|  |
| --- |
| **Subodh Koparde** |
| **Enterprise Cloud Architect**  **Transforming Fortune 500 Enterprises with Modern Technologies** |
|  |  |

**Summary:**

* 27 years of IT industry experience in Microsoft and Java technologies.
  + 12+ years’ experience with SharePoint Architecture, Development and Administration.
  + 3+ years of architecting experience on cloud platforms such as AWS & Azure.
* Worked for Fortune 500 financial clients such as Freddie Mac, Towers Watson, Merryll Lynch(BofA), Chase Financials, Fortigent (Financial Services Company), Rydex Funds etc.
* Successfully designed and delivered secure cloud solutions and earned a reputation for designing cloud deployment strategies that mitigate risk while meeting infrastructure, employee, customer and budgetary needs.
* Successful with Integration architecture of discrete sub systems.
* Security Architecture using various risk mitigation planning.
* Excellent communication, interpersonal, analytical and presentation skills.

**Skills & Experience:**

* Azure Cloud, IaaS, PaaS, SaaS
* Azure Storage, Functions, Redis Cache, Databricks
* Docker, Kubernetes, Containers
* JQuery, AngularJS, BootStrap
* SharePoint 201x, Online
* SharePoint Apps, SPFx
* C# 5.0, Visual Studio .Net
* SQL Server 2010/2008R2/2008
* HTML5, CSS3, Javascript
* Agile Development, DevOps, Azure DevOps

**Certifications:**

* Microsoft Certified Technology Specialist: SharePoint 2010 Configuration (Code 70-667)
* Microsoft Windows SharePoint Services 3.0 – Application Development (Code 70-541.)
* Developing Applications with C# and ASP.Net. (Code 70-315. First of the four in MCSD.)
* Developing XML Web Services and Server Components (Code 70-320 Second of four)
* Sun Certification in Java 2

**Education:**

* Bachelor of Engineering in Computer Science and Engineering 1995
* Kuvempu University - India

**Professional Experience\**

**Solution Architect**

**Deloitte Aug 2022 to Date**

**Auditing tool**

This application is built for multi region Azure cloud. It is a highly scalable microservices application. It is integrated with CosmosDB, SQL db and Blob Storage. Service bus is used for making application easy to scale.

As Solution Architect I proposed performance improvements using Azure Workbooks. I implemented the API Management Gateway to interface with the front end. I implemented Circuit Breaker pattern to manage the calls to Microosft Graph API.

**Environment: Azure, C#, .Net, Angular 12, Azure Application Gateway, API Management Gateway, Cosmos DBStorage Qs, Blob Storage**

**SES Mar 2021 to Jul 2022**

**Capacity Planning Application**

SES Capacity Planning Application is Azure cloud native Microseevices application implemented using Azure FunctionApps, Azure Stoarage and Azure Web Application. The front end is a VueJs application that interacts with the azure function apps using Messaging protocol for easy scaling. Storage Queues are used for call routing. Blob storage is used for NoSQL data storage.

As an Architect I created the POC and demonstrated it to stakeholders. I assembled a team of developers and mentored them to implement the project.

**Environment: Azure, C#, .Net, Angular 12, VueJs, Azure Application Gateway, Storage Qs, Blob Storage**

**PPL Oct 2020 to Mar 2021**

**Content Classification using Azure Cognitive Services** (Forms)

A large set of Documents in PPL are unclassified. It is challenging to find out metadata of each document. Azure Cognitive services offers Forms Recognition that can be used to extract the content and metadata. This application used the Forms Recognition to identify sensitive information and classify it as sensitive.

As an architect, I developed the architecture of the application in multiple phases to address evolving requirements. Interfaced with Logic Apps and connectors to SharePoint and other subsystems.

**Environment: Azure Cognitive Services, Azure Logic Apps, Azure Storage, C#, .Net, Javascript ect**

**Vision Works April 2020 to Sep 2020**

**Point of Sale**

**(Remote) Austin TX**

Vision Works point of sale application is a cloud application built on AWS. It needs back end integration with on premises servers such as SQL Servers and DB2. The application spreads as multi cloud solution using HATEOAS Rest API. As part of development team, I built 4 Azure Functions that take care of Real Time Claims, Eligibility and Order Processing. These Azure functions are exposed using API Management gateway. I built a Azure Storage Que to manage the incoming requests. For On Premises backends, I used VNet integration for most back ends. Also used Azure relay for on critical backend.

