

# Working in the Cloud

---

*Web-based Version Control System for Task-oriented Group and Individual Projects*

Sheng Yu

September 2011

Report on a project submitted in part fulfilment for the degree of

Master of Science in Information Technology

Department of Computer Science

The University of York

Supervisor: Dr. Stefano Pirandola

*Number of words = xx,xxx, as counted by the Microsoft Word's word count command, excluding appendices and words on figures.*

## Abstract

Version control is one of the most common ways to manage computer based projects. However, due to the software based design and file-oriented mechanism, the existing version control systems are too large and too complicated for lightweight use, as well as difficult to be deployed in the diversified environments, such as computers without administrator account and mobile devices. In this project, a web-based version control system for task-oriented group and individual projects has been developed as supplement to the existing version control systems by following the standard process of web application design and development: requirements analysis, methodologies, design, implementation and evaluation.

Under the concept of cloud computing, the version control system which this project design and developed is a fully web-based system. It does not need of installation of any software at client side, and can be accessed at anywhere, even though on mobile devices or computer without administrator privilege in installing software, just need a web browser and network connection.

To effective assign works to group members and avoid hassle in overlap editing files, the system has been designed a task-oriented mechanism in dividing work of files as tasks for assignment.

## Statement of Ethics

Ethics are defined as moral judgement and moral decision of questions about human behaviour. In computer science, it is related to the responsibility and accountability of people design and implements information systems [1].

In this project, as a function set in the title, private project function may involve personal sensitive data of users. According to UK Data Protection Act 1998, Schedule 1 – The Data Protection Principles [2] and ACM Code of Ethics and Professional Conduct, article 1.7 – Respect the privacy of others [3], all the personal data stored within the system should be carefully protected to avoid leak to unauthorised people. In order to protect user privacy, each operation regards personal data would be force requested sign in before continue, people who have not membership or not been authorised will be rejected. Private project data will never use for other purposes without consent of the users.

According to article 1.1 of ACM Code of Ethics and Professional Conduct [3], the working of information system should contribute to society and human well-being. As this project is focuses on aiming computer based group and individual project, it is actually improving efficiency of people's work. The increasing efficiency will gradually improve the quality of life. There are no immediately threats to health and safety.

The article 2.7 of ACM Code of Ethics and Professional Conduct [3] advocated that public understanding of computing and its consequences should be improved. This project is design for doing version control for different kind of computer-based projects, including document writing, digital art and audio processing, etc. After time of using the system, I believe it will gradually affect the understanding of computer in the public.

## Contents

Abstract .....	i
Statement of Ethics .....	ii
Index of Figures .....	vi
Index of Tables .....	vii
1. Introduction.....	1
1.1 Motivation .....	1
1.2 Introduction of this project report .....	1
2. Literature Review .....	2
2.1 Version control .....	2
2.1.1 Existing version control systems .....	3
2.2 Software-based, web-based and cloud .....	4
2.3 Programming languages.....	5
2.4 Interaction design.....	6
3. Requirements Analysis .....	8
3.1 Project goal.....	8
3.1.1 Web-based.....	8
3.1.2 Task-oriented.....	9
3.2 User needs.....	9
3.2.1 Deployment and Portability .....	9
3.2.2 Accessibility .....	10
3.2.3 Easy to use .....	10
3.2.4 Privacy and safety of data .....	11
3.2.5 Response time .....	11
3.2.6 Platform compatibility .....	12
3.2.7 Reporting .....	12
4. Methodologies .....	14
4.1 Software as a service (SaaS) .....	14
4.2 Waterfall model.....	14
4.3 Prototyping.....	15
4.4 Entity-relationship modelling .....	15
4.4.1 Third normal form (3NF).....	16
4.4.2 Entity-relationship diagram .....	16

5.	Design and implementation .....	18
5.1	Work flow .....	18
5.2	Function and mechanism .....	22
5.2.1	File version control .....	22
5.2.2	Task-oriented design .....	23
5.2.3	Task relationship.....	23
5.2.4	Directory version and relationship .....	24
5.2.5	File storage .....	24
5.2.6	Error handling .....	25
5.2.7	Login and Safety .....	25
5.2.8	Performance optimisation.....	26
5.2.9	Table sort.....	26
5.2.10	Migration and modification.....	27
5.3	Database model.....	28
5.3.1	Entity-relationship modelling .....	28
5.3.2	Attribute details of entities .....	30
5.4	Interface design .....	33
5.5	Accessibility .....	34
5.6	Prototype design .....	35
5.7	Prototype evaluation.....	35
5.8	Prototype re-design.....	35
5.9	Two layer PHP architecture .....	35
5.10	Compatibility .....	35
5.11	CSS classes multiple use .....	35
6.	Evaluation .....	37
6.1	Testing of version control.....	37
6.2	Testing of administration .....	37
6.3	Compatibility testing .....	37
6.4	Stress testing and response time testing .....	37
7.	Conclusion .....	38
7.1	Easier local storage.....	38
7.2	Multi task assignment .....	38
7.3	Diff storage and analysis.....	38

7.4	Automatic merging .....	38
7.5	Branching support .....	38
7.6	Distributed system design .....	38
7.7	Multi-level administration .....	38
Appendices .....		47
A.	Set-up guide.....	47
A.1	Environment requirement.....	47
A.2	Step-by-step guide of installation at server side .....	47
B.	Questionnaire for requirement analysis .....	47
C.	Interface samples .....	47
D.	File structure.....	47
E.	Key source code.....	47
D.1	Libraries .....	47
D.2	Styles.....	47
D.3	Pages.....	47

## Index of Figures

Figure 1 - Typical five stages revised waterfall model [52] .....	15
Figure 2 - Group project work flow (general) .....	18
Figure 3 - Group project work flow: Create project and assign tasks .....	19
Figure 4 - Group project work flow: Do tasks.....	20
Figure 5 - Private project work flow: Create private project and tasks; do tasks. ....	21
Figure 6 - Entity-relationship diagram.....	29

## Index of Tables

Table 1 - Nielsen's framework of system acceptability [30] .....	7
Table 2 - Configuration file information .....	28
Table 3 - Attribute details of user table .....	31
Table 4 - Attribute details of project table .....	31
Table 5 - Attribute details of task table .....	31
Table 6 - Attribute details of task_history table .....	32
Table 7 - Attribute details of directory table .....	32
Table 8 - Attribute details of directory_change table .....	32
Table 9 - Attribute details of file table .....	33
Table 10 - Attribute details of file_change table .....	33

