# Introduction

## Background

**Normal internet browser**

The popular internet browsers like Chrome, Firefox, and Internet Explorer request the internet resources directly. Their goal is fast internet browsing, that’s why they request webpages directly. First user enters the required website address, called Uniform Resource Identifier. A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. Then the browser do all the works of getting the requested page and displaying it on the screen correctly, this is called parsing and rendering. The process is very complex but happens very fast.

This type of browsers run on local computers. So all the work is done by the local computer. Also there is thing called HTTP compression. HTTP compression is a capability that can be built into web servers and web clients to make better use of available bandwidth by compressing the transmission packets. This saves some data usage and help in efficient use of bandwidth.

**Cloud**

In very simple terms. Cloud is network of computer servers which give service through internet connection. It is very reliable group of computer servers since there is multiple instances of servers running same task. This makes cloud more reliable. Cloud servers are distributed across the globe which makes the running service available from multiple location on Globe.

Cloud computation technology uses the internet as main structure. Cloud computing has become the biggest focus in the IT community. Cloud computing has aligned with market needs today. As various business clients are fighting with issues of extreme cost, complexity in logic and need for increasing speed, cloud technologies are looked upon as a vehicle for change.



Fig 1.1: Microsoft azure cloud servers

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). Cloud computing offers much more advantages than traditional application deployment models since it is built around scalable development resources, and involves less costly maintenance.

One of types of services provided by cloud is SaaS. It is abbreviations for Software as a service. In SaaS cloud service users are provided access to the application software and databases. SaaS allows a business the potential to reduce IT operational costs by outsourcing hardware and software maintenance and support to the cloud provider. This enables the business to reallocate IT operations costs away from hardware/software spending and personnel expenses, towards meeting other IT goals. In addition, with applications hosted centrally, updates can be released without the need for users to install new software. One drawback of SaaS is that the users' data are stored on the cloud provider’s server. As a result, there could be unauthorized access to the data.

**Windows 8.1 Store app**

Windows 8 is a version of Microsoft Windows (an operating system developed by Microsoft) for use on personal computers, including home and business desktops, laptops, tablets, and home theater PCs. Windows 8.1 is update for Windows 8 OS.

A Windows Store app is a new type of application that runs on Windows 8 running devices. Unlike traditional desktop apps, a Windows Store app has a single, chrome less window that fills the entire screen by default, so there are no distractions. Windows Store apps provide several new controls which are specially optimized for touch based devices.

## Problem Statement and Objectives

The project is trying to solve the problem of how to minimize the load on internet bandwidth of client computer while browsing the web pages. This project also tries to minimize the processing overhead on client side by moving most of processing to cloud side.

Also on volume based internet packages, this data usage savings can also save money. The project tries to move most of the overhead of the ping requests, data requests, and data upload / download to the cloud. In addition, the cloud also compress the data before sending to the client computer. Only the compressed result is downloaded to the client, so the client load is very much reduced to just downloading a page with one request.

## Two Projects

The full solution of the problem requires at least two sub projects. One is a cloud-side service or Web services project (which can be deployed to cloud) to process request and compress the webpages. Another is a client-side project, which decompress the compressed data from actual processing remote server.

In this case two projects are a WCF service (Windows Communication Foundation service) and Windows 8.1 store app. These are more detailed on later sections. The client side app is named ‘SurferLite’ and server side cloud service is named ‘surferlitedev’.

Basically, there are two options for building the Web services project: Windows Communication Foundation (WCF) or the Active Server Page in .NET (ASP.NET) Web API (Application Programming Interface), which is included with ASP.NET MVC4 (Model View Controller). Because exposing services via WCF is widely documented, for the scenario it is easy to use the more modern approach than the ASP.NET Web API brings to the table, truly embracing HTTP (Hyper-Text Transfer Protocol) concepts (URIs and verbs).

## Project Features

SurferLite, software from the project possesses the following features:

### Server side compression

To make the browsing faster, the server downloads the web page and then compresses it, thus the client browser downloads the compressed web pages from the server. The users have the capability to turn ON/OFF the compression as well as the images and scripts.

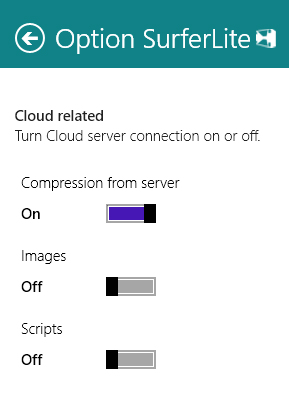


Fig 1.2: Compression option

### Bookmarks and History

Users can bookmark the web pages for viewing them later and the history of the navigations will also be stored.

### Other Basic browser functionalities:

The browser contains the basic functionalities like refreshing pages, stopping the navigation of the page, forward & backward buttons, new tabs and settings. The application provides the user with a user-friendly interface.