Built pipelines using Azure DevOps. Set up the Repos to store code and implemented CI CD.

I used Automations to provision the Resource Groups and compute resources using ARM templates, CLI and Powershell

**Environment: Azure, C# .Net, SQL Server, Elastic Pools, Azure Application Gateway, Storage Tables, Queues. Azure DevOps**

**Mariner Finance Jan 2020 to March 2020**

**Internal Auditing System**

**(Remote) White Marsh MD**

Mariner Finance is a retail banking company where branches need to be audited periodically. The auditing division needs a robust application for its associates to use to record efficiencies of the branches.

This application is built with VueJS, ASP.Net Core, CosmosDB and SQL Dbs. I built this Single Page Application (SPA) using modern development principles such as DevOps and microservices. I developed VueX components, VueRouter etc. The services were exposed using a Application Gateway class and Service classes.

**Environment: VueJS, C#, ASP.Net MVC, RavenDB, SQL Server 2017, GitHub, Jira.**

**Freddie Mac, McLean VA June 2017 to Dec 2019**

Enterprise Cloud Architect

**Azure PaaS Readiness**

I built reference architecture for Azure PaaS, for Credit Monitoring application. I used Microservices architecture to build this application using Azure Functions and Azure Application Gateway, SOA, DevOps principles to build this application using various Microservice ASP.Net MVC application for Credit Risk application. I also deployed the microservices using Docker and Kubernetes cluster on Azure. Also, Azure Functions were used to host some function

I used AuzreAD, OpenID, OAuth for authentication and authorization.

**Environment: Azure PaaS, C#, ASP.Net MVC, CosmosDB, Azure Functions, Application Gateway, Event Grid, Logc Apps, SQL Server, GitHub, Azure, HTML 5, Bootstrap, WebPack, Babel, Typescript**

**SharePoint Development for modern day**

Office 365 Modern Applications Readiness

I built SharePoint Pages and webparts using SPFx. I used custom and SP REST API. I used Node.js, Visual Studio Code. Deployed webparts to SharePoint online. Created Microservices using Azure Functions. Created Logic Apps to distribute notifications to the intranet users.

Enforced data security using AKS. Encryption was used at rest and in transit. Used client certificates and disk encryptions. Evaluated 25 block security risks and mitigation plans.

I evaluated the technological and architectural risks for this solution. Since the roadmap of related technologies would change over the next few years, I factored in alternative solutions.

**Environment: SharePoint Fx, RactJS, NodeJs, NPM, Yeoman, C#, ASP.Net MVC, CosmosDB, Azure Functions, Application Gateway, Event Grid, Logc Apps, SQL Server, GitHub, Azure, HTML 5, Bootstrap, WebPack, Babel, Typescript**

**BlitzDocs ECM enhancements**.

 BlitzDocs is SaaS service that provides ECM Capabilities. It lacks many features required by the business division. The enhancement project downloads the documents nightly and syncs them periodically. The Web Application interface enables users to update various aspects of the documents downloaded. The documents will be synced to keep them consistent.

As senior architect and developer, I created the application POC using user requirements. Once it was accepted, a full-blown project was initiated with developers, testers and BAs on board. Additional Features were added.

I used my C#, ASP.Net expertise to build the application. I used APS.Net MVC structure. I built Web Services layer. Continuous Integration was implemented, and DevOps was used for regular updates to the application. This has been successful product within the division.

I used the Modern Delivery methodologies. DevOps principles were applied to this project which made CI/CD an important aspect. I used Docker container model to deploy application into Microsoft Azure.

**EnvironmentC#, ASP.Net MVC, GitHub, Azure, HTML 5, Bootstrap, WebPack, Babel, Typescript, CosmosDB, RavenDB, SQL DB**

**Wahington Gas Light Company, Springfield VA May 2016 to June 2017**

**SME/SharePoint Architect**

As a consultant .Net SME, I joined the project in critical phase. The previous launch had issues and performance was a problem. I established baselines, performance metrics and other requirements using AppDynamics to measure response times.

I developed WCF services using c#. I developed a model application using C#, ASP.Net, WCF to mimic the SharePoint 2010 implementation to tune the WCF services. I also built custom load generator using C# to produce 100+ RPM of load on the application. I implemented NetPipeBinding instead of NetTcpBinding.

This helped a lot in isolating and fine tuning Application Performance.

