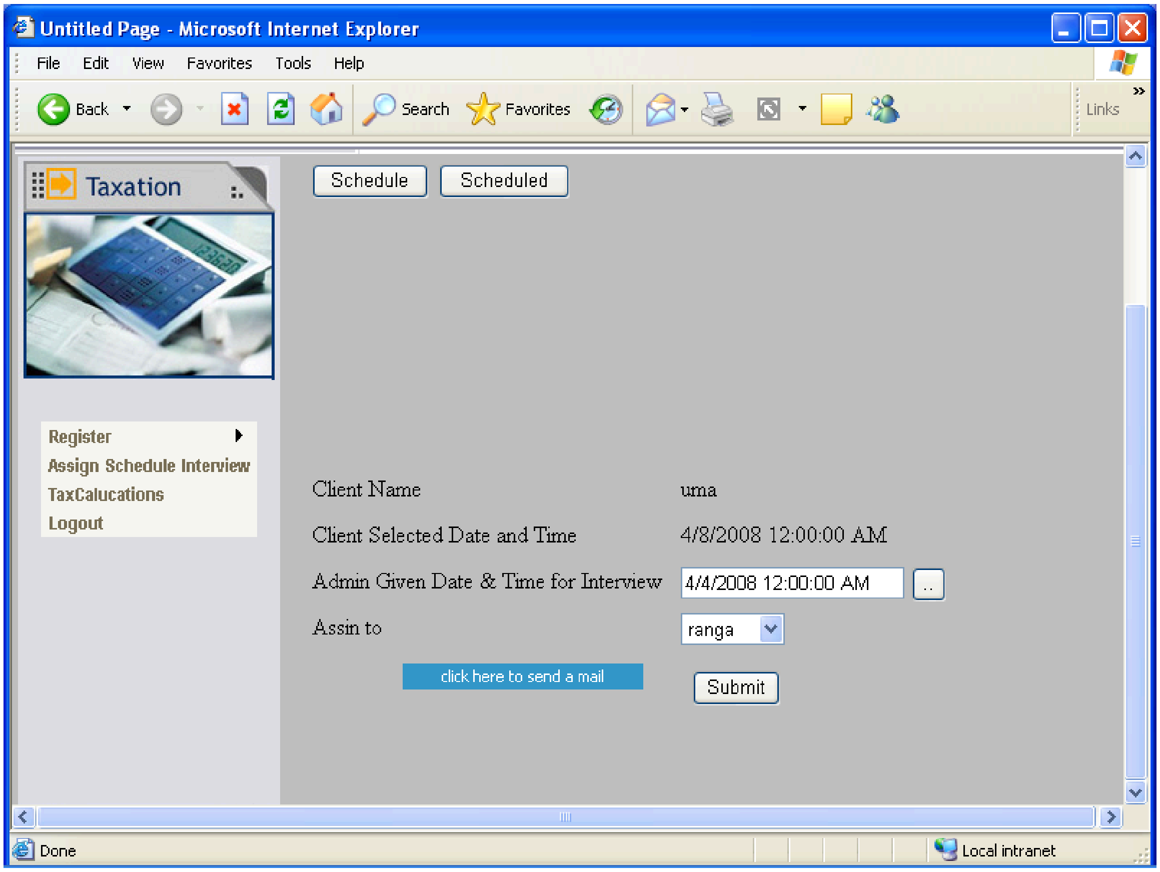


Client Name uma

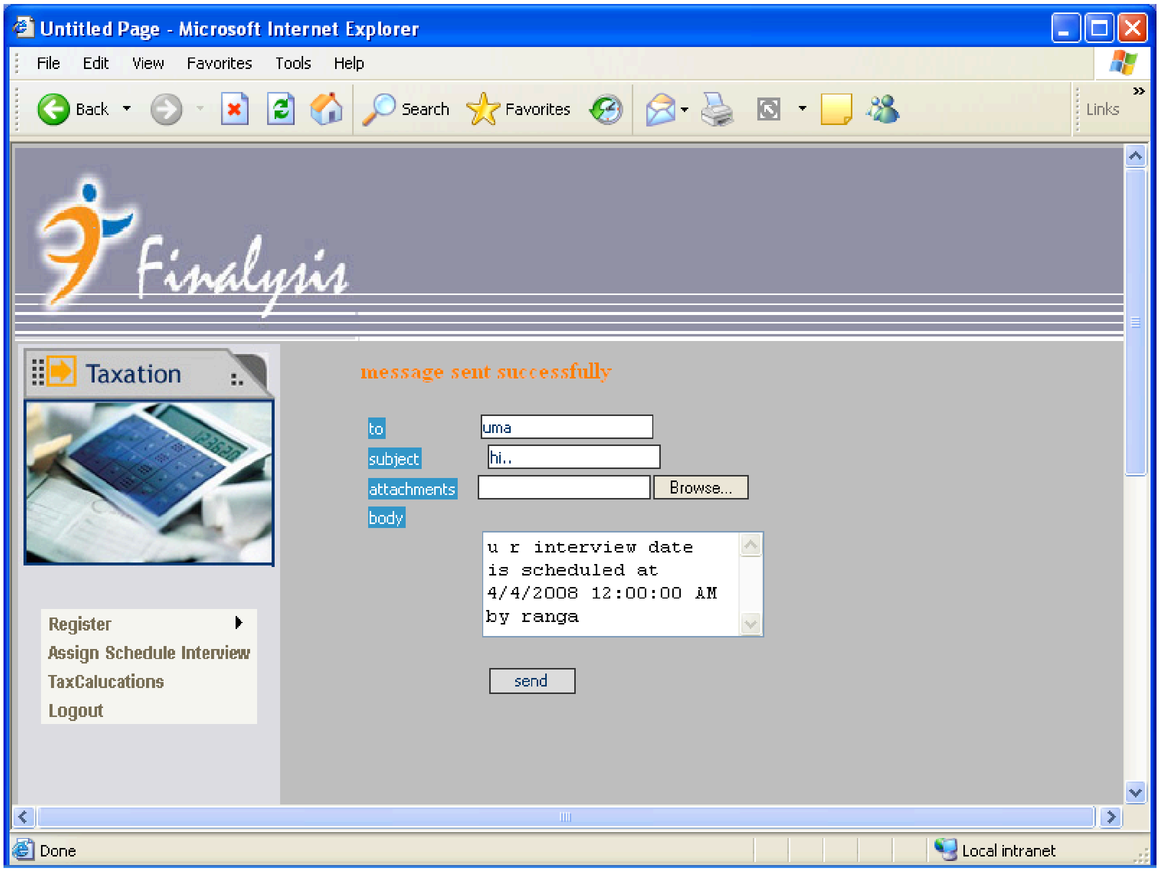
Client Selected Date and Time 4/8/2008 12:00:00 AM