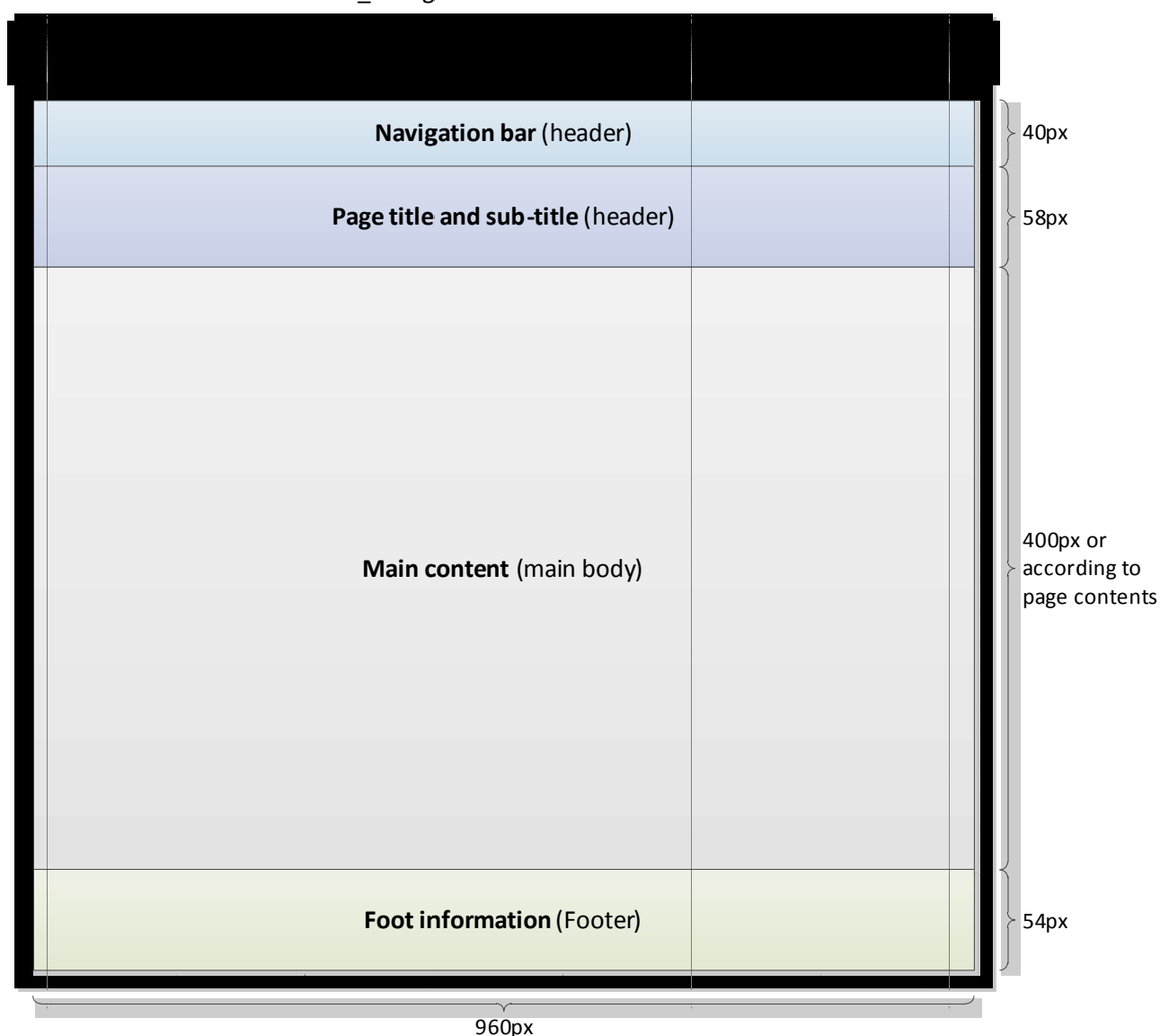


Table 11 - Page layout .....	34
------------------------------	----



## 1. Introduction

### 1.1 Motivation

Version control is a famous way in controlling revisions of works, especially programming and documentation writing. It keeps all the historical change of work and gives use chances of roll back their work at any time. It also can be used as a synchronised platform to make the work can be worked at many places and with collaborations from other people. However, as I observed, the people uses version control in managing their work is in absolute a few number, even the computer science students. I think the reason lead this may be the existing version control systems have too many limitations, especially must be pre-deployed before use. The concepts and using of existing version control system is also complicated. Some experienced user with version control system may think the system can tracking file divided by different tasks. So I wish to design and develop an easy to use web-based version control system with task-oriented feature for computer based group and individual projects as supplement of existing version control systems.

### 1.2 Introduction of this project report

This report is about describing the system in the project which I did.

In the report of this project, it has been displayed in the following sections: literature review, requirements analysis, methodologies, design and implementation, evaluation and conclusion.

## 2. Literature Review

This section was about the literatures review related to this project – a web-based version control system. As this project was to develop an application as a web-based system for doing task-oriented version control for group and individual computer-based projects, the literature review could be divided into these parts: version control, existing systems, web and cloud, programming languages and interaction design. These parts were all about the features of the system.

### 2.1 Version control

Version control, also be called as revision control or source control, was a method of managing files related to development of projects through their whole life cycle [4]. It is essential for multi-developer projects [5]. Lots of popular version control software includes CVS and Subversion run as client-server model, so they support more than one users working especially programming for a same project, because the multi-clients can be connected to a central version control server to be synchronised [6] [7]. The general features provided by version control include storing each commit/version of files or directories, allowing rollback, showing modification history and assist merge/integration [4]. Based on the features, many kinds or parts of project can use version control to be effective managed, such as projects of software development and documentation writing. Each commit/modification can be marked with properties of which author did this commit and what time it committed. The commits also comes with a unique version code to identify the times of modifications of it. Users can also add some comments to the version when they commit it, so it is very clear to identify what the users modified in there committed version. When a version commit has been identified as worse than an older version, the modification s can be easily roll backed to a previous version at any time [4].

Even though the existing version control systems sound wonderful enough, however there still an important point should be noticed, that is the “concurrent access” problem exists in almost all the version control systems [8]. Every project developer can access an entire project, and have privilege in changing any part of the project. It is hard to assign responsibility as tasks to the developers, so the developers may forget where their positions in developing the projects are. When a developer mixed up his/her range of development, and changed some files which someone else is working for, it may leading to serious problem, or waste of time in combining works manually, even though merge policy can do combination of most their works, the result cannot be guaranteed as the most expected one due to the complicity of different kinds of working [8] [9] [10]. Some version control system, such as Subversion, has designed the “lock” feature to prevent this problem. However, even though a file has been locked by a developer, other developers can also get copy of it from their local update before file locked, or from archive of older versions. It is hard to restrict this problem under the current version control policy [9].

To solve this problem, this MSc project was aimed to develop a version control system which has “task-oriented” feature avoids developers forget their role in development, also prevents the happen of concurrent editing of files. Task-oriented development supports more than one people working with one project at its separated tasks, without interrupt and overlap. Assignment of tasks for the developers can be confirmed by a discussion in a group of each developer’s strong points and weak points, and finish assigning by group leader in the new system. To achieve task assignment, a schedulable feature may be useful. In the system of this project, the tasks could be set up with relationship with other tasks. In 1910, Henry Gantt first published the concept of predecessor in his famous Gantt chart [11]. It resolved the organisation problem of group collaboration in scheduling very well. This project uses the concept from Gantt chart to scheduling tasks. A task could have a predecessor/father task, which means in order to make sure tasks to be worked in sequence, a task can be start only when its predecessor task already finished.

### 2.1.1 Existing version control systems

As coordination and work management were important features of the system, some design can be learnt from operating mechanism of existing version control system. In the following, the popular software-based client-server version control system CVS [6] and Subversion [7] will be analysed by their important features and associated with the idea inspired by them.

#### *a. Unique version number*

Every content modification of directory or file will generate a new version attached to the directory or file itself. Moving, renaming and deleting will also be considered as a change. Each historical version will be kept for in-case use [7].

#### *b. Atomic commits*

For coordination considering, every submitting of modification set will generate a version of modification history, even only a part of the set has been modified [7]. As the system which this project developed was task-based, so it should be different from the existing SVN versioning mechanism (all files will be updated to a latest same version code after an every submitting command). Task-based feature limited area of modification set to tasks, therefore, the versioning mechanism could be designed as: every time of file and directory modification will increase the version number of file and directory individually, and the task version will be increased when a submit operation has been executed, no matter how many modifications of file and directory have been made, even only renamed a file. For example, when commit the modification of one file in a four file task, the version code of the one modified file will be increased by 1, other unchanged file will still remain the it old version code, the version code of the entire task contains this file will also be increased by 1, because the new mechanism is designed to able to track modification of each file associated with whole task.

### *c. Locking*

To avoid unsynchronised editing, lock-work-submit-unlock is a very good feature provided by SVN for conflicts free editing [7]. In this project, the task as a minimum assignable unit, should be locked when a user start doing the task. When a task starts by a user, a lock would be set to a task, other user except the user who sets the lock cannot request to do any modification of the task, even he/she was already been assigned to the task.