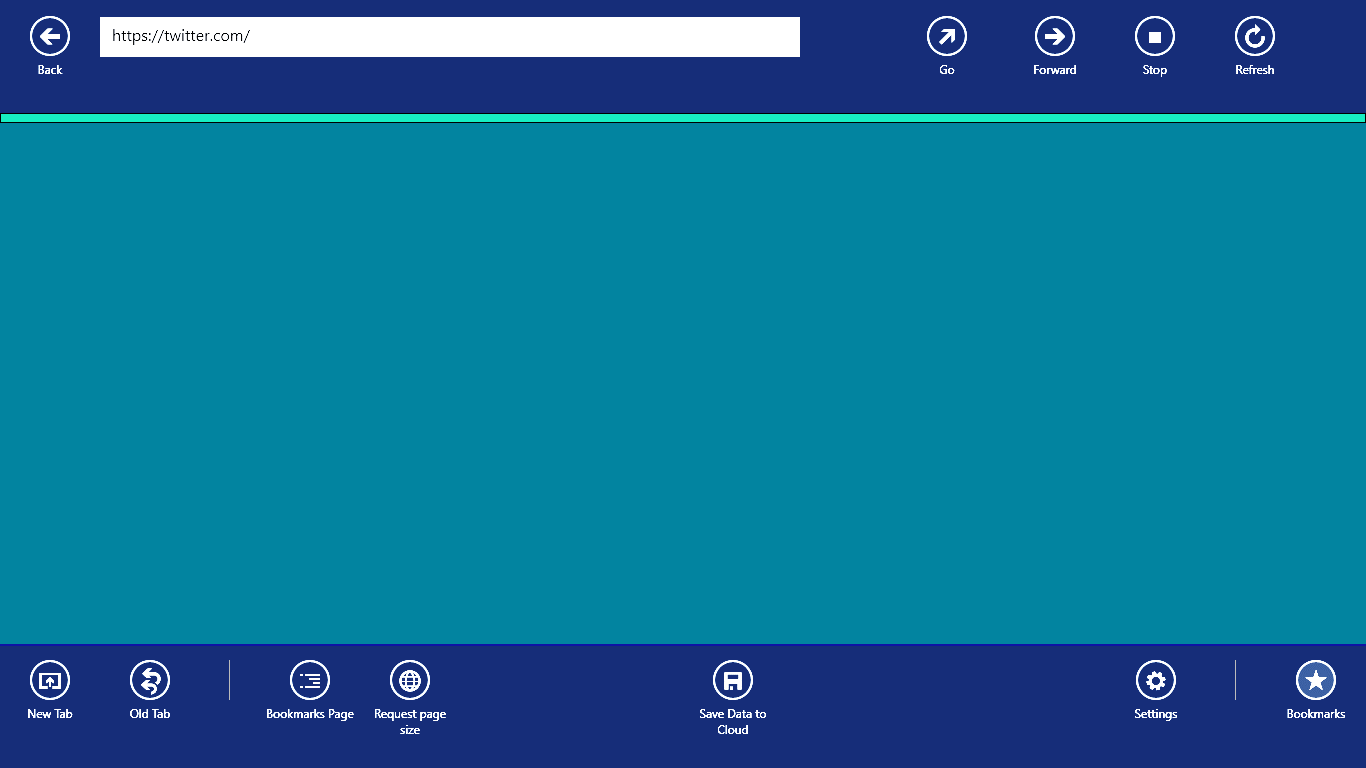


Fig 1.3: Browser interface

## System Requirement

**Windows 8.1 Preview Requirements**

1. Processor: 1 gigahertz (GHz) or faster
2. RAM: 1 gigabyte (GB) (32-bit) or 2 GB (64-bit)
3. Free hard disk space: 16 GB (32-bit) or 20 GB (64-bit)
4. Graphics card: Microsoft DirectX 9 graphics device with WDDM driver

**SurferLite app requirements**

1. Active Internet connection, a screen resolution of at least 1024 x 768, and a Microsoft account to sync settings across multiple devices.
2. Internet access (ISP fees might apply).

# Literature Review



## Cloud computing

Cloud computing is the next stage in the Internet's evolution, providing the means through which everything from computing power to computing infrastructure, applications, business processes to personal collaboration can be delivered to you as a service wherever and whenever you need. The “cloud” in cloud computing can be defined as the set of hardware, networks, storage, services, and interfaces that combine to deliver aspects of computing as a service. Cloud services include the delivery of software, infrastructure, and storage over the Internet (either as separate components or a complete platform) based on user demand.



Fig 2.1 Cloud computing

Cloud computing has four essential characteristics: elasticity and the ability to scale up and down, self-service provisioning and automatic de-provisioning, application programming interfaces (APIs), billing and metering of service usage in a pay-as-you-go model. This flexibility is what is attracting individuals and businesses to move to the cloud.

The world of the cloud has lots of participants:

* + The end user who doesn’t have to know anything about the underlying technology.
  + Business management who needs to take responsibility for the governance of data or services living in a cloud. Cloud service providers must provide a predictable and guaranteed service level and security to all their constituents.
  + The cloud service provider who is responsible for IT assets and maintenance.

## Services offered by the cloud computing

Cloud computing provides the following services to the users of cloud:

**Platform as a Service (PaaS)**

This is a platform or interface for the users or businesses for developing apps according to somebody’s particular necessities. It provides a computing platform and a solution stack as a service.

**Software as a Service (SaaS)**

This is a service where software apps are offered to businesses that are looking for it. The service is prompted by the service on demand viewpoint and the software & the associated data are centrally hosted on the cloud.

**Infrastructure as a Service (IaaS)**

Infrastructure as a Service is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis.

Other than the above mentioned services, developers also offer storage as a service or (StaaS), data as a service or (DaaS), security as a service or (SECaaS), desktop as a service (DaaS), Test environment as a service (TEaaS) and API as a service (APIaaS).