Admin Given Date & Time for Interview

Assin to select



Client Name	uma
Client Selected Date and Time	4/8/2008 12:00:00 AM
Admin Given Date & Time for Interview	4/4/2008 12:00:00 AM
Assin to	ranga



message sent successfully

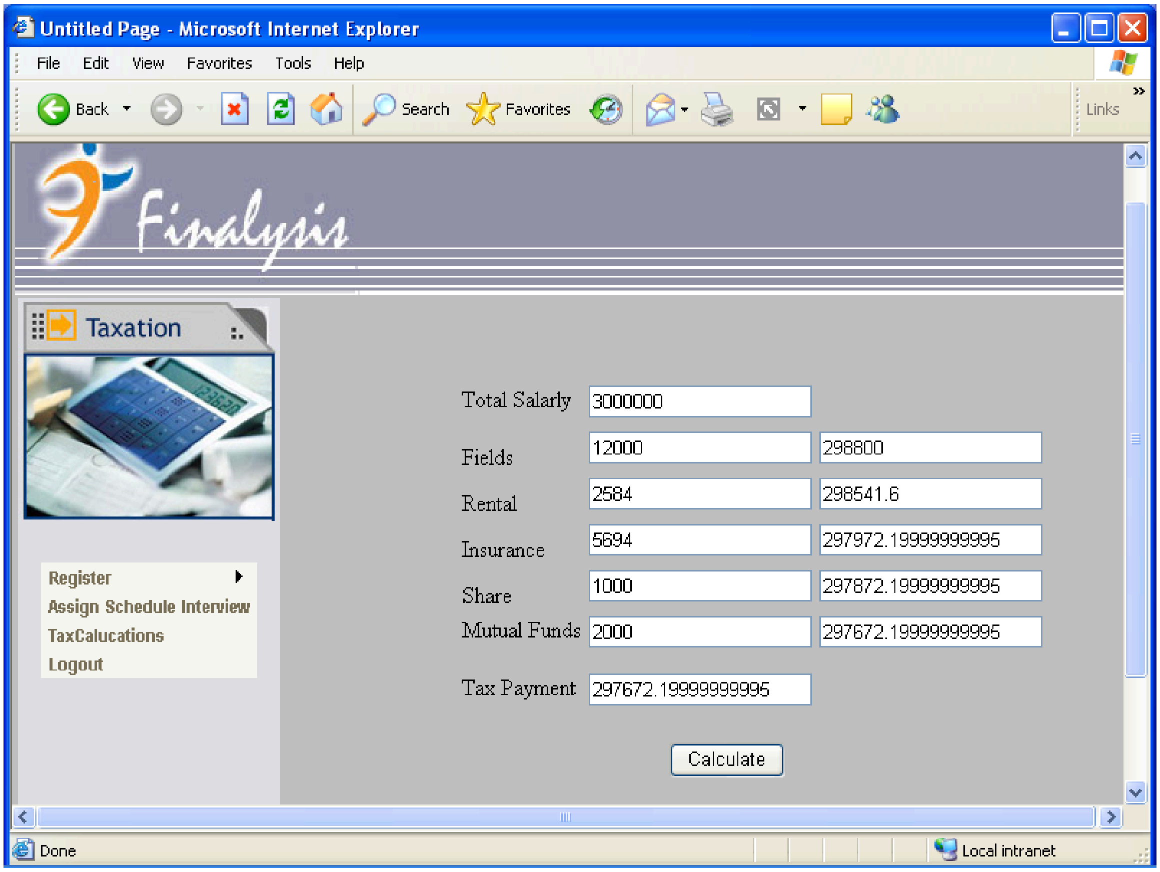
to: uma

subject: hi..