## 2.2 Software-based, web-based and cloud

Lots of popular version control systems are traditional client-server model based [12], even though some version control systems were built with the distributed approach, like Git<sup>1</sup>, however, it is very complicated for small group and individual projects to use, due to it is hard to understand and use for non-experience users [13]. To set up a traditional version control system, it requires sets up three parts which will be configured to work together: centralised server runs server side software, client(s) runs client side software and reliable network connections between server and client(s). Server stores every versions/commits of each files, client(s) stores a copy of latest version of project files and allow user working and changing it for future commit [4] [14] [15]. Network connection is a very important component in version control system, because it connects server and client(s) for both of them can be synchronised to latest status and keep files up-to-date. In normal way of doing a project in a version control system, users as clients usually sign in by client side software to the repository of the project in version control server, download (update) latest version of copy of all files related to the project from server to their own computer, and work for the files. The users may have arrangement in advance to avoid possible overlap working in same files, at least in same class of a programming project or same paragraph of a documentation writing project, because the existing merge algorithm in the most version control systems could not combine overlapped work in different commits without conflicts faultlessly. When a user finished a milestone of working in his/her files, he/she needs to re-login by client software into project repository in the server, commits all the working he/she did to the server. After commit, a new version code will be generated. When other users as clients tried to download/update the project, if newer version of files on the server has been detected after compared to the local version, the files in the client side would be updated to the latest version [4] [16].

The helpful features made version control systems work great for managing many kinds of project. However, it needs to install software at both server side and client side to perform version control actions [4] [16]. For entry-level users, it may be hard for them to install and configure client side software to work with server which providing version control service; For users who often working in different kinds of environment instead of their own computer, such as working in computer lab and on mobile devices like iPad, they may have

---

<sup>1</sup> Git is a distributed version control system developed by father of Linux – Linus Torvalds, used for manages development of Linux kernel originally, now for some large scale projects [67] [68].

not privilege to install client side software of version control systems, or the mobile device does not support software for version control.

To allow client side user can working at most environment with version control support, it is important to find out a way deliver the service without need of installing software. By the inspiring of more and more popular cloud computing concepts like Google's Cloud [17], this project was considered to develop a fully web-based version control system which does not require any installation of client side software. Web-based design can overcomes some drawbacks of software-based design, such as hard to use at anywhere and hard to configure by entry-level user. It can be easily accessed at any computer, even a mobile phone, just need a browser and network access to the version control server. When transfer from existing version control system to web-based system, user can not only benefit from the "access everywhere" feature, but also gain from an important feature – "easy upgrade at cloud with less disruption". If the system needs to upgrade to latest release, it does not need to ask user to do upgrade of their client side software like traditional way. To upgrade the web-based system, just need to change the server side software, and all users could start using the new system as usual via their own web browsers [17]. The concept of "network", "cloud" and "web-based" not indicates the system is only Internet-based. However, it can run at a local area network as a "local cloud", because the Ethernet supports same technology of Internet's, such as IP based TCP connection and HTTP protocol, which allows web-based system run at local area network similar to run at Internet [18]. To set up the web-based system in a company-wide network, it needs to allocate a computer as server, configure its Apache, PHP and MySQL running environment, and install web-based version control systems in it. After installation, users can direct access the server's domain name<sup>2</sup> or IP address, even a private IP address<sup>3</sup>, on any browsers by devices connected to the same local area network with the server. Due to the centralised structure and network connection dependency of existing version control systems, the cost of migration from existing systems to the new web-based system might be very low – just need do configuration on server side with new web-based system, and tells users to access the new system via their browsers, even mobile browsers, and then continue their working.

## 2.3 Programming languages

In order to develop the system to be used at web, there are several programming languages available, such as Java/JSP<sup>4</sup>, ASP<sup>5</sup>, ASP.net<sup>6</sup>, CGI<sup>7</sup> and PHP. In these languages, Java as a

---

<sup>2</sup> Domain name, an easy to remember name identifies a computer, which have mapping relation (records) to IP addresses, allows user to access various service on a server by only use its domain name, such as http and ftp (A record), mail service (mx record), etc. [73] [74].

<sup>3</sup> Private IP address, is IP address in range of pre-reserved network address space. It usually use in local area networks, which can only be accessed by computer in the same network [69] [70].

<sup>4</sup> JSP, JavaServer Pages, a technology uses Java language creates dynamic web content [66].

<sup>5</sup> ASP, Active Server Pages, is a server-side script engine from Microsoft for dynamically web pages, uses VBScript and JavaScript as server-side programming language [78].

<sup>6</sup> ASP.net, is Microsoft's second generation server-side script engine, uses Microsoft's .Net Framework as libraries, and also uses object-oriented programming languages such as VB.net and C# [76] [77].

popular programming language in object-oriented software development can also provide web service by working with JSP on Servlet<sup>8</sup> [19], even though it provides the most object-oriented structure for programming, however, it is hard to set up server side environment and also hard to programming in the scale of this kind of project, and its structure is too complicated for lightweight development, because of its full object-oriented design and great number of components requirement for running [20]; ASP, an Microsoft's outdated web application engine, which is very popular at the era of Microsoft Windows NT 4.0 and Windows 2000, with many down sides such as high cost, slower speed, lack of library support, low safety design, no debugging support and hard to do migration to other platforms [21]; ASP.net is the latest Microsoft's web application engine, overcomes many drawbacks existed at ASP, however it still not an open source platform like before, so it still hard to do migration in the future, and high cost in setting up [21]; finally, PHP, is the most shining web programming language and platform with lots of great features, such as fully open-source, object-oriented support, abundant built-in library functions and rich high quality open source resources. It is totally free [22], but still has great features with commercial programming language and platforms. The running environment of PHP is also easy to be set up. A number of pre-configured server kits are available to do one-click installation of PHP running environment [23] [24] [25]. It can run with open source web server software such as Apache<sup>9</sup> and Nginx<sup>10</sup>, also Microsoft's IIS<sup>11</sup> series. Some opponent would say PHP is too simple so it can only be used in developing lightweight application, however, Facebook as one of the largest websites, uses PHP as its main programming language and also did great contributions to make PHP better [26]. Overall, PHP has been chosen as the programming language for development of this project.

As a great partner of PHP in LAMP<sup>12</sup> group, MySQL<sup>13</sup> has been chosen as the database system for development of this project, because it provides great number of features in small installation size with easy configuration. MySQL is open-source and free, also have lots of useful features, such as "view" and "lock", it was a great choice for different kinds of use and easy to be customized [27].

## 2.4 Interaction design

Interaction design is "the practice of designing interactive digital products, environments, systems, and services." [28] It defines and designs the behaviours of human in perceiving

---

<sup>7</sup> CGI, Common Gateway Interface, is a platform-free interface for client to execute application on web server [79].

<sup>8</sup> Servlet, is a server-side Java application, generates web pages as interlayer between client request and server response with platform-free and protocol-free features [19].

<sup>9</sup> Apache, a common name of Apache HTTP Server, is an open-source web server application which has been used most widely in the world, and can be used in lots of operating systems [80].

<sup>10</sup> Nginx (Engine X) is an emerging high performance open source HTTP and proxy server [81].

<sup>11</sup> IIS, Internet Information Services, is Microsoft's Windows-based internet server application [82].

<sup>12</sup> LAMP, a powerful bundle of open-source software working together as a web server, includes Linux (operating system), Apache (HTTP server), MySQL (database) and PHP (script language) [83].

<sup>13</sup> MySQL, an open-source database system, developed by MySQL AB, now is a part of Oracle [84].

and using digital objects [29]. All the researches of interaction design are focus on its objective: to make product easy to be used; to technology happy to be experienced. In order to achieve the objective, there are several things to be finished in sequence: the expectation of different kind of user should be classified for analysis; understanding what behaviours of user would do or already did during their using an interface; understanding their psychological characteristics of using the interface. The relationship between product and user would be built by the interaction design of interface, to help user to do what he/she need via the interface effective and relaxed [29]. In measuring the usability of a product and its interaction design result, Jakob Nielsen has written a framework of system acceptability [30]:

<b>Learnability</b>	How easy is it for users to accomplish basic tasks the first time they encounter the design?
<b>Efficiency</b>	Once users have learned the design, how quickly can they perform tasks?
<b>Memorability</b>	When users return to the design after a period of not using it, how easily can they are establish proficiency?
<b>Errors</b>	How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
<b>Satisfaction</b>	How pleasant is it to use the design?

**Table 1 - Nielsen's framework of system acceptability [30]**

The five points from Jakob Nielsen describes the measuring standard of usability, also told product designer especially interface designer make sure their works meets the five points to enable good user experience.