## Existence of similar system

### Opera Off-Road

* Off-Road mode uses the cloud-based compression technology from the Opera Mini browser. This has some advantages and some drawbacks.
* On slow and unreliable networks, Off-Road mode loads pages significantly faster than the normal browsing mode.
* Data usage is significantly reduced and can be adjusted by changing the image quality setting.
* On slower devices, page rendering and scrolling can be faster.
* Some dynamic website features might not work flawlessly in Off-Road mode due to its architecture.

### Skyfire web browser

Skyfire Web Browser is a mobile web browser which renders requested web page on a proprietary server and relays it to the browser on the end user's mobile phone which displays the content. Skyfire comprises two distinct generations of mobile browser technology. In Skyfire's first generation (1.x) browser, a web page is fully rendered by a server separate from the mobile device, similar to the operation of a thin client. This approach is also used by Opera Mini. Skyfire's second generation (2.x) browser employs a hybrid approach, using a conventional rendering of Web pages on the handheld device, but streaming video from Skyfire's servers.

## Windows Azure

Windows Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters. It provides both platform as a service (PaaS) and information as a service (IaaS) and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems. You pay only for the resources that you require and then scale them up and down at any time. And you don't have to own the hardware or supporting infrastructure to do this. If your business can leverage the platform to increase agility, lower costs, or lower risk, then Windows Azure is a good fit for your application.

## Team Foundation Server (TFS)

Visual Studio Team Foundation Server 2012 (TFS) is the collaboration platform at the core of Microsoft's application lifecycle management (ALM) solution. TFS supports agile development practices, multiple IDEs and platforms locally or in the cloud and gives you the tools you need to effectively manage software development projects throughout the IT lifecycle. It is a Microsoft product offering source control, data collection, data collection, reporting, and project tracking, and is intended for collaborative software development projects. It is available either as stand-alone software, or as the server side back end platform for Visual Studio Team System (VSTS).

## GitHub

GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub like other version control systems, manages and stores revisions of projects. Although it is mostly used for code, it can also be used to manage any type of file, such as Word documents or Final Cut projects. The site provides social networking functionality such as feeds, followers and the social network graph to display how developers work on their versions of a repository. GitHub is a social network for programmers. It is a hosted Git repository. It allows you to take part in collaboration by forking projects, sending and pulling requests, and monitoring development.

## Cloud related concepts

Cloud computing shares characteristics with:

1. **Autonomic computing:** Computer systems capable of self-management.
2. **Client–server model:** Client–server computing refers broadly to any distributed application that distinguishes between service providers (servers) and service requesters (clients).
3. **Grid computing:** "A form of distributed and parallel computing, whereby a 'super and virtual computer' is composed of a cluster of networked, loosely coupled computers acting in concert to perform very large tasks."
4. **Mainframe computer:** Powerful computers used mainly by large organizations for critical applications, typically bulk data processing such as census, industry and consumer statistics, police and secret intelligence services, enterprise resource planning, and financial transaction processing.
5. **Utility computing:** The "packaging of computing resources, such as computation and storage, as a metered service similar to a traditional public utility, such as electricity."
6. **Peer-to-peer:** Distributed architecture without the need for central coordination, with participants being at the same time both suppliers and consumers of resources (in contrast to the traditional client–server model).
7. **Cloud gaming:** Also known as on-demand gaming, this is a way of delivering games to computers. The gaming data will be stored in the provider's server, so that gaming will be independent of client computers used to play the game.

## 2.8 Windows 8.1

Windows 8.1 is the update of the Windows 8. The update is intended to address aspects of Windows 8 that were criticized by reviewers and early adopters (such as the removal of the Start button from the desktop interface), and will also incorporate additional improvements to various aspects of the operating system. Some of the features of Windows 8.1 are: Lock screen slideshow, evolving of start screen with more colors and backgrounds including some motion, different tile sizes, aggregated search, enhancement of apps such as Xbox Music & Mail, an enhanced windows store, a new internet explorer, better with a mouse and keyboard, a change to the start tip and start button, improvements to desktop and All programs and many more.

## 2.9 Technology used

### 2.9.1 XAML

Extensible Application Markup Language, or XAML, is an XML-based markup language developed by Microsoft. XAML is the language behind the visual presentation of an application that you develop in Microsoft Expression Blend, just as HTML is the language behind the visual presentation of a Web page. Anything that is created or implemented in XAML can be expressed using a more traditional .NET language, such as C# or Visual Basic.NET.

### 2.9.2 WCF

The Windows Communication Foundation (or WCF) is a runtime and a set of APIs application programming interface in the .NET Framework for building connected, service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application. An endpoint can be a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or word sent as XML, or as complex as a stream of binary data.

### 2.9.3 Microsoft Visual C# and .NET

Microsoft Visual C# is a programming environment used to create computer applications for the Microsoft Windows family of operating systems. It combines the C# language and the .NET Framework.

The .NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. It includes a large library and provides language interoperability (each language can use code written in other languages) across several programming languages. Programs written for the .NET Framework execute in a software environment (as contrasted to hardware environment), known as the Common Language Runtime (CLR), an application virtual machine that provides services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework.

# System Development

The application is developed using scrum which is one of the popular software development lifecycle. Scrum is a way for teams to work together to develop a product. Product development, using Scrum, occurs in small pieces, with each piece building upon previously created pieces. Building products one small piece at a time encourages creativity and enables teams to respond to feedback and change, to build exactly and only what is needed.