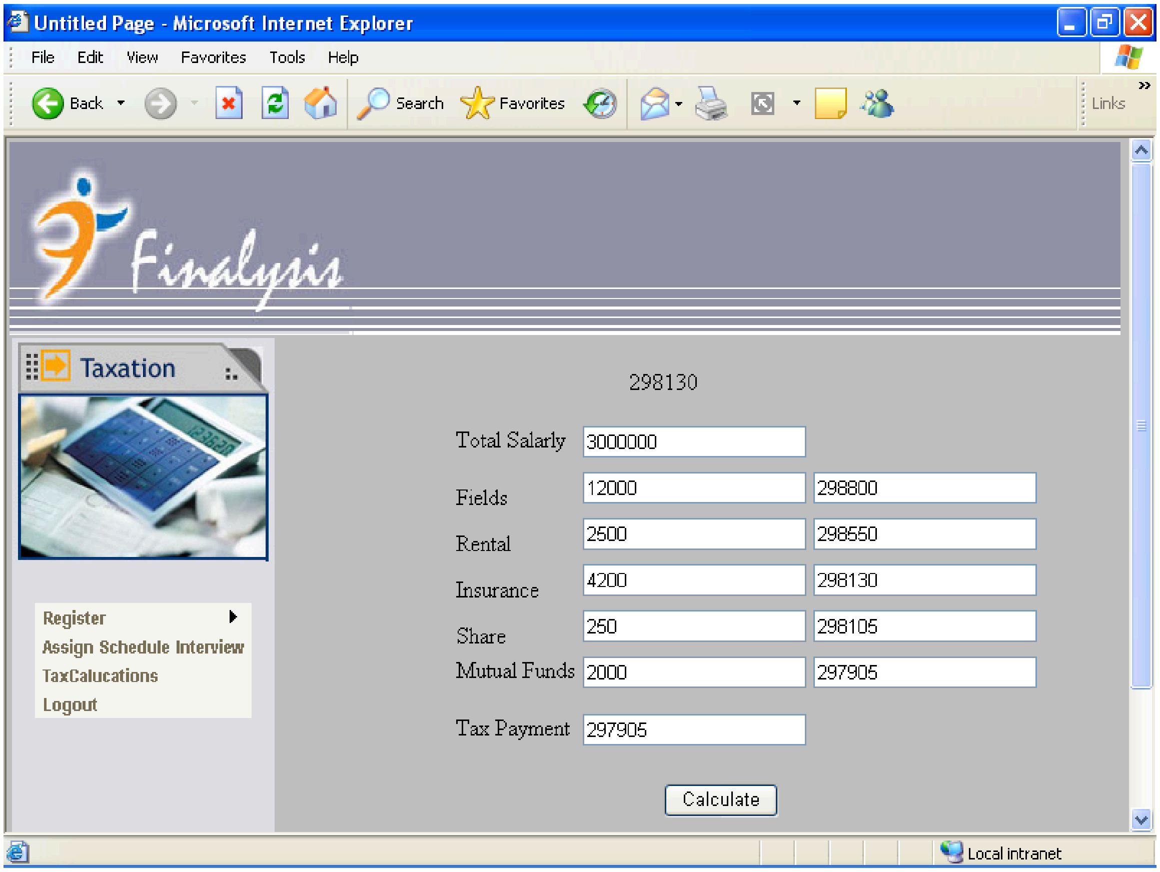
attachments:

body:

u r interview date
is scheduled at
4/4/2008 12:00:00 AM
by ranga



Total Salarly	3000000	
Fields	12000	298800
Rental	2584	298541.6
Insurance	5694	297972.19999999995
Share	1000	297872.19999999995
Mutual Funds	2000	297672.19999999995
Tax Payment	297672.19999999995	



298130

Total Salarly	3000000	
Fields	12000	298800
Rental	2500	298550
Insurance	4200	298130
Share	250	298105
Mutual Funds	2000	297905
Tax Payment	297905	

ADMINMODULE

Check.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class admin_Check : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql;
    protected void Page_Load(object sender, EventArgs e)
    {
       // str = Request.QueryString["taxid"].ToString();
        cn.Open();
        if (!IsPostBack)
        {
            FillData();
        }
    }

    public void FillData()
    {
        sql = "select m.taxid,m.fname,m.email,s.sdate from myprofile m,schedule s 
where m.taxid=s.taxid";
       // sql = "select m.taxid,m.fname,m.email,d.status,d.dtype,s.sdate from 
myprofile m,documents d,schedule s where m.taxid="+"'"+str+"'";
        SqlDataAdapter da = new SqlDataAdapter(sql, cn);
        DataSet ds = new DataSet();
        da.Fill(ds);
        GridView1.DataSource = ds;
        GridView1.DataBind();
    }
}
```

checkdocument.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class admin_CheckDocuments : System.Web.UI.Page
{
    SqlConnection cn = new SqlConnection("integrated 
security=true;database=satya");
    string sql;
    string str;
    protected void Page_Load(object sender, EventArgs e)
    {
        cn.Open();
        str = Request.QueryString["taxid"];
        if (!IsPostBack)
        {
           filldata();
        }
    }
    private void filldata()
    {

        SqlDataAdapter da = new SqlDataAdapter("select * from documents where
taxid='"+str+"'", cn);
        DataSet ds = new DataSet();
        da.Fill(ds);
        GridView1.DataSource = ds;
        GridView1.DataBind();
    }
    protected void GridView1_SelectedIndexChanging(object sender,
GridViewSelectEventArgs e)
    {
        Label lbldoc =
(Label)GridView1.Rows[e.NewSelectedIndex].FindControl("Lable1");

        sql=" update documents set status='Document is Verified' where
dtype='"+lbldoc.Text+"' and taxid='" + str + "'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        cmd.ExecuteNonQuery();
        filldata();

    }
}
```