I also manipulated Throttling settings to optimize performance. I benchmarked the performance and set up operating guide lines, SLA for the Production Support team.

**Environment: Asp.Net, WCF 4.0, Ninject, AppDynaics, SharePoint Designer 2013, XML, XSLT, HTML, CSS, JavaScript, JSON**

**SUS LLC Jan 2016 to May 2016**

**Architect**

I extended existing ASP.Net application to MVC Structure converting user controls into WebParts. I migrated ASPX pages to site pages and built WSPs and additional RESTful WCF Services using C# services to bridge some of data requirements.

The client wanted to implement the SharePoint App to easily distribute the application. I made two separate applications. One for SharePoint hosted App and another one as IIS hosted app. I used AngularJS 2.0 to build both. The application needed to be easily morph to any device. I used Responsive Design principles. I used Bootstrap library for it.

I also created PowerShell scripts to deploy the changes to the SharePoint instances at Dev, QA and Prod. The scripts were instrumental in controlled propagation of changes.

**Environment: SharePoint 2013, SharePoint Online, SharePoint App Store, WCF, AngularJS, Bootstrap**

**Title: Lead Developer**

**Vana Solutions October 2015 to Jan 2016.**

**CIMS**

This is SharePoint application with a Silverlight UI. The UI is interfaced with SharePoint using WCF layer of services to facilitate and encapsulate the SharePoint related idiosyncrasies. SharePoint sub-webs and document libraries morph into entities and sub-entities in the UI.

I set up the Content Type Hub inside Managed Metadata Services for this project. I developed several Content Types and Document Templates. I created a Feature and implemented Feature Receiver class to provision all artifacts needed for the application to work. The Receiver instantiate sites, sub sites, document libraries, insert base line documents and list items. **Environment: Silverlight, WCF, SharePoint 2013 on prem, SharePoint Designer 2013, XML, XSLT, HTML, CSS, JavaScript, JSON**

**Title: Enterprise Architect**

**Stora Enso (Remote Location: Stockholm Sweden). August 2014 to October 2015**

**WeShare**

This is an Intranet SharePoint 2013 on prem version. I configured Kerberos authentication for this. Search Service Application is major part of this farm. I configured 4 dedicated application servers for search. Also, I configured SQL Mirroring as DR plan. The application uses SharePoint social features. I configured Activity Feeds to drive the web part to show latest changes.

Performance is of key importance to this huge SharePoint application. The content database size is 4TB. I incorporated performance counters locally and also using SCOM server to ensure performance.

I overlooked and mentored developers and junior admins to adhere SharePoint best practices. When I took over the project it was suffering from poor performance. I redid the dual farm architecture into stage-prod structure to handle the load better. This also made deployment an easier task.

I have used various JavaScript libraries to improve client side experience on this application. I have used JQuery extensively. Also, for Responsive design, I have used Bootstrap, MeanMenu etc.

The company uses two types of workflows. Simple workflows built using Nintex tools were used frequently. I developed some of the complex workflows using Nintex. This reduced a lot of manual work in the organization.

**WeShare China**

I built stage and test farms for this application and used them to upgrade the farm to latest CU. Implemented shared MMS with main WeShare application

**Dotcom (WWW site)**

I created this site for anonymous access. I architected the search service application to better handle the load of the server. I rebuilt the Test farm and updated it to latest CU.

I used client side scripting (JSOM) to implement submitting of user input into a SharePoint list. I used **JSON** to serialize and de-serialize.

I used **TFS** (Team Foundation Server) as code repository and also for change management.

**Environment: SharePoint 2013 on prem, Kerberos, ADFS, Bootstrap, JQuery, AngularJS, SharePoint Designer 2013, Nintex Workflows, PowerShell for Office 365, XML, XSLT, HTML, CSS, JavaScript, JSON**

**Title: Architect / Lead Developer**

**Voice – (Remote Location: Kentucky) June 2014 to July 2014**

This is an Azure/Office 365 using E3 business plan. This is built to customize Community template to brand it and include custom lists and custom document libraries.

I created a custom master template using SharePoint Designer. I created scripts using PowerShell and executed using O365 PowerShell plugin. These scripts created the custom lists and libraries. I also branded the search site of this site.