More specifically, Scrum is a simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create just enough structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge.

However, Scrum is much more than a simple framework. Scrum supports the need to be human at work: to belong, to learn, to do, to create and be creative, to grow, to improve, and to interact with other people. In other words, Scrum leverages the innate traits and characteristics in people to allow them to do great things together.

**How does Scrum work?**

Building complex products for customers is an inherently difficult task. Scrum provides structure to allow teams to deal with that difficulty. However, the fundamental process is incredibly simple, and at its core is governed by 3 primary roles.

1. Product Owners determine what needs to be built in the next 30 days or less.

2. Development Teams build what is needed in 30 days (or less), and then demonstrate what they have built. Based on this demonstration, the Product Owner determines what to build next.

3. Scrum Masters ensure this process happens as smoothly as possible, and continually help improve the process, the team and product being created.

While this is an incredibly simplified view of how Scrum works, it captures the essence of this highly productive approach for team collaboration and product development.

## 3.1 Project Management

The project is managed using OOAD and scrum. Scrum is facilitated by a Scrum Master, who is accountable for removing impediments to the ability of the team to deliver the sprint goal/deliverables. The Scrum Master is not the team leader, but acts as a buffer between the team and any distracting influences. The Scrum Master ensures that the Scrum process is used as intended. The Scrum Master is the enforcer of the rules of Scrum, often chairs key meetings, and challenges the team to improve. The role has also been referred to as servant-leader to reinforce these dual perspectives. The Scrum master differs from a Project Manager in that the latter may have people management responsibilities unrelated to the role of Scrum Master. The Scrum Master role excludes any such additional people responsibilities.

## 3.2 System Analysis

In this phase, requirements and system analysis is done. Gathering of all the features, which the application must or required to do. Initially, the core focus was on the functional requirements of the application i.e. what are the features/capabilities of the application are. Then the focus was on the non-functional requirements of the application such as the documentations, performance, response time, etc. of the application. In this phase, core focus was on the application, viewing functionalities in terms of the system.

### 3.2.1 User Stories

In software development and product management, a user story is one or more sentences in the everyday or business language of the end user or user of a system that captures what a user does or needs to do as part of his or her job function. User stories are used with agile software development methodologies as the basis for defining the functions a business system must provide, and to facilitate requirements management. It captures the 'who', 'what' and 'why' of a requirement in a simple, concise way, often limited in detail by what can be hand-written on a small paper note card.

User stories are written by or for the business user as that user's primary way to influence the functionality of the system being developed. User stories may also be written by developers to express non-functional requirements (security, performance, quality, etc.), though primarily it is the task of a product manager to ensure user stories are captured. User stories are a quick way of handling customer requirements without having to create formalized requirement documents and without performing administrative tasks related to maintaining them. The intention of the user story is to be able to respond faster and with less overhead to rapidly changing real-world requirements.

The requirements of the application, which was collected in the previous stage, is then used in this stage. Now, the focus was changed from the feature focus, to the user focus i.e. how does the user accomplish a particular goal from the application? To describe the application, in the perspective of users, plain English was used in a non-technical way as possible. The format used in this phase was the user story format.

User story format: As a (type of user), I want (goal), so that (REASON).

## 3.3 System Design

After writing the user stories, identification of the most important objects were done. In this phase, finally, object oriented construction of the application was started. Without describing the system in the form of user stories, next phase is not entered. Now, scrum is used. Steps involved in scrum:-

1. The product backlog is created by collecting all the user stories. It is like a wish list of the customer.

2. Release planning: The most important user stories are selected from the product backlog and kept in a release backlog. The user stories in the release backlog are then prioritized and estimated the amount of work for each item. Larger user stories are broken into smaller pieces. The work is estimated in hours.

3. Estimate work: The total hours in the release backlog is the rough estimation of the work.

4. Sprint planning: From the release backlog, several sprints are made. Each sprint is a manageable chunk of the application. A sprint generally ranges of 2 days to 30 days. The release backlog is changed to several sprint backlog in this stage.