C**heckProfile.aspx.cs**

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class admin_MyProfile : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql,str;
    protected void Page_Load(object sender, EventArgs e)
    {
        cn.Open();
        str = Request.QueryString["taxid"].ToString();
        if (!IsPostBack)
        {
            FillData();
        }
    }

    public void FillData()
    {
        sql = "select * from myprofile where taxid='"+str+"'";
        SqlDataAdapter da = new SqlDataAdapter(sql, cn);
        DataSet ds = new DataSet();
        da.Fill(ds);
        DetailsView1.DataSource = ds;
        DetailsView1.DataBind();
    }
}
```

New.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class admin_new : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql;
    protected void Page_Load(object sender, EventArgs e)
    {
        cn.Open();
        if (!IsPostBack)
        {
            //FillData();
            FillData1();
        }

    }

    public void FillData1()
    {
        sql = "select taxid,count(*) as count from documents where taxid in(select
taxid from myprofile) group by taxid";
        SqlDataAdapter da = new SqlDataAdapter(sql, cn);
        DataSet ds = new DataSet();
        da.Fill(ds);
        GridView2.DataSource = ds;
        GridView2.DataBind();

    }

    public void FillData()
    {
        //sql = "select m.taxid,m.fname,m.email,s.sdate,d.dtype,d.status from
myprofile m,documents d,schedule s where m.taxid=s.taxid and m.taxid=d.taxid";
        //SqlDataAdapter da = new SqlDataAdapter(sql, cn);
        //DataSet ds = new DataSet();
        //da.Fill(ds);
        //GridView1.DataSource = ds;
        //GridView1.DataBind();
    }
}
```

Schedule.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;


public partial class admin_Scheduled : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql;
    static string taxid;
    protected void Page_Load(object sender, EventArgs e)
    {
        cn.Open();
        GridView1.Visible = false;
        GridView2.Visible = false;

       // Calendar1.Visible = false;
```

```
        //Label1.Visible = false;
        //Label2.Visible = false;
        //Label3.Visible = false;
        //Label4.Visible = false;
        //Label5.Visible = false;
        //Label6.Visible = false;
        //Label7.Visible = false;
        //TextBox2.Visible = false;
        //DropDownList2.Visible = false;
        //Button1.Visible = false;
        //Button2.Visible = false;
        if (!IsPostBack)
        {
            FillData();
            FillData1();
        }
    }
    public void FillData()
    {
        sql = "select distinct m.taxid, m.fname,s.sdate,s.confimdate from myprofile
m,schedule s where m.taxid=s.taxid ";
        SqlDataAdapter da = new SqlDataAdapter(sql, cn);
        DataSet ds = new DataSet();
        da.Fill(ds);
        GridView1.DataSource = ds;
        GridView1.DataBind();
    }

    public void FillData1()
    {
        sql = "select distinct m.taxid, m.fname,s.sdate,s.confimdate,s.assign from
myprofile m,schedule s where m.taxid=s.taxid ";
        SqlDataAdapter da = new SqlDataAdapter(sql, cn);
        DataSet ds = new DataSet();
        da.Fill(ds);
        GridView2.DataSource = ds;
        GridView2.DataBind();
    }
    protected void GridView1_SelectedIndexChanging(object sender,
GridViewSelectEventArgs e)
    {
        Label
l=(Label)GridView1.Rows[e.NewSelectedIndex].FindControl("Label1");
        taxid = l.Text;
        sql = "select distinct m.taxid, m.fname,s.sdate,s.confimdate from myprofile
m,schedule s where m.taxid="+"'"+l.Text+"'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            Label4.Text = dr[1].ToString();
            Label5.Text = dr[2].ToString();
        }
        dr.Close();
    }
    protected void Button3_Click(object sender, EventArgs e)
    {
        GridView1.Visible = true;
        Label1.Visible = true;
        Label2.Visible = true;
        Label3.Visible = true;
        Label4.Visible = true;
        Label5.Visible = true;
        Label6.Visible = true;
        Label7.Visible = true;
        TextBox2.Visible = true;
        DropDownList2.Visible = true;
        Button1.Visible = true;
        Button2.Visible = true;
    }
    protected void Button4_Click(object sender, EventArgs e)
    {
        GridView2.Visible = true;
        Label1.Visible = false;
        Label2.Visible = false;
        Label3.Visible = false;
        Label4.Visible = false;
        Label5.Visible = false;
        Label6.Visible = false;
        Label7.Visible = false;
        TextBox2.Visible = false;
        DropDownList2.Visible = false;
        Button1.Visible = false;
        Button2.Visible = false;
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
        Calendar1.Visible = true;
    }
    //Label l;
    protected void Button1_Click(object sender, EventArgs e)
    {

        //for (int i = 0; i < GridView2.Rows.Count - 1; i++)
        //{
        //    l = (Label)GridView2.Rows[i].FindControl("Label2");
        //}
            //l = (Label)grv.FindControl("Label2");
        //foreach (GridViewRow gr in GridView1.Rows)
        //{
        //    l=(Label)gr.FindControl("
        //}
        sql = "update schedule set status='Yes',confimdate='" + TextBox2.Text +
"',assign='" + DropDownList2.SelectedItem.Text + "' where taxid='" +taxid+ "'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        cmd.ExecuteNonQuery();
        FillData1();
    }
    protected void Calendar1_SelectionChanged(object sender, EventArgs e)
    {
        if (Calendar1.SelectedDate <= DateTime.Now)
        {
            Label7.Text = "Your Selected Date is not Correct";
        }
        else
        {
            TextBox2.Text = Calendar1.SelectedDate.ToString();
        }
    }
}
```

CLIENT MODULE

Login.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class Login : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql,sql1;

    protected void Page_Load(object sender, EventArgs e)
    {
        cn.Open();
    }
    protected void btnlogin_Click(object sender, EventArgs e)
    {
        if ((txtuserid.Text == "admin") && (txtpwd.Text == "babu"))
        {
            Response.Redirect("~/admin/adminpage.aspx");
        }
        else
        {
            sql = "select taxid from taxregistration where userid='" + txtuserid.Text + "'
and pwd='" + txtpwd.Text + "'";
            SqlCommand cmd = new SqlCommand(sql, cn);
            SqlDataReader dr = cmd.ExecuteReader();
            if (dr.Read())
            {
                Session["taxid"] = dr[0].ToString();
                Response.Redirect("welcome.aspx");
            }

            else
            {
                sql1 = "select userid from interviewer where userid='" + txtuserid.Text +
"' and pwd='" + txtpwd.Text + "'";
                SqlCommand cmd1 = new SqlCommand(sql1, cn);
                SqlDataReader dr1 = cmd1.ExecuteReader();
                if (dr1.Read())
                {
                    Session["userid"] = dr1[0].ToString();
                    Response.Redirect("~/Interviewer/Interview.aspx");
                }
                else
                {
                    lblmsg.Text = "Invalid UserID and Password";
                }

                dr1.Close();
            }
            dr.Close();

        }
    }
}
```