In this project, as the aim of the web application is to provide an interface for user easy to discover functions and feels relax in using, interaction design is an important part of this project, because users may have less experience in using the system. Users may not clear how the mechanism work in the system, however this project should make sure them can feel free in using the system with less barriers.

### 3. Requirements Analysis

In the requirement analysis section, it has been investigated in areas about requirements to be considered in to design and implementation of this project.

#### 3.1 Project goal

As described above, the goal of this project was about designs and develops a version control system with task-oriented feature for groups and individuals to use it in aiming their computer-based projects via web interface. In comparing with the existing version control systems, the most two important different features of the system of this project are web-based and task-oriented.

##### 3.1.1 Web-based

The first features, web-based, is influenced by the more and more popular cloud computing concepts. In this era full of cloud computing, computer is becoming a tool of using cloud services. The widely use of internet provides many favourable factors for collaboration as group [31]. For example, in Google's design of cloud computing, user just need a device with web browser to use nearly all the Google services, no matter what devices you have and where you are [17]. Google Apps is an interest product should be noted. It is a fully web-based enterprise office solution, which contains enough tools for companies to word with it more flexible than using local software. User do not need to do installation of any programme, and do not need to waste time in considering software compatibility between different hardware and software combination. To run an online web-based programme, users just need a mainstream web browser, even though on a tablet or mobile phone, without requirement of hardware or software configuration [17]. In Google Apps, users could work together without sending works anywhere. They just need to create or upload a previous work into Google Docs, and ask the group members to sign in with their own accounts, after that, all the group members can see the entire work and their can modify or add new things, or comment of some paragraphs. When a user did his/her changing of the work, the old version would be archived as "version control", in case roll back to old versions if the modification is harmful. Users could working together at their own computers by sign in to a same system without need of sending their work to each other, as well as prevent the confusion of versions of works [32]. Another product should be noted is Google Chrome OS. It is the result of a pure web-based cloud computing concept. The Chrome OS only have a web browser, and the applications running in the OS are all web-based. The netbook loaded with Chrome OS do not need powerful processing capacity, just few basic processing power of web content is enough, because the core of web-based cloud application is running in the cloud side or server side, the user side or client side computer just in role of helping user in interacting with user interface of the application [33]. By experience of the plentiful advantages of cloud computing, as a trend of moving from traditional software-based application to web-based application, this project is aiming in designing and developing an



easy to use version control system in providing a fully web-based interface for users can be used at anywhere.

### 3.1.2 Task-oriented

In the existing version control systems, files are base units to be version controlled by project. When commit modifications, all files modified in once would be counted as one version of a project [6]. In the common group working for a computer based project, they may have several members working together. In most cases of computer based project, members in a group are working their works paralleled, and combined/merged at the end when finished working. During the working, they are communicating with each other, and then continue doing their work in files. To improve efficiency of a computer based group work, a better way is to clearly define and design tasks within a project and assign them to group members by analyses points their good at or not, because distinct task assignment can maximum uses group psychosocial traits, as well as give pressure for member for motivation by compare progress and quality of tasks [34]. In the existing file based version control mechanism, group members working in a same project may usually confuse in identify duty of files [10]. To make the “task” can be a part of the version control system, in this project, the system was focus on build a task based mechanism in order to tracking two level of versions in a project: task level versions, directory and file level versions, for tracking directory and file within task assignment.

## 3.2 User needs

As an application for people, user is one of the most important things which this project should be focused. As defined in the project title, this system is for computer based group and personal projects with version control. According to my observations, people who doing documentation writing or programming may focus on version control in aiming their work of keep working safe or make collaboration easy.

### 3.2.1 Deployment and Portability

For some kind of users, they may be used to working in computer lab or with public computers which have not administrator privilege. In using that kind of computer for work, a big problem is the user does not have any right in installing software. If the user is also rely on using version control system in helping his/her work, he/she may crazy in the truth of no version control client software available for use.

In order to help this kind of user can use the version control system as usual, the deployment of the system especially at client side may not require administrator privilege. At the same time, if the client side software not support in running at different kind of operating systems, it may limits the usage of the system, so it should support different kinds of operating systems.

For people who like to use version control system on a mobile device, such as on netbook, tablet or mobile phone, they may wish to get the similar experience or familiar interface of when they use it on a regular desktop PC. If user can access the system at any platform with same or similar feeling and process, the system would get better result in Jakob Nielsen's framework of system acceptability [30], which means the system have better learnability for users to found how to use the systems when they first time transfer to a new platform and better efficiency for they to start perform tasks quickly. So the system should have a universal design of interface for different kind of platforms.

Even though the system was install-free for client side use, it still needs to set up before use at server side. To make sure the system can be installed at different kind of servers (at least come with PHP and MySQL), the parameters of running will not be integrated in the programme. A better solution may be set an external configuration file to places all the parameters such as database connection information, system name, time-zone, administrator e-mail address, etc. When the programme needs to know the value of parameters, the configuration file can loaded in the beginning to be initialized.

### 3.2.2 Accessibility

In order to serve as much people as the system can, it is very important to give accessibility to people who uses screen reading software. The W3C has defined standards for better accessibility, which is the "Web Content Accessibility Guidelines" (WCAG). For example, as WCAG 1.0 Guideline 1 defined, all the images on the web page should have an alternative text label for screen reader to find and read; in guideline 4, the natural language usage should be clarify in HTML head; in guideline 5, table header should be defined to identify property of columns [35]. By following the guidance, blind user, amblyopia user, people in dark environment or any kind of people who is not convenient in understanding written pages on screen, could benefit from the special consideration, and get what they need on the web pages much easier [36]. As this project is aim to provide good accessibility for user, it is necessary to design by following the guidance of WCAG document carefully.

### 3.2.3 Easy to use

As I observed, the York CS students in doing their group or individual coursework often not interested in using version control system, even some Software Engineering students. The most common reason was "it is too complicated to configure the coursework to be version controlled" (investigated by the questionnaire in Appendix B). In setting up a traditional version control system to be worked at client side, it needs to register an account first. After registration, the user needs to apply for a repository/project to be opened at the client side of the version control system. Then the user downloads the client side software of the version control system, log in with administrator account, install it. When finished installation, the user needs to configure a file folder as a project with username, password,

server information and repository address to “check out”<sup>14</sup> at version control server. If all the above steps finished without face any error, the user can do first commit of his/her files and start using the version control system. Once the user decided to move his/her work to another platform, he/she needs to redo everything to re-configure the client side settings [4].

This project aim to design an easy to use version control system. To make the system easy to use indeed in some small project like coursework in York CS department, it should focus on the core functions and make the process in using core functions as simple as possible. As analysed above, the web-based interface would be a simple way to provide core functions with an easy to use interface. User may use the system like they use a webmail system, sign in and tick the “remember me” label to perform automatic log in at next time, then create a project with project description, and create tasks for work to be better scheduled. After all, the user can easily upload files into task, as simple as add an attachment when composing an e-mail. Now a new version of the files would be created. That is the easy to use requirement.

### 3.2.4 Privacy and safety of data

When working for a project at the version control system, the group members or the private project owners may not wish others to see what they are doing and what they did. They may prefer not to use version control system, rather than fear in leaking their work to unauthorized people. People usually have privacy consideration when placing sensitive data on a shared media, even though the media provider promised the data will be well protected [37]. To reduce people’s concerns, the system needs to be developed with well-designed mechanism in making sure the safety of data. All the data must not be obtained by people without permission.

### 3.2.5 Response time

In user experience evaluation, page response time is an important factor in influent service quality [38]. As defined by Miller of his basic advice regarding response times [39], one second is the maximum cap for user in feeling freely of navigation between pages. If response time of switching pages is longer than one second, user may feel waiting, and the experience would be reduced.

To provide best user experience with fast response time, the times of database queries need to be fewer and fewer, because database query often costs much longer time than PHP internal processing. Large size content like images usage within pages should be also avoided as possible. The consideration of page response time should include delay of network connect speed, even though the Internet bandwidth today is much higher than years ago.