**Environment: Microsoft Azure, Office 365, SharePoint Online, SharePoint Designer 2013, PowerShell for Office 365, HTML, CSS, Javascript, XSLT, HTML**

**Title: Chief Architect**

**Wisconsin Interactive Network (WIN) Aug 2013 to Jun 2014**

**Madison WI**

As part of WIN team and as a single SharePoint resource I architected the solution for the revitalized site of http://www.wisconsin.gov for state of Wisconsin. I worked with two Web designers to produce this website. This was set up in a record breaking time.

I created project proposal documents, system architecture diagrams and server network diagrams so that client can understand what we are building. I used Publishing Features and Content Deployment Paths as part of my solution. Multiple site collections are hosted as Host Named Site Collections

**Environment: SharePoint 2013, SQL Server 2008 R2, Net Framework 4.0, Visual Studio 2013, C#, ADO.Net, SQL Server 2008, HTML, CSS, Javascript**

**Title: Senior Web Developer / Sr. SharePoint Architect**

**SHRM –Alexandria, VA Aug 2010 to Aug 2013**

**SHRM website and SHRNet intranet site both hosted on SharePoint Server**

SHRM (Society for Human Resource Management) is a professional organization for HR people. It manages and publishes a huge amount of Web Content. This content is built and stored in SharePoint Server.

I built 4 SharePoint farms to cater to needs of developers, designers, QA and staging. I automated the binary deployment to each farm using SVN for code version control and TeamCity for build server. This put end to frequent unknown code errors on the production farms. I also configured other .Net applications to use build server.

Searching content was a challenge and often users complained about not finding content. SharePoint’s search engine was not sufficient to handle the load of general public which is close to a million hits per day. Hence, installed and integrated Google Search Appliance with the SharePoint server. Now, the search works faster yet to the users the change is very transparent.

I developed webparts to interface with shopping cart based e-commerce application to purchase day passes for the site.

I developed custom content-deployment time job to selectively migrate content from one farm to other farm. This helped designers to make changes to their farm and let the change go to QA and then to Staging and to Production.

I managed and improvised an ASP.Net based mirror site of SharePoint production server that can be used to handle the user load if the production needs to be down for maintenance. I also built a ASP.Net MVC application to administer the backups.

I used Google Web Analytics and Business Intelligence to advice the company on best use of Search engine features and best practices in Information architecture on the public facing site.

I created a timer job to update the Taxonomy based report pages that help users discover all articles tagged with a taxonomy. This timer job was deployed using PowerShell scripts I created as alternative to creating SharePoint provisioning feature. This saved the extra binary foot print on the server.

I have built several web services using **WCF, SOAP, JSON** to be consumed by the SharePoint server as well as by other systems that are written in C# as well as java languages.

I have also ported the content of 28 gigs from MOSS 2007b to SharePoint 2010 platform and tested the performance. Further migrated to a SharePoint 2013 farm.

I used **TFS** (Team Foundation Server) as code repository and also for change management.

**Microsoft Visio** was used as part of the Office applications via OWA (Office Web Apps). I integrated it with the application.

**Environment: SharePoint 2013, SharePoint 2010, MOSS 2007, Net Framework 4.0, Visual Studio 2010, ASP.Net, MVC, C#, ADO.Net, SQL Server 2008, Unit Testing, Code Analysis, Final Builder, HTML, XHTML, CSS, Javascript, WCF, Web Service, SOAP, XSLT, XML**

**TITLE: Sr. Developer / Architect**

**Client: Watson Wyatt, Arlington VA Feb 2009 to July 2010**

**OnePlace and DataLink**

OnePlace is a collaboration site for clients and internal consultants. I designed and developed two flavors of site templates for this, Meeting Space and Work Space.

One of the features of the Work Space is facility to add, search and display Topical Issues from across multiple site collections. I customized the search crawler to crawl custom metadata and index. This enables clients to search specific documents in a metadata category. I created custom actions to add custom settings to the list settings. This action upgrades the list to add additional columns needed and a proxy list to perform full text search on. The proxy list also contains site columns created using the SPFeature. The feature contains all these components and a web part to display topical issues

I created BDC application to integrate with EVAL database and display folder name field and file count column in the DataLink application.

I added Custom Actions using **CAML** to Document Library’s drop down menu to link it to a specific EVAL folder. This action invokes email notification page to notify consultant that a file is ready for review.

I created a custom Purge action and associated it with the document library’s Policy file. This was necessary because, after expiry of document we wanted to purge it rather than move it to recycle bin.