6. Now all the necessary classes needed are identified in the application using the user stories.

**Client side class diagram:**

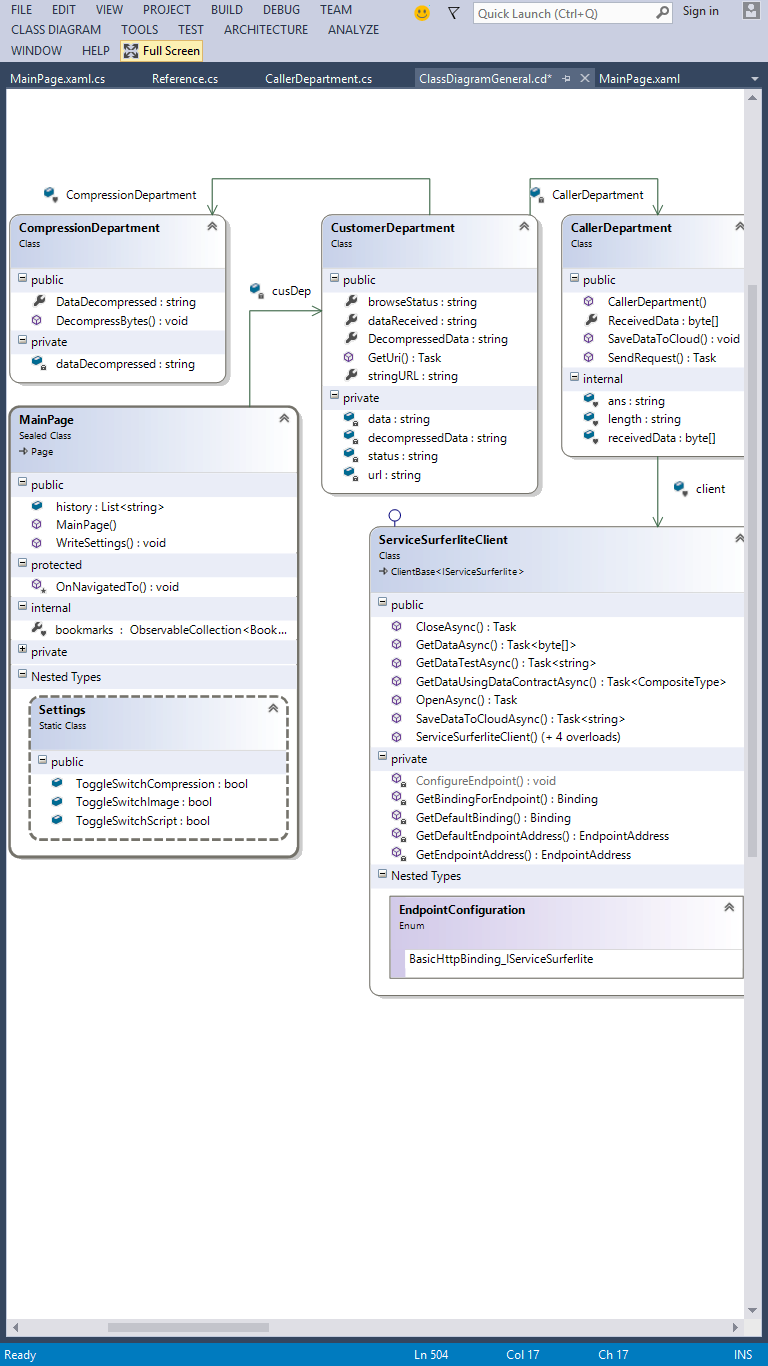


Fig 3.1 Client Side Class diagram

**Server side class diagram:**

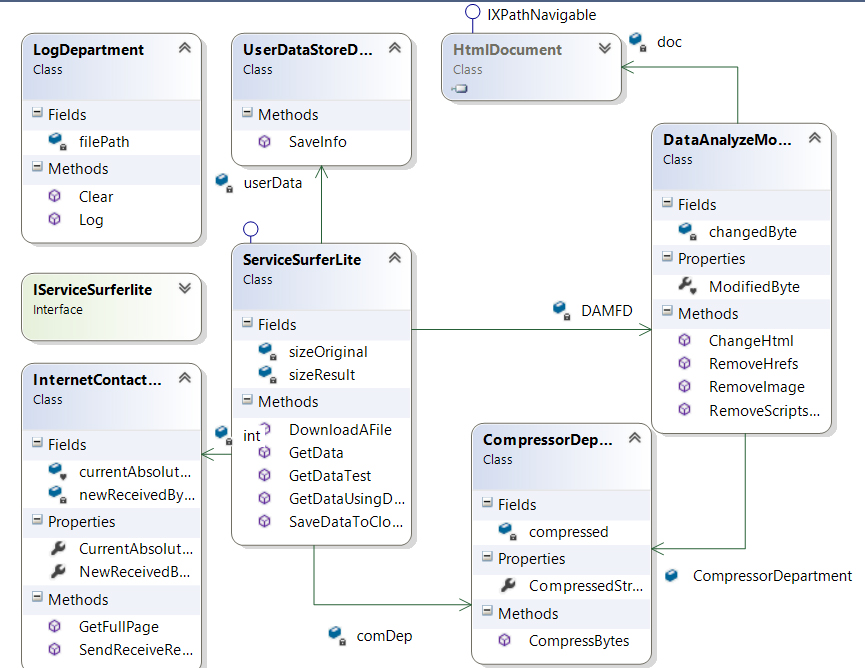


Fig 3.2 Server side class diagram

## 3.4 System testing / Implementation

After designing the classes code was directly written for the application. System testing of the application is done on the basis of hit and trial method. The syntactical errors in the code is resolved with the help of or IDE. Git hub was used to track the changes in the codes and for collaboration between project members. The project can be found on

http://github.com/KECProjectGroupURMS/SurferLiteDev.

# Results

The project is trying to solve the problem of How to save bandwidth of internet connection while browsing the web pages. The minimization of network load on client computers help in saving of data usage as well as processing power.

On volume based internet packages also, this data usage savings can also save money. The project tries to move most of the ping requests, data requests, and data upload / download to the server side, and only the result is downloaded to the client. The client load is very much reduced to just downloading a page with one request.