---

<sup>14</sup> Check out is a Subversion (SVN) command, which is make server side repository can be associated with a local file folder as a version controlled project [4].

### 3.2.6 Platform compatibility

Compatibility of website is about the similarity of web pages in expression when visits via different browsers and platforms [40]. In the past years, during the development of Internet, Microsoft Internet Explorer as a built-in web browser of Microsoft Windows has been the most widely used web browser [41]. At the age of Internet Explorer occupies absolutely the most market share of web browser (2005 and before), web designers should only need to consider how the pages displays on Internet Explorer, because users who use other browsers are minority groups with only fraction market share. However, after 2005, by the grown up of Mozilla Firefox, the market share of Internet browser is reducing day by day [41]. More and more people start using operating systems other than Microsoft Windows, which Internet Explorer not supports, such as Linux. At same time, some Windows user also starts use other web browsers, because some browsers such as Mozilla Firefox, Google Chrome, Apple Safari and Opera have better compatibility of W3C standards. In the Browser Statistics form W3Schools.com [41], the market share of both Mozilla Firefox and Google Chrome after April of 2011 has already beyond the share of Internet Explorer. In order to provide similar experience for various browsers, a good way is to make sure the website is W3C compliance.

Data from W3School Browser Display Statistics [42] shows there are still number of users using screen resolution of 1024x768 pixels. To avoid these users always to drag pages horizontally in browsing, the page must no wider than their maximized browser windows. This would make the 1024x768 screen users using the system much more relaxed.

JavaScript is a widely supported scripting language in processing client side user behaviours locally. Some web designers started relied on JavaScript in more and more operations, even though use JavaScript for input validation instead of server side validations [43]. They may think the use of JavaScript can provides better user experience, however, they forgot, if the web browser not support JavaScript, it will be a disaster. To avoid disaster from non-JavaScript-support browsers, some key validations should be designed at least at server side.

### 3.2.7 Reporting

When using the system, users may found problem of tasks, project or even the system itself. To make the problem can be easily communicated with staff or other group member in the same project, the obvious indication of contact should be designed in pages. User must be made sure can find the way of communication easily, and the way of communication need can be commutated in two-way.

If user made an illegitimate operation which the system cannot deal with, an error would be generated. When error occurred, the system must not be crashed. The reason of error should be displayed on the screen for user to be announced. If the error is caused by system itself, the system administrator should be sent an e-mail for a notice of this error.



## 4. Methodologies

Methodologies of this project were related to the models and methods used in design and development. As the project was based on the concept of cloud, software development and web design, it should be designed and implemented according to the methodologies of software as a service (SaaS) in cloud computing, waterfall model in software engineering, prototype design in interaction design and entity-relationship modelling in database design

### 4.1 Software as a service (SaaS)

Software as a service (SaaS), also known as on-demand software, is one of the three<sup>15</sup> well-known service models of cloud computing [44], which delivers software applications by user needs via Internet [45]. The growth and public acceptance increase of SaaS applications give great experience to the designer, and the most important characteristic made SaaS solutions successful should be the use of web-based interface [46].

In contrast to traditional software, the SaaS applications have less platform dependence and resource usage. For ordinary users, thin client with browser and Internet access is enough to support the running of cloud applications. At the same time, by the popularity of the high-speed Internet connection and the more standardized web technologies such as HTML, CSS, JavaScript and HTTP reduced the cost of developing SaaS solutions, more competitive applications were being developed and promoted.

Some people may have concern of the security of data, because of all the data was stored in the centralised cloud, however, the cloud service provider usually have more experience in protecting personal data than individuals [47].

SaaS solution advantages have been summarised as low cost, fast deployment, easy access, safe, etc. [48] As this project is aim to build a SaaS application, it should have features of [48]:

- Web-based interface;
- Well data protection;
- Less software dependence;
- Anywhere access;
- Fast deployment;
- Low cost

### 4.2 Waterfall model

Waterfall model is a popular software development processes, which commonly have stages of conception, initiation, analysis, design, implementation, evaluation and maintenance, first

---

<sup>15</sup> The other two models are Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [44].

formal described by Royce in 1970 [49]. The stages above made sure software can be designed and developed with good quality and well scheduled, however, it may not adapt to all software development projects. In reality, the stages can always hardly be linear executed. Some stages such as conception, initiation and analysis have been disputed as too cumbersome. To make sure the waterfall model efficient and effective, some stages have been combined. As shown in Figure 1, the most acceptable version of revised waterfall model now have five stages, including requirement analysis, design, implementation, evaluation and testing, maintenance [50]. It should be noted that the waterfall model is now a standard in software development [51].

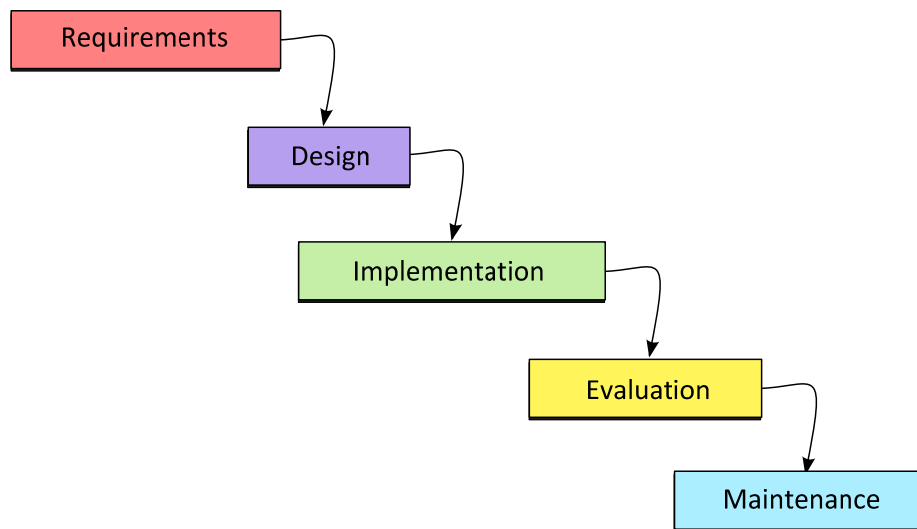


Figure 1 - Typical five stages revised waterfall model [52]

### 4.3 Prototyping

Prototyping is the process of creating early sample by an idea or a design [53]. In information system, prototype helps system designer build an intuitive model for doing analysis and basic evaluation. For design of information system, low fidelity<sup>16</sup> prototyping is great and simple step in displaying idea from designer. After first version of prototype finished, it can be used at simple evaluation for problem finding. Further version of prototype re-design may require for making the design better. During the progress of repeating prototyping and evaluation, more and more problems behind design would be continuous discovered and solved, especially user experience related problems.

### 4.4 Entity-relationship modelling

Entity-relationship modelling is method in describing data, structure and relationship in relational database [54]. The conceptual design of database is usually based on the entity-

---

<sup>16</sup> Low fidelity prototype, usually be written as low-fi prototype and be called as paper-based prototype, is basic and the most low-cost type of prototype, often be used in initial stage of design. It can be performed by several software, or just using pen and paper [89].

relationship modelling, and is it a very important step within the design stage of software development. A strict (good) E-R model is usually [55]:

- Has many entities (tables<sup>17</sup>)
- In third normal form
- Connected by primary key and foreign key
- One to many relationships

#### 4.4.1 Third normal form (3NF)

The third normal form, often abbreviated as 3NF, is a normal form in database normalization, which defined by Edgar F. Codd in 1971 [56] [57]. All database tables in 3NF should meet the conditions: table is already in 2NF<sup>18</sup> and its attributes except primary key (ID) should not dependent on other attribute [56]. In 3NF, database tables could minimal their redundancy and make the SQL statements could be connected together for multi-table operation much easier [58]. This means if a table in database meets requirement of 2NF, but not meets 3NF, it need to split into several tables. Redundancy attribute, which is the attributes dependent on other attribute, must to move out to a new table to keep avoid mistake/exception in operations (insert, update and delete) may lead by redundancy, also help tables can be isolated and uses SQL statements to be connected by their primary key and foreign key [59].