I manipulated the Policy file on the document library and on site collection to change the default behavior of recycling when deleted. I also created custom permissions and permission groups to assign to specific user group.

I designed and developed DataLink application that is used by external clients to upload documents securely into Eval System. Each client has exclusive access to a separate Document Library. The target Eval folder id is stored on the sub folder of the document library. The list event handler propagates the file uploaded to eval system.

I created Document Information policy to provide different level of permissions to clients, internal consultants and administrators.

I created SharePoint Job to auto notify file uploaded to DataLink library before the file is moved to recycle bin. Another Job auto uploads the file to eval folder before it is deleted.

I developed web services hosted inside SharePoint site. These services would perform privileged tasks such as creating document library or purging document.

I used **TFS** (Team Foundation Server) as code repository and also for change management.

**Environment: MOSS 2007, Net Framework 3.5, Visual Studio 2008, Visual Basic.Net, ADO.Net, SQL Server 2005, Unit Testing, Code Analysis, Final Builder, HTML, CSS, Javascript**

***TITLE: SharePoint Developer***

**Fortigent, Rockville MD Nov 2007 to Feb 2009**

**Advisor Portal – using SharePoint 3.0**

**Financial Portfolio Research and Management Tool for Advisors**

This is a major product for Fortigent. The company primarily is a back office for the financial advisor firms that cater to very high net worth individuals. Typical clients are on net worth 10 million or more. This web application developed using SharePoint technology uses two types of WebParts that can be personalized and customized for each advisor. The simple ones are customized SharePoint out of the box web parts. The others are home grown WebParts to integrate with legacy applications and database.

As an application developer, I developed the authentication provider to the SharePoint so that a single authentication token can be shared by all sub systems of the company. The authorization part of it would use a role based security model which ensures that one user can not access data of another user by any means. I also developed an HTTPModule to authenticate calls to SSRS server and to decrypt account ids in query string.

I developed 15 to 20 SSRS reports using RDL (Report Definition Language). These reports were complex in structure but performed better than Crystal Report counterparts.

I developed a report viewer page that uses **AJAX** to obtain a URL to SSRS report and create DOM objects at client side to show IFrame with the report. I also developed 14 WebParts related to Top10Investments, WeeklyUpdates, Client Accounts etc. and extended News list and WebPart of SharePoint to include icons. I authored ONET.xml for site template creation.

I used WSPBuilder tool to package and deploy the solutions. Being an open source application, it required certain amount of taming to get it to do what we wanted to achieve with our deployment style.

I used **LINQ to SQL** and **LINQ to XML** as part of querying database and querying custom config file for a report viewer application.

**Environment: MOSS 2007, SharePoint WSS 3.0, .Net Framework 3.5, Visual Studio 2008, C# 3.5, ASP .Net, ADO.Net, SQL Server 2005, SSRS, Unit Testing, Code Analysis, WSPBuilder, HTML, CSS, Javascript, AJAX**

***TITLE: .Net Developer:***

**Fortigent, Rockville MD, Apr. 2006 to Nov 2007**

**ARI (Axys Replacement Initiative)**

**Financial Portfolio Research and Management Tool for Reconciliation Analysts**

This is a major tool used by Fortigent to capture, research and report the portfolios of various wealthy high net worth clients – both individual and corporate. It replaced a legacy OTS application. This application was built to perform. Speed of data entry, intelligent rules and shortcuts and quick reports are the selling features this application.

As a lead developer, I used MVC pattern to design the application. I also used Proxy, Flyweight and Modified Singleton to make the application perform faster. I decided to use modified Singleton pattern to cache the business data. I built a generic application framework to provide basic application features. This framework was then extended to the ARI application. I used .Net Memory profiler to find out and resolve memory build up issues. I performed stress test and load tests on the application using a home grown tool. I extended the tool to better suite needs of this application. I tuned up various stored procedures in SQL Server 2005 database.

We used agile methodology as our process. Frequent code reviews and code refactoring was key to bring up a well understood and clean code base. This also helped in resolving performance bottlenecks. I setup and streamlined use of Cruise control. I integrated it with Visual Studio 2005 and Borland Starteam.

As part of data download module, I was responsible for evaluating BizTalk server 2006. I created an in-port, out-port to periodically download data from custodian interfaces and save it to the daily transactions table. This piece was evaluated but the cost of the technology was too high to incorporate it. Hence we built a custom solution for this.

I used **TFS** (Team Foundation Server) as code repository and also for change management.