The final output of the project was two sub-projects:



## 4.1 Client side Windows 8.1 store app

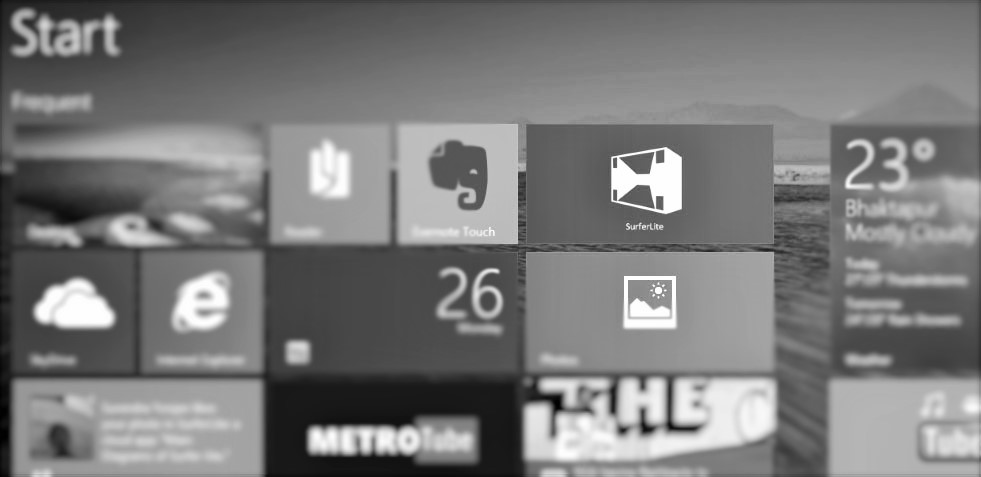


Fig 4.1 SurferLite on Windows 8.1 Preview start screen

This client side Windows 8.1 store app is simple browsing interface to the server Azure Cloud Service. This Client sends the browsing request to the Cloud service. The UI of the app look like shown in the figure.

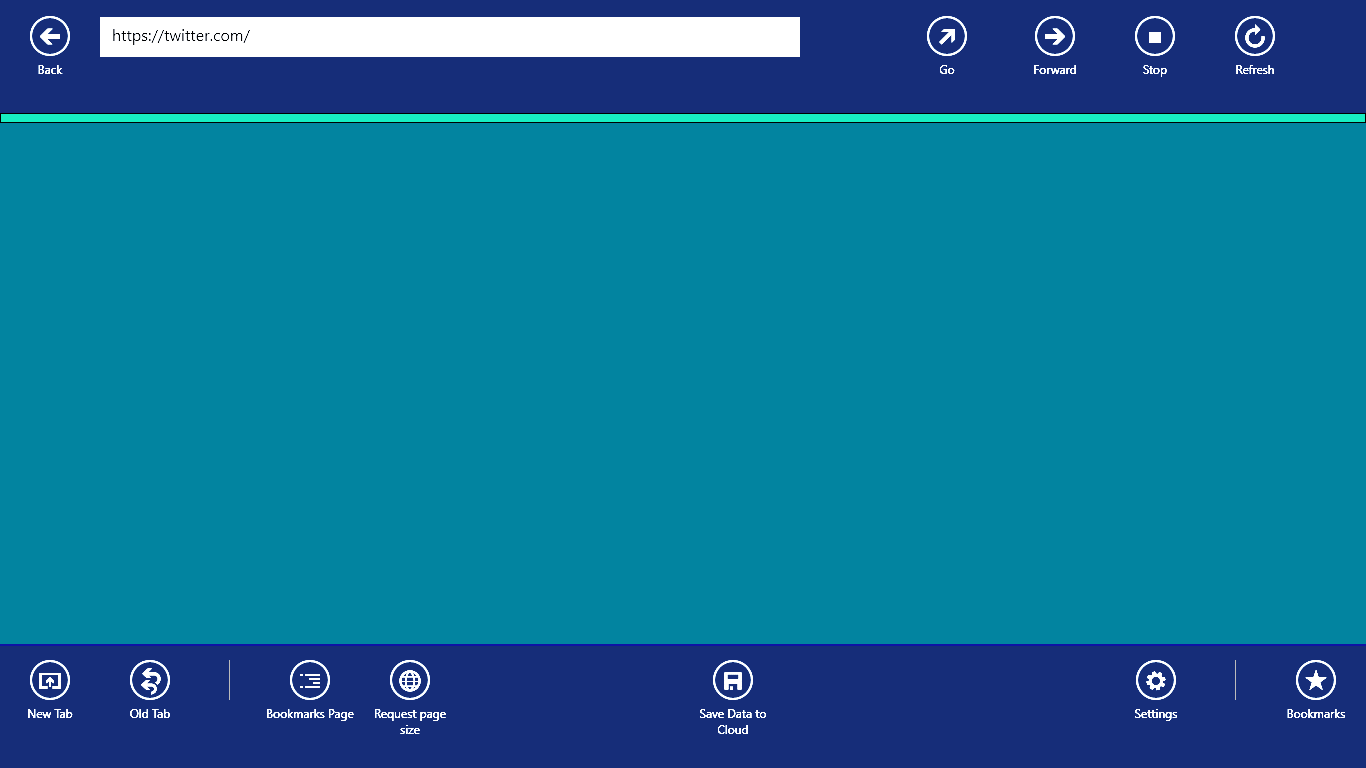


Fig 4.2 the main screen of the client SurferLite app.

This application has features like Bookmarks, tabs, settings to enable or disable the compression etc. The client is mainly to decompresses and display page sent by azure cloud service. For decompression GZip is used and to displaying page the IE11 (Internet Explorer 11) renderer was used.