#### 4.4.2 Entity-relationship diagram

Entity-relationship diagram (ERD) is the chart shows entities of attributes in database and their relationships [60]. It looks like the class diagram of UML<sup>19</sup> in software engineering. In ERD, tables should be connected by primary keys and foreign keys referral. To identify the “one to one”, “one to many”, “many to many”<sup>20</sup> relationship between tables, some notations has been defined for distinguish them.

It should be noted that there are four syntax of data modelling notations [55]:

- Information Engineering
- Barker Notation
- IDEF1X
- UML

---

<sup>17</sup> Entity is a more academic name, and it usually be called as “table” as an easy to understand name.

<sup>18</sup> 2NF, the second normal form, states all the records in database table should be unique identified [57]. The common way of making table meets requirement of 2NF is adding a unique ID to each record as its primary key [85].

<sup>19</sup> UML, Unified Modelling Language, is the standardised modelling language in object-oriented software engineering for its structure, active, process, etc. It can be demonstrated by number kind of diagrams, such as use case diagram, class diagram, sequence diagram and active diagram [90].

<sup>20</sup> “Many to many” relationship is not yet supported in relational database [91].





## 5. Design and implementation

This section was about the details of design and how it be implemented into programming.

### 5.1 Work flow

In using the system, all the users including group leaders and group members, even private project holders, should follow a work flow to use the system in controlling versions of works. As the system is task-oriented design, working units in the system would be divided as project, task, directory and file. File is belonging to directory; directory is under the control of task; task is managed by group member within project, project is monitored by group leader. Each person registered in the system as a user can work for both group project and private project. In a project group, the group leader is also a member of the group. If the system has several project groups, a user can work for more than one group at same time.

[Project-task-directory-file figure here](#)

[Group member relationship figure here \(Boundary-less Organisations\)](#)

By analyses the relationship between file, directory, task and project, also user of group leader, group member and private holder, a general group project work flow chart has been drawn below.

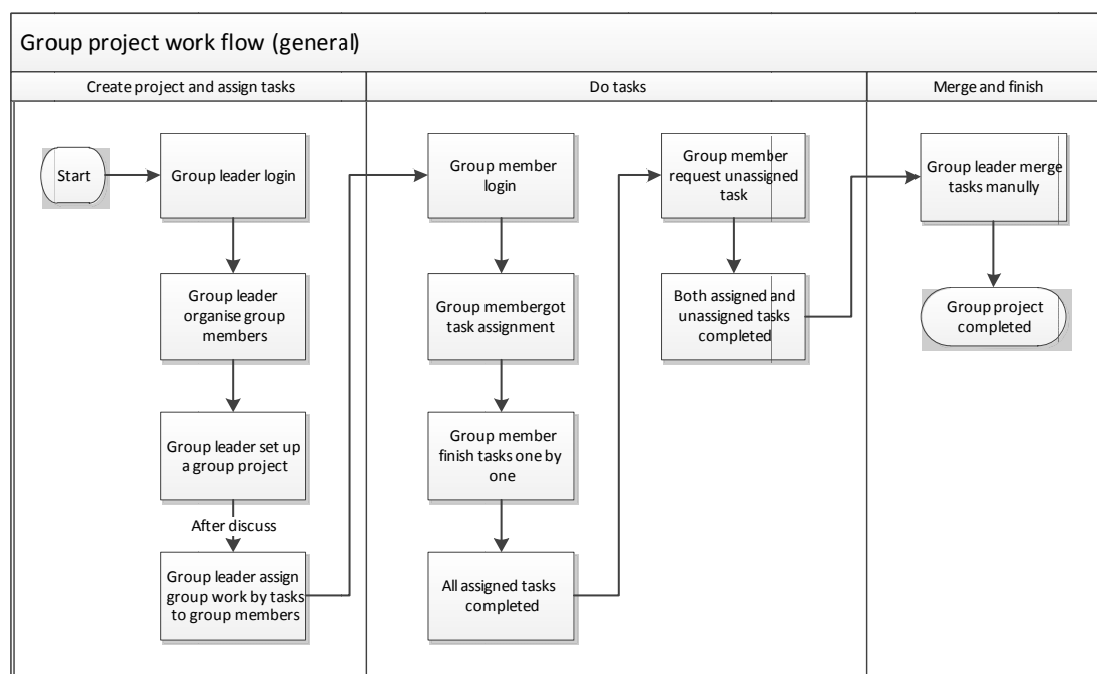


Figure 2 - Group project work flow (general)

Figure 2 shows the sequence and general steps of doing a group project. There are three main stage of working in a project. The first stage is about creating project and assigning tasks. In this stage, project and its tasks should be created and assigned to group members

by group leader after a meeting of discussion. The second stage is about group member doing tasks. In this stage, after assignment at the pervious stage, group member sign in the system by each of their accounts, start doing the assigned tasks, finish doing an upload it. If all assigned tasks of a user already finished, the user can request remain unassigned task (if exist) to do. After every task finished, group leader merge all of them. After that, the task will be finished.

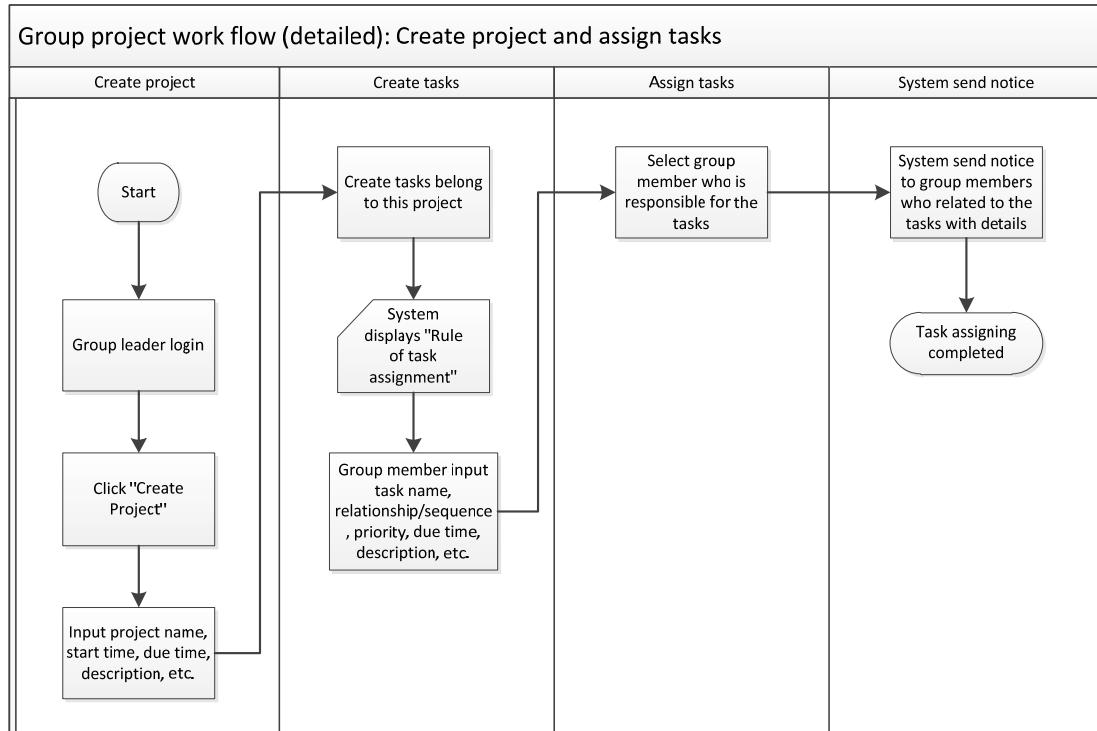


Figure 3 - Group project work flow: Create project and assign tasks

Figure 3 describes the details of first stage of Figure 2. When starting a group project, everyone in the group should have a meeting to discuss each member's different abilities to the project. The member who has the most obvious leadership should be act as a group leader. After decided the group leader, the group leader should sign in the system, create a group project and input some details of the project, such as project name, start time, end time and description. When finished creating a project, the user who did this will be treat as group leader automatically, then the group leader can continue create tasks belong to the project. In creating tasks, the group leader should give details to each task, like task name, start time, due time, description. A task also have a property of predecessor task, it could be set when creating the task. After finished creating the task, the group leader can assign the task to a group member. After finished assignment to all group members, some unassigned task can be leave temporarily to wait group member who can finish his/her assigned task quickly and back then request unassigned tasks. If the task assignment request has been confirmed, an e-mail with task information will sent to the group member.