I also developed SSIS packages to import data from external Data Sources into SQL Dbs.

**Environment: VB.Net, WinForms, .Net, ADO.Net, SQL Server 2005, SSIS, SSRS, Unit Testing, Code Analysis, CruiseControl, BizTalk Server 2006**

**EARLIER PROJECTS IN BRIEF**

**Collegiate Funding Services, LLC, Fredericksburg VA,** Aug. 2005 to Mar. 2006

***TITLE: Lead Developer:***

Project: SPIDR, Call Center Tool, SPIDRWEB

Loan application for students, Email Broker Service, Dashboard – DotNetNuke

Environment: C#, WinForms, .Net Remoting, XML Web Services, XML, XSL, ADO.Net, Sql Server 2000, NUnit**,** HTML, CSS, Javascript, ORM

**Baltimore Gas and Electric (BGE) Baltimore, MD** March 2005 to August 2005  
***TITLE: Lead Developer:***

DSIS – Intranet Application  
Environment: C#, ASP.Net, XML Web Services, XML, XSL, ADO.Net, Oracle 9i, NUnit, Rational Suite Dev Studio (Clear Case, Clear Quest, Requisite Pro, Rose, etc) UML, OOAD HTML, CSS, Javascript

**Rydex Funds - Rockville, MD** July 2004 to March 2005  
***TITLE: Lead Developer:***Rydex Application Console for plug-ins:

Environment: C#, .Net Windows, Windows Services, BizTalk, XML, XSL, ADO.Net, SQL Server, NUnit, Rational Suite Dev Studio (Clear Case, Clear Quest, Requisite Pro, Rose, etc) UML, OOAD, Log4Net , **NDoc**

**Merrill Lynch Wealth Management Online System** March 2003 to July 2004  
**Client: Thomson Financials - Rockville, MD   
*TITLE: Senior Developer***Environment: ASP.Net, C#, Web Services, ADO.Net, MS-CMS, SQL Server, NUnit, Rational Suite Dev Studio (Clear Case, Clear Quest, Requisite Pro, Rose, etc) UML, OOAD, Log4Net, HTML, CSS, Javascript

**Lexis-Nexis, Bethesda, MD** Aug.2001 to Feb 2003

Environment: C#, ASP.Net, Visual Studio.Net, Oracle 8, XML, Parsers, ADO.Net, RUP, Rational Rose, UML, OOD, Design Patterns,

**ONEOK Inc., Tulsa, Oklahoma**

PGAS Data TransfersJan. 2001 to Aug. 2001

**Team Lead**

Environment: Java, JBuilder, Swing, JDBC, MS ACCESS 97, SQL Server 6.5/7.0, Oracle 8.1.4, UML, Rational Rose, RUP

**El Camino Resources Ltd., California** Feb. 1999 to Dec. 2000

**Team Lead**

GENESIS – A Web based Sales Force Automation System

Environment: VB6, IIS, MTS 2.0, COM, DCOM, Java, Oracle 8.1.4, ADO, XML,

Active Directory, Rational Rose, UML, RUP.

**EDOCS – Document Generation and Management System**

**Senior Developer**

Environment: Visual Basic 6.0, VBA, MTS 2.0, MS ACCESS, ADO, PL/SQL, Oracle 7.3

**ICICI** July 1998 to Jan. 1999

ICISR: Integrated Client Information Storage & Retrieval System – Financial Inst.

**Senior Developer**

Environment: Visual Basic 5.0, Oracle 7.2, PL/SQL, SQL\*Loader

STPR - Online Interest Rates Management System – Financial Inst.

**Developer**

Environment: Oracle 7.2, SQL \*Net, Forms 4.5

CRIS: Credit Rating Information System – Financial Inst.

**Developer**

Environment: Visual Basic 5.0, Crystal Reports, PL/SQL, SQL\*Loader, Oracle 7.2

**Emirates Airlines** Sept. 1997 to June 1998

**Developer**

*DMACS-GUI 1.0 - Dynamic Multiple Access Check in System, for Airports*

**Client: Emirates Airlines**

Environment: Visual Basic 5.0, Crystal Reports, PL/SQL, SQL\*Loader, Oracle 7.0

Document Processing System 2.0**Jan.1996 to Aug. 1997**

**Developer**

**Unit Trust of India**

Environment: Visual C++ 3.0, Oracle 7.2, Windows 95, PL/SQL