Myprofile.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class MyProfile : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql,sql1,sql2,sql3,str;
    protected void Page_Load(object sender, EventArgs e)
    {

        lbltaxid.Text = Session["taxid"].ToString();
        cn.Open();
        if (!IsPostBack)
        {
            FillData();
        }
    }

    public void FillData()
    {
        sql = "select * from myprofile where taxid=" + "'" + Session["taxid"] + "'";
        ///Response.Write(sql);
        SqlCommand cmd = new SqlCommand(sql, cn);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            lbltaxid.Text = dr[0].ToString();
            txtfname.Text = dr[1].ToString();
            txtmname.Text = dr[2].ToString();
            txtlname.Text = dr[3].ToString();
            radiogender.Text = dr[4].ToString();
            txtdob.Text = dr[5].ToString();
            txtssn.Text = dr[6].ToString();
            txtaddress.Text = dr[7].ToString();
            txtcity.Text = dr[8].ToString();
            txtstate.Text = dr[9].ToString();
            txtzip.Text = dr[10].ToString();
            txtemail.Text = dr[11].ToString();
            listoccuiption.Text = dr[12].ToString();
            radiomaritalstatus.Text = dr[13].ToString();
            radiodependency.Text = dr[14].ToString();
        }
    }
    protected void btnregister_Click(object sender, EventArgs e)
    {
        str = "New";

        //lbltaxid.Text = Session["taxid"].ToString();
        sql = "select taxid from myprofile where taxid=" + "'" + Session["taxid"] +
"'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        SqlDataReader dr=cmd.ExecuteReader();
        if (dr.Read())
        {
            sql1 = "update myprofile set fname='" + txtfname.Text + "',mname='" +
txtmname.Text + "',lname='" + txtlname.Text + "',gender='" +
radiogender.SelectedItem.Text + "',dob='" + txtdob.Text + "',ssnitin='" +
txtssn.Text + "',address='" + txtaddress.Text + "',city='" + txtcity.Text + "',state='" +
txtstate.Text + "',zip='" + txtzip.Text + "',email='" + txtemail.Text + "',occuiption
='" + listoccuiption.SelectedItem.Text + "',mstatus='" +
radiomaritalstatus.SelectedItem.Text + "',dependency='" +
radiodependency.SelectedItem.Text + "' where taxid=" + "'" + Session["taxid"] +
"'";
            SqlCommand cmd1 = new SqlCommand(sql1, cn);
            cmd1.ExecuteNonQuery();
        }
        else if (txtssn.Text == "")
        {
            sql2 = "insert
myprofile(taxid,fname,mname,lname,gender,dob,address,city,state,zip,email,occui
ption,mstatus,dependency,ctype) values('" + Session["taxid"].ToString() + "','" +
txtfname.Text + "','" + txtmname.Text + "','" + txtlname.Text + "','" +
radiogender.SelectedItem.Text + "','" + txtdob.Text + "','" + txtaddress.Text + "','" +
txtcity.Text + "','" + txtstate.Text + "','" + txtzip.Text + "','" + txtemail.Text + "','" +
listoccuiption.SelectedItem.Text + "','" + radiomaritalstatus.SelectedItem.Text +
"','" + radiodependency.SelectedItem.Text + "','" + str + "')";
            //Label2.Text = sql2;
            SqlCommand cmd2 = new SqlCommand(sql2, cn);
            cmd2.ExecuteNonQuery();
            //Clear();
        }
        else
        {
            sql3 = "insert
myprofile(taxid,fname,mname,lname,gender,dob,ssnitin,address,city,state,zip,email
,occuiption,mstatus,dependency,ctype) values('" + Session["taxid"].ToString() +
"','" + txtfname.Text + "','" + txtmname.Text + "','" + txtlname.Text + "','" +
radiogender.SelectedItem.Text + "','" + txtdob.Text + "','" + txtssn.Text + "','" +
txtaddress.Text + "','" + txtcity.Text + "','" + txtstate.Text + "','" + txtzip.Text + "','"
+ txtemail.Text + "','" + listoccuiption.SelectedItem.Text + "','" +
radiomaritalstatus.SelectedItem.Text + "','" + radiodependency.SelectedItem.Text +
"','" + str + "')";
            //Label2.Text = sql2;
            SqlCommand cmd3 = new SqlCommand(sql3, cn);
            cmd3.ExecuteNonQuery();
        }
        dr.Close();
    }
    public void Clear()
    {
        txtfname.Text = "";
        txtmname.Text = "";
        txtlname.Text = "";
        foreach (ListItem li in radiogender.Items)
        {
            li.Selected = false;
        }
        txtdob.Text = "";
        txtssn.Text = "";
        txtaddress.Text = "";
        txtcity.Text = "";
        txtstate.Text = "";
        txtzip.Text = "";
        txtemail.Text = "";
        foreach (ListItem li1 in listoccuiption.Items)
        {
            li1.Selected = false;
        }
        foreach (ListItem li2 in radiomaritalstatus.Items)
        {
            li2.Selected  = false;
        }
        foreach (ListItem li3 in radiodependency.Items)
        {
            li3.Selected = false;
        }
    }
    protected void radiomaritalstatus_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (radiomaritalstatus.SelectedItem.Text == "Marrried")
        {
            Response.Write("<script type='text/javascript'>");
            Response.Write("window.open('Spouse.aspx')");
            Response.Write("</script>");
        }
    }
    protected void radiodependency_SelectedIndexChanged(object sender,
EventArgs e)
```