Also the client app runs on windows 8.1 and the settings are synced on all computers. The Roaming settings are uploaded to cloud on Microsoft’s online account.

When the compression feature is turned off, the app works like normal browser. On this mode all the requests are directly sent to the target host not the azure service. So data usage is normal. But when compression is turned on all requests pass through the cloud service surferlitedev.

The UI (User Interface) is user friendly with known controls and icons. So using the app is easy. Also the app supports live tiles that is the update of tile shown in start screen by data from app.

## 4.2 Server side Azure Cloud Service

The server side cloud service is named ‘surferlitedev’. This accepts the request from client app and return the resulting webpage to the client. This service first download the requested webpages to cloud server’s Server path. Then parse the html to remove image and scripts from the webpage. Then the resulting webpage is compressed using GZip compression technology. Then sent back to the client app.



Fig 4.3 the running server side cloud service performance monitoring

# Result Analysis



## 5.1 Comparing with Opera.

Opera says:

“Speed up on slow networks:

Off-Road mode compresses pages for faster, all-conditions browsing. It helps you stay online when your connection slows down.”

Opera compress the data before reaching to the client computer and provide data usage saving. But it is available as a desktop application. SurferLite provide a store app for similar task.

But, Opera Off-Road mode is very advanced and supports a lot of features. It is a full power browser.

## 5.2 Comparing with Skyfire web browser

It also has server compression capability but it is only for mobile phones. SurferLite is for Windows.

## 5.3 Performance

Basic performance test was done in Visual Studio IDE to check the CPU usage on the software. And following results were found.

### 5.3.1 Client side

The program uses very less CPU to function. Even the highest bump is below 40 per cent. Also a lot of CPU was used by the user interface.

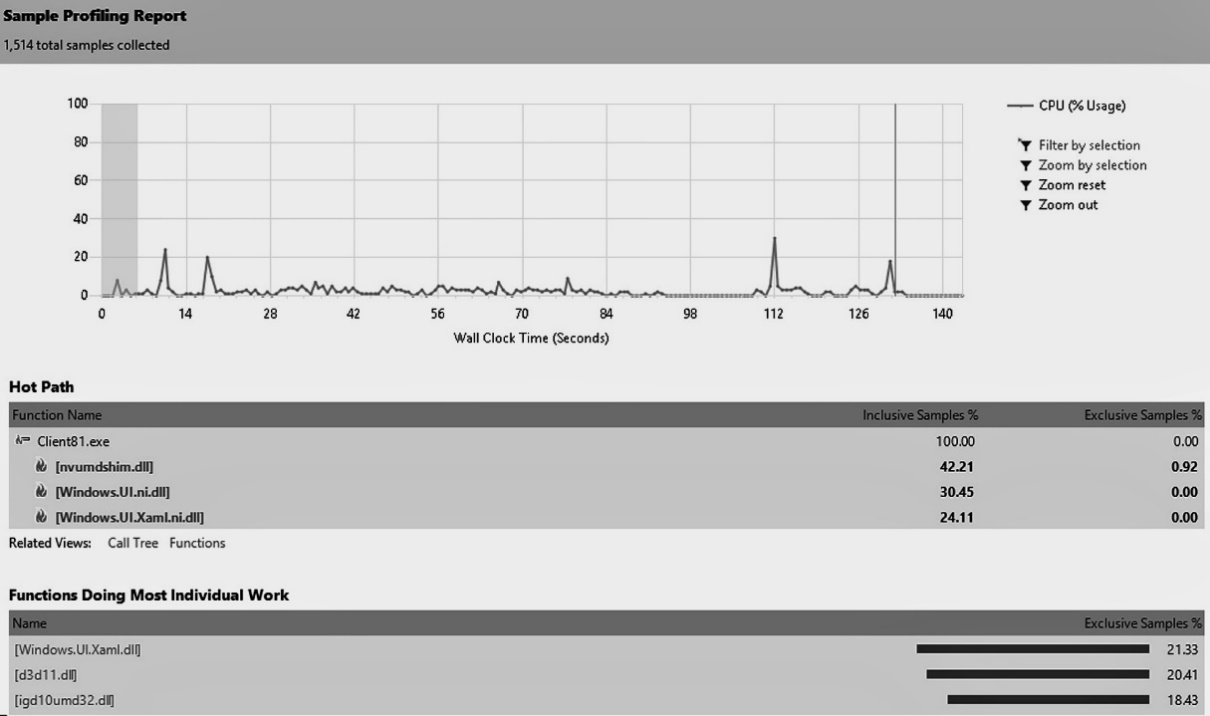


Fig 5.1 CPU usage sampling on client side

### 5.3.2 Server side

The program uses very less CPU to function on server side also. The highest bump is below 40 per cent. Also a lot of CPU was used while getting response from remote hosts.