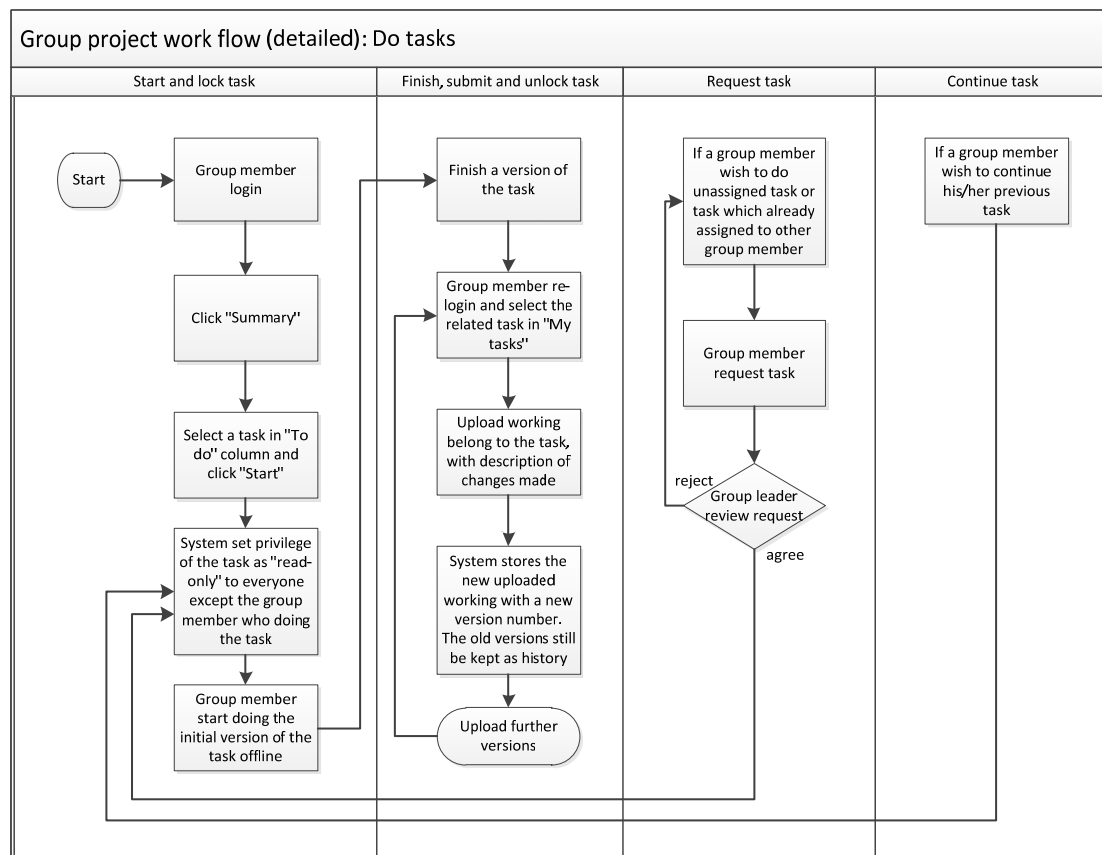


Figure 4 - Group project work flow: Do tasks

Figure 4 is related to the stage of doing task in Figure 2. In the progress of this stage, group member login into the system and then they will be redirect to the summary page. In the summary page, they will see columns including tasks doing, tasks to do, tasks waiting and tasks finished. If there is no task in a column, the column will be hided to save space of the summary page.

[Meaning and conditions of tasks columns table here](#)

When a user signed in, all the user's tasks which not yet begun will be placed at the "tasks to do" column. When the user clicks one of the tasks, a page of starting the task will be displayed. After this, user can create directories to the task as a structure of file storage. Then the user can start doing the initial version of files within for the task offline. Once some files finished, or at a milestone, the user can upload them to the task at any time. When uploading file, the user will be force requested to input some description to the uploading/change of the file, and also need to describe the task version commit, because each time of file change will generate both a new version of file and task by the tracking consideration. After submit changes, the user can download his/her change at any time for further work. If he/she re-uploads the file with changes, of course both a new version of file and its task will be generated. The older version will still be kept in the system as a historical version for in case use.

In a group project, there may have some unassigned tasks for group member who has already finished their work and still have time remaining to do more work. If this kind of group member wish take unassigned tasks, the can click buttons near the unassigned tasks to send request to group member. When group member received the request, he/she can approve the request and assign the task to the user wish to do it.

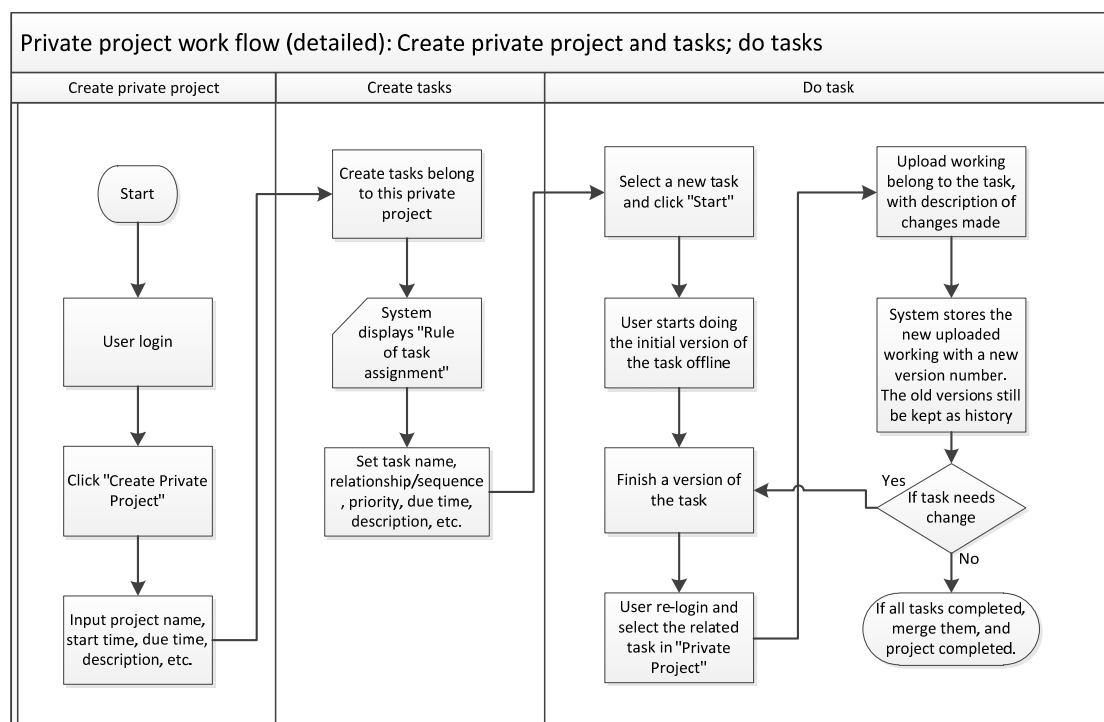


Figure 5 - Private project work flow: Create private project and tasks; do tasks.

In Figure 5, as defined in the project title, the system can also do version control for private project. Every member in the system can create private project. The process of creating private project is very similar to a group project. Just assumes the group leader of the

private project and the only member of the project is the private project owner itself. There is an option (tick box) of private project in creating a project. If the box has been ticked, the project will be set to a private mode, and anyone else than the project holder will not got right to access anything related to the project.