```
    {
        if (radiodependency.SelectedItem.Text == "Yes")
        {
            Response.Write("<script type='text/javascript'>");
            Response.Write("window.open('Dependent.aspx')");
            Response.Write("</script>");
        }
    }
}
```

Spouse.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class Spouse : System.Web.UI.Page
{

    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql;
    string str;
    protected void Page_Load(object sender, EventArgs e)
    {
        cn.Open();
        if (!IsPostBack)
        {
            FillData();
        }
    }
    public void FillData()
    {
        sql = "select * from spouse where taxid=" + "'" + Session["taxid"] + "'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            txtfname.Text = dr[1].ToString();
            txtmname.Text = dr[2].ToString();
            txtlname.Text = dr[3].ToString();
            txtdob.Text = dr[4].ToString();
            occupationdropdownlist.Text = dr[5].ToString();
            string str = dr[6].ToString();
            if (str == "Yes")
            {
                checkssntax.Checked = true;
            }
            else
                checkssntax.Checked = false;
        }

    }

    protected void btnsubmit_Click(object sender, EventArgs e)
    {
        if (checkssntax.Checked == true)
            str = "Yes";
        else

            str = "No";
        sql = "insert into spouse values('" + Session["taxid"] + "','" + txtfname.Text +
"','" + txtmname.Text + "','" + txtlname.Text + "','" + txtdob.Text + "','" +
occupationdropdownlist.SelectedItem.Text + "','" + str + "')";
        SqlCommand cmd = new SqlCommand(sql, cn);
        cmd.ExecuteNonQuery();



        Response.Write("<script type='text/javascript'>");
        Response.Write("window.close('Spouse.aspx')");
        Response.Write("</script>");
    }
    protected void checkssntax_CheckedChanged(object sender, EventArgs e)
    {
        if (checkssntax.Checked == true)
        {
            Response.Write("<script type='text/javascript'>");
            Response.Write("window.open('ITINSpouse.aspx')");
            Response.Write("</script>");

            Session["fname"] = txtfname.Text;
            Session["mname"] = txtmname.Text;
            Session["lname"] = txtlname.Text;
        }
    }
    protected void txtfname_TextChanged(object sender, EventArgs e)
    {

    }
}
```

ITINSpouse.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class ITIN : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    //SqlConnection cn =new SqlConnection  ("integrated
security=true;database=satya");
    string sql;

    protected void Page_Load(object sender, EventArgs e)
    {


        cn.Open();
        if (!IsPostBack)
        {
          FillData();
        }
    }
    public void FillData()
    {
        sql = "select * from itinspouse where taxid=" + "'" + Session["taxid"] + "'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            txtfname.Text = dr[1].ToString();
            txtmname.Text = dr[2].ToString();
            txtlname.Text = dr[3].ToString();
            txtpassportno.Text = dr[4].ToString();
            txtpassportedate.Text = dr[5].ToString();
            txtvisano.Text = dr[6].ToString();
            txtvisaedate.Text = dr[7].ToString();
            txtplaceofbirth.Text = dr[8].ToString();
            txtaddressinthepassport.Text = dr[9].ToString();
            txtdateofentryusa.Text = dr[10].ToString();
        }
        else
        {
            txtfname.Text = Session["fname"].ToString();
            txtmname.Text = Session["mname"].ToString();
            txtlname.Text = Session["lname"].ToString();
        }
        dr.Close();
    }
    protected void btnsubmit_Click(object sender, EventArgs e)
    {
        sql = "insert into itinspouse values('" + Session["taxid"] + "','" + txtfname.Text
+ "','" + txtmname.Text + "','" + txtlname.Text + "','" + txtpassportno.Text + "','" +
txtpassportedate.Text + "','" + txtvisano.Text + "','" + txtvisaedate.Text + "','" +
txtplaceofbirth.Text + "','" + txtaddressinthepassport.Text + "','" +
txtdateofentryusa.Text + "')";
        SqlCommand cmd = new SqlCommand(sql, cn);
        cmd.ExecuteNonQuery();