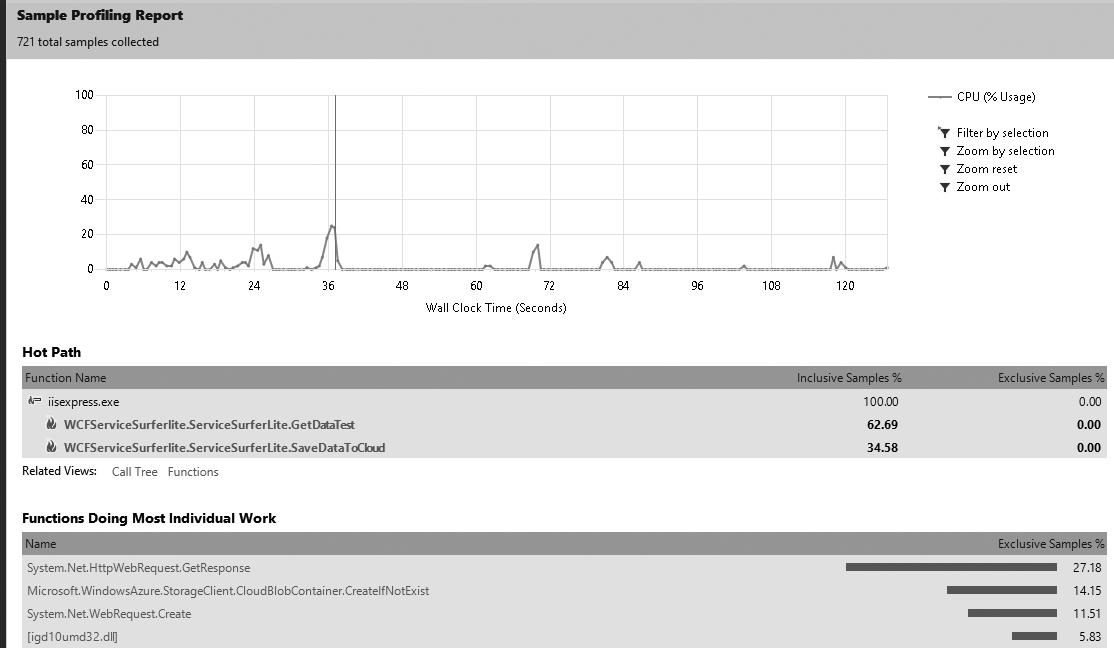


Fig 5.2 CPU usage sampling on server side

In conclusion, project gave result to a combination of software distributed on client and server. It is to be used especially for browsing internet and saving some bandwidth. It also saved some processing overhead on client computers and may save money for volume based internet packages by saving money. The performance was very good on both client and server side.

# Problems encountered

The project was started hoping all will go smoothly that is optimistic view. Most of the problems were faced during the initiation of the project. Later on development and test phases there were few hurdles. But on progressing these are the main problems encountered during development:

1. The project was not clearly defined and there was a lot of choices to be made. The clarity was one of the major problems faced.
2. Technically, there were so many different ways to do a single work that a lot of work was repeated many times after knowing better way.
3. The project was never on schedule, never. The planned time was exceeded by more than 5 times. A work scheduled for 1 hour took more than a day.
4. At some point, there was question of economic feasibility of the project because. Windows Azure is expensive cloud platform. But it was assumed the project to be test project. Also, it was decided of using only free account of 90 days.
5. Codes from multiple ‘check in’ and ‘check out’ was entangled with each other resulting in loosing of track, and needing to restart the Version Control System. So Team Foundation Server was replaced by GitHub for version controlling.
6. A lot of product backlog were non-functional. For example: how to know if a user interface is good. One common method of this is survey, but it is not applicable in this case.
7. Microsoft used to give free azure account for 90 days. But later somewhere between the project days. It gave that only for 30 days and with free usage limitation of 200$ worth of service.
8. Other than above mentioned there were a lot of other problems also, but above are most realized problems till the writing of the report. As the project is just at start phase. More problems are expected.
9. The “Do It Tomorrow” Problem: The project week allotted for completion sounds like a long time, but it goes by quickly. A lot of works were postponed on the project because the due date seemed so far off.

# Limitations and Future enhancement

These are some things that can be improved to make SurferLite more usable:

* Now it has only client for Windows 8.1. Enhancement can be done by developing client for Windows phone with just a little change and for others OS like Ubuntu, Mac OS X using the already made design.
* Currently SurferLite doesn’t support for scripts and images, it can be added.
* It can be made to support image compression through lossy compression and resizing.
* It can be made to also compress streaming video by resizing or resampling but this may need very fast server
* Currently it doesn’t support choosing compression level. It can be added to select compression level to compress more but this increase the load time of page.
* Currently there is no graphical representation of how much data is saved by the program. It can be shown in later versions.

# Conclusion

A product named SurferLite was produced which is a distributed software in client computer and server computer. The client program is a minimal browser which could browse with requesting to the cloud service or without. If the cloud service was on then the page is compressed before sending to the local computer. So data usage was reduced on local computer. The client app is also a decompression client which decompress the compressed data from cloud server and display the result. The app was develop for Windows 8.1 Preview and Windows Azure. The tools used were C#, XAML, WCF. All coding were done in Visual Studio IDE. All the version controlling was done in GitHub. The TFS (Team foundation Server) was used to track the scrum process.

Hence, the project was closed after producing a software SurferLite which could save data usage on local computers.

# References

1. A knowledge base “Windows Azure Training Kit, December 2012” by Microsoft™
2. MSDN magazine “Windows 8 and Windows Azure: Convergence in the Cloud”, Bruno Terkaly, Ricardo Villalobos
3. Offline content “Windows 8 Camp in a box” by Microsoft.
4. “Building Windows 8 Apps with C# and XAML”, Jeremy Likness
5. Web link: <http://en.wikipedia.org/wiki/Cloud_computing>
6. Web link: <http://en.wikipedia.org/wiki/Windows_Azure>