        Response.Write("<script type='text/javascript'>");
        Response.Write("window.close('ITINSpouse.aspx')");
        Response.Write("</script>");
    }
}
```

Dependent.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class Dependent : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql,sql1,str;
    static DataSet ds;
    static int n;


    protected void Page_Load(object sender, EventArgs e)
    {
        cn.Open();
        lblother.Visible = false;
        txtother.Visible = false;
        lblmsg3.Visible = false;
        if (!IsPostBack)
        {
            FillData();
            FillData1();
        }
    }

    public void FillData1()
    {
        sql = "select * from dependent";
        SqlDataAdapter da = new SqlDataAdapter(sql, cn);
        ds = new DataSet();
        da.Fill(ds, "dependent");
    }

    private void GetRecord(int n)
    {
        txtfname.Text = ds.Tables[0].Rows[n].ItemArray[1].ToString();
        txtmname.Text = ds.Tables[0].Rows[n].ItemArray[2].ToString();
        txtlname.Text = ds.Tables[0].Rows[n].ItemArray[3].ToString();
        droprelation.Text = ds.Tables[0].Rows[n].ItemArray[4].ToString();
        txtother.Text = ds.Tables[0].Rows[n].ItemArray[5].ToString();
        txtdob.Text = ds.Tables[0].Rows[n].ItemArray[6].ToString();
        string str = ds.Tables[0].Rows[n].ItemArray[7].ToString();
        if (str == "Yes")
        {
            checkssntax.Checked = true;
        }
        else
        {
            checkssntax.Checked = false;
        }
        //checkssntax.Text = ds.Tables[0].Rows[n].ItemArray[7].ToString();
        txtnoofyear.Text = ds.Tables[0].Rows[n].ItemArray[8].ToString();
    }

    public void FillData()
    {
        sql = "select * from dependent where taxid=" + "'" +
Session["taxid"].ToString() + "'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            txtfname.Text = dr[1].ToString();
            txtmname.Text = dr[2].ToString();
            txtlname.Text = dr[3].ToString();
            droprelation.Text = dr[4].ToString();
            txtother.Text = dr[5].ToString();
            txtdob.Text = dr[6].ToString();
            string str = dr[7].ToString();
            if (str == "Yes")
            {
                checkssntax.Checked = true;
            }
            else
            {
                checkssntax.Checked = false;
            }
            txtnoofyear.Text = dr[8].ToString();
        }
    }
    protected void btnsubmit_Click(object sender, EventArgs e)
    {
        if (checkssntax.Checked == true)
            str = "Yes";
        else
            str = "No";

        if (droprelation.SelectedItem.Text == "Other")
        {
            sql = "insert into dependent values('" + Session["taxid"] + "','" +
txtfname.Text + "','" + txtmname.Text + "','" + txtlname.Text + "','" +
droprelation.SelectedItem.Text + "','" + txtother.Text + "','" + txtdob.Text + "','" +
str + "','" + txtnoofyear.Text + "')";
            SqlCommand cmd = new SqlCommand(sql, cn);
            cmd.ExecuteNonQuery();
        }
        else
        {
            sql1 = "insert into
dependent(taxid,fname,mname,lname,relation,dob,ssnitin,noofyearsinusa) values('"
+ Session["taxid"] + "','" + txtfname.Text + "','" + txtmname.Text + "','" +
txtlname.Text + "','" + droprelation.SelectedItem.Text + "','" + txtdob.Text + "','" +
str + "','" + txtnoofyear.Text + "')";
            SqlCommand cmd1 = new SqlCommand(sql1, cn);
            cmd1.ExecuteNonQuery();
        }

        Response.Write("<script type='text/javascript'>");
        Response.Write("window.close('Dependent.aspx')");
        Response.Write("</script>");
    }
    protected void droprelation_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (droprelation.SelectedItem.Text == "Other")
        {
            lblother.Visible = true;
            txtother.Visible = true;
        }
    }
    protected void checkssntax_CheckedChanged(object sender, EventArgs e)
    {
        if (checkssntax.Checked == true)
        {
            Response.Write("<script type='text/javascript'>");
            Response.Write("window.open('ITINDependent.aspx')");
            Response.Write("</script>");


                    //Session["fname"] = txtfname.Text;
                    //Session["mname"] = txtmname.Text;
                    //Session["lname"] = txtlname.Text;

        }
    }
    protected void btnnext_Click(object sender, EventArgs e)
    {
        if (n < ds.Tables[0].Rows.Count - 1)
        {
            n += 1;
            GetRecord(n);
        }
        else
        {
            lblmsg3.Visible = true;
            lblmsg3.Text = "This is Last Record";
        }
    }
    protected void btnprevious_Click(object sender, EventArgs e)
    {
        if (n > 0)
        {
            n -= 1;
            GetRecord(n);
        }
        else
        {
            lblmsg3.Visible = true;
            lblmsg3.Text = "This is First Record";
        }
    }
}
```

ITINDependent.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class ITINDependent : System.Web.UI.Page
{
    SqlConnection cn = new
SqlConnection(ConfigurationManager.AppSettings["con"]);
    string sql;

    protected void Page_Load(object sender, EventArgs e)
    {


        cn.Open();
        if (!IsPostBack)
        {
            FillData();
        }
    }
    public void FillData()
    {
        sql = "select * from itindependent where taxid=" + "'" +
Session["taxid"].ToString()+ "'";
        SqlCommand cmd = new SqlCommand(sql, cn);
        SqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            txtfname.Text = dr[1].ToString();
            txtmname.Text = dr[2].ToString();
            txtlname.Text = dr[3].ToString();
            txtpassportno.Text = dr[4].ToString();
            txtpassportedate.Text = dr[5].ToString();
            txtvisano.Text = dr[6].ToString();
            txtvisaedate.Text = dr[7].ToString();
            txtplaceofbirth.Text = dr[8].ToString();
            txtaddressinthepassport.Text = dr[9].ToString();
            txtdateofentryusa.Text = dr[10].ToString();
        }
        //else
        //{
        //    //if(Session["fname"].ToString()!= "")
        //    //{
        //    txtfname.Text = Session["fname"].ToString();
        //    txtmname.Text = Session["mname"].ToString();
        //    txtlname.Text = Session["lname"].ToString();
        //    //}
        //}
        dr.Close();
    }
    protected void btnsubmit_Click(object sender, EventArgs e)
    {
        sql = "insert into itindependent values('" + Session["taxid"] + "','" +
txtfname.Text + "','" + txtmname.Text + "','" + txtlname.Text + "','" +
txtpassportno.Text + "','" + txtpassportedate.Text + "','" + txtvisano.Text + "','" +
txtvisaedate.Text + "','" + txtplaceofbirth.Text + "','" + txtaddressinthepassport.Text
+ "','" + txtdateofentryusa.Text + "')";
        SqlCommand cmd = new SqlCommand(sql, cn);
```

```
            cmd.ExecuteNonQuery();


            Response.Write("<script type='text/javascript'>");
            Response.Write("window.close('ITINDependent.aspx')");
            Response.Write("</script>");
        }
    }
```

TEST CASE NAME : ADMIN LOGIN

Description	Input	Expected Output	Actual Output	Remarks
User name	Admin	Control passes to password field	Control passes to password field	pass
Password	****	Control passes to home page	Control does not passes to home page	fail
Password	****	Control passes to home page	Control passes to home page	pass

TEST CASE NAME : CLIENT LOGIN

Description	Input	Expected Output	Actual Output	Remarks
User name	Britle	Control passes to password field	Control passes to password field	pass
Password	****	Control passes to home page	Control does not passes to home page(user id does not matches with password)	fail
Password	****	Control passes to home page	Control passes to home page	pass

TEST CASE NAME : MYPROFILE

Description	Input	Expected Output	Actual Output	Remarks
Taxid	Anil	Control passes to next field	Control passes to next field	pass
Fname	Venkata	Control passes to next field	Control does not passes to next field	pass
Mname	Anil	Control passes to next field	Control passes to next field	pass
Lname	Kumar	Control passes to next field	Control passes to next field	pass
D.O.B	02/06/1984	Control passes to next field	Control passes to next field	pass
SSNtin	5569	Control passes to next field	Control passes to next field	pass
Address	Vijayawada	Control passes to next field	Control passes to next field	pass
City	Hyderabad	Control passes to next field	Control passes to next field	pass
State	Andhra Pradesh	Control passes to next field	Control passes to next field	pass
Zip	500001	Control passes to next field	Control passes to next field	pass

Email	anil@gmail.com	Control passes to next field	Control passes to next field	pass
Occupation	Software Engineer	Control passes to next field	Control passes to next field	pass
Mstatus	yes	Control goes to next page	Control goes tonext page	pass
Dependency	yes	Control goes to next page	Control goes to next page	pass
Ctype	New	Control goes to next page	Control goes to next page	pass
Bank	yes	Control goes to next page	Control goes to next page	pass
Vehicle	yes	Control goes to next page	Control goes to next page	pass
Pnumber	B1758469	Control passes to next field	Control passes to next field	pass
Pexp	02/05/2018	Control passes to next field	Control passes to next field	pass
Vnumber	ID2879340554	Control passes to next field	Control passes to next field	pass
Vexp	06/04/2020	Control passes to next field	Control passes to next field	pass

TEST CASE NAME : BANK

Description	Input	Expected Output	Actual Output	Remarks
Bankname	SBI	Control passes to next field	Control passes to next field	pass
Accountno	SBI586	Control passes to next field	Control passes to next field	pass
Routingno	5894	Control passes to next field	Control passes to next field	pass

TEST CASE NAME : VEHICLE

Description	Input	Expected Output	Actual Output	Remarks
Make	Ford	Control passes to next field	Control passes to next field	pass
Model	Fiesta	Control passes to next field	Control passes to next field	pass
Purchasedate	08/06/2007	Control passes to next field	Control passes to next field	pass

Beginning mileage date	11/08/2007	Control passes to next field	Control passes to next field	pass
Ending mileage date	12/05/2008	Control passes to next field	Control passes to next field	pass

TEST CASE NAME : CHANGE PASSWORD

Description	Input	Expected Output	Actual Output	Remarks
Old password	*****	Control passes to next field	Control passes to next field	pass
New password	***	Control passes to next field	Control passes to next field	pass
Confirm Password	***	Control passes to next field	Control passes to next field	pass

TEST CASE NAME : PAYMENT

Description	Input	Expected Output	Actual Output	Remarks
Bank number	SBI435	Control passes to next field	Control passes to next field	pass
Debit/Credit number	JI259	Control passes to next field	Control passes to next field	pass
Validity Date	05/09/2010	Control passes to next field	Control passes to next field	pass

CONCLUSIONS AND RECOMMENDATIONS

The entire project has been developed and deployed as per the requirements stated by the user, it is found to be bug free as per the testing standards that are implemented. Any specification-untraced errors will be concentrated in the coming versions, which are planned to be developed in near future. The system at present does not take care off the money payment methods, as the consolidated constructs need SSL standards and are critically to be initiated in the first face, the application of the credit card transactions is applied as a developmental phase in the coming days. The system needs more elaborative technicality for its inception and evolution.

BIBLIOGRAPHY

References for the Project Development Were Taken From the following Books and Web Sites.

SQL Server

Mastering SQL Server 2000 by Gunderloy,Jorden BPB

Publications

Beginning SQL Server 2000 by Thearon Willis wrox publications

C# .NET

Programming Visual Basic .NET, Mircrosoft Press

C# .NET by Mc Donald, Microsoft Press