## 5.2 Function and mechanism

In this part, the function and mechanism of the requirements of users' real needs will be designed and implemented below, including file version control, task-oriented design, task relationship, directory version, directory relationship, file storage, error handling, login, safety and performance optimisation

### 5.2.1 File version control

As a system of doing version control, like existing version control systems, the new system should records a list of history/versions of user's files when he/she submit every modifications, even though some special change such as add, rename or move a file. Each commit of a change will stored in the system as a new copy of the file, and the older version of it will still be kept in the system as a historical version for archive without affect, in case if a change of heavy mistake has been made, the user can return back to an older version at any time. With file versioning function, for personal user, he/she can use the system to make backup of each step of their file working in case of mistake editing, mistake deleting or losing files. He/she also can track modification history of file with in the system to do statistics after working.

For some kind of users who enjoy in working in different working environments, the system at least can be used as a platform to transfer the latest version of file to their computer at anywhere. User can commit a file modification for a new version, after that, they can download the new version no matter where they are.

To clearly identify each time of modification, user need to input some description to file when every time he/she commit it. At the same time, the file size and the commit time will be recorded in the file database. The type of operation will also can be identified and stored as an operation code (int type) in the database table, this is to reduce database storage and avoid unsynchronised name to be confused in the future.

Of course, each time of file commit will generate a new version number of this file. At first time of create the file, it version number will be "1", after any kind of modification at once, the version number will be increased by 1 then equal to "2", etc.

[Operation code table here](#)

In programming, several PHP functions as libraries were used for doing version control (see Appendix D, especially "project.lib.php"), including fetch information of project, task,

directory and file from database, receive uploaded file, compare it to older version, classify and convert file information, update related database record and store the file into physical hard drive at server side.

### 5.2.2 Task-oriented design

In comparing to the existing mode of treating file as a base version control unit, this project was designed the system as a “task-oriented” version control system. The definition of “task” within the system is a set of directories and files in a project, which means a project has several tasks, and a task contains several directories, and files belong to their directories. In a group work, task is the base unit of assignment to group members. If a group member has been assigned a task, files of the task can only be edited by this group member. Within the task, the member can create new file, rename file and changing file and directory structure in the task. For other members which not been assigned to a task, this task is read-only task, they can only download and use or view the files within the task, but cannot do any modification of it.

Each task has a name to be identified within the project. A start time and due time can be set to help scheduling tasks with time management. The status of task will be identified automatically. It includes “waiting for predecessor task”, “Task expired”, “Not yet start” and “Finished xx%”.

To make the task can be sorted by importance level, tasks has a property recording priority, which can be set from 1 to 5. 1 means most important, and 5 means this task can be did after tasks finished. User can change the priority of a task at any time, and it will displays at the table of task list for sorting purpose.

[File/directory task relationship chart here](#)

When a directory or file commits with an operation, and triggered with generating a new version code of the directory or file, the version code of the file’s or directory’s task will also increase by 1.

### 5.2.3 Task relationship

Task can be set with a property of “predecessor task”, which means a task can have a father task to be slaved with. If a task’s predecessor task has been set, the task can only be started when its predecessor task finished or expired. It like the relationship in Gantt chart, predecessor task and its sub task can be connected by end and head in showing the relationship. When setting a predecessor task for a task, it needs to judge by conditions of identify validity of relationship. For example, the due time of predecessor task should be earlier than the sub task’s due time, etc.

[Task relationship chart here](#)

In the implementation, the `project.lib.php` is also used for processing with task, directory and file, also relationship between them (see Appendix D). Functions in `project.lib.php` can deal with task predecessor set up and task status judgement by various conditions.

#### 5.2.4 Directory version and relationship

In this project, directory will be treated as a unit in version controlling. Any operation related to a directory including adding, renaming and moving will trigger a version record. The modification version records of directory are very similar to the version records of files, contain description of modification, type of operation and commit time.

For a multi-level directory, it is not stored in a single database record. A series of single-level directory records composed a multi-level directory. Each directory record in database has an attribute in recording its father directory. If the father directory filed exist in database, the directory will be connected its father directory together as a two-level directory; if its father directory still has a father directory, they will be connected together as a three-level directory, until the father level directory has not father directory, a full multi-level directory has been built. For example, the multi-level directory `/A/B/C` is composed by three single-level directory A, B and C, which have father directory of none, A and B. Each of them is an independent version control unit. They uses father directory attribute to join together.

[Directory relationship chart here](#)

The file `project.lib.php` is focusing on provide most functions related to file, directory, task and project, so the function to connect pieces of directories is also be placed in this file. Every PHP file which need to call the functions in `project.lib.php` can simply “include” the file name of it with its location “style/”.

#### 5.2.5 File storage

In this system, all the file modifications will be issued a unique “file change ID”. All the files uploaded into the system will be extracted by name and file size. The file name and file size will be stored in database along with the file change record by its unique generated ID, then the content of file will be stored in the “files” directory of the system’s physical driver. In order to make file easy to be stored and loaded, the files stored in physical storage at server side will be named as its unique ID, without any extension. For example, if a file has been uploaded by a user via file uploading page, a unique ID (FCID, see Figure 6) of the file version will be generated. If the FCID of this file change is 235, the file will be stored in the “files” folder as file name “235”.

[File storage chart here](#)

In implementation, the `file_operation.lib.php` in libraries folder has some functions controlling the file storage operations. And the `download.php` in root directory processes



the download of files by request, the file name can be restored to its original name when user upload it, by combining file contents in physical disk and real file name from related database record.

### 5.2.6 Error handling

Once error happened, if the error is a user level error (may be a mistake), the user will be provided an alert message with the reason of this error, and the user can back to the previous page to continue after correction it. If the error is at system level or caused by a bug, the user will be redirect to an error page with error information. At the same time, an e-mail includes error details will be sent to e-mail address of system administrator, which pre-defined in the configuration file.

The `error()` function in file `general.lib.php` will deal with serious errors. It will be wrote in the place of code where may be judge as a serious error. The `error()` function has a parameter of error information. If the function has been called with information by the parameter, the page will be redirected by the function to an error page with displaying the information, also sent the error e-mail to administrator in the background.

### 5.2.7 Login and Safety

To protect user's work, the system is forced require user sign in before use. A library file "identify.inc.php" is used for checking login status of users. This file has been "included" in all PHP files to be executed at their beginning part to make sure every time of operation can be protected by the login identify mechanism of both Session and Cookie detection. Once a operation has been detected as not yet sign in, the page will be forced redirected to the login page for user to sign in before continue his/her operation.

It is obviously that it is too boring to be requested to login at each time of visit after close and open the browser. To make login easier at next time use, the Cookie based automatic login mechanism has been designed and implemented in the file "login\_check.php". When a user ticked the "remember me" box at login page, a Cookie contains login information will be stored at client side browser cache area with expire time pre-defined in the configuration file. After the browser closed and the Session has been destroyed<sup>21</sup>, if there are no Cookie of login information exist, the user will be automatic logged out; if the login Cookie exist due to the "tick" at pervious sign in, the system will login by the information in Cookie automatically, and the Session information will be restored to keep the user's status as logged in before close browser or manual sign out.

For operation related to private project, the logged in user information will be carefully detected and compared to the owner information in record of related private project. If the user information meets, the operation can be executed as usual; if there is different user

---

<sup>21</sup> Lifetime of Session is only before the browser close. If the browser closed, all the Session will be destroyed automatically [87].

request operation of private project, the operation will be rejected and an error message will be displayed. The `project.lib.php` is responsible for place functions of protecting private project.

### 5.2.8 Performance optimisation

According to the requirements analysis related to response time, the performance of the system will affect the user experience directly. To give user better experience in doing operations within the system, the performance issue of the programme should be considered carefully. To enhance the performance of the system, in the programme part, the code related to the most time used operations such as database queries has be designed to reduced time using. If a page need to use information from a database table for many times, it may require very long time for processing. To reduce time wasted in this situation, the database tables will be only loaded for once and stored into a local data array. The repeat request of information at database table will be provided by data array instead, and the time of requesting data will be times quicker than repeating database selects, because the information in database is stored at physical hard drive and the data in data array is stored in RAM memory. The access speed of memory is faster than hard drive for more than 500 times [61]. So the performance will be increased very obviously. The file related to the performance improvement is the “`project.lib.php`” with its functions of selecting database and return two dimension data array for other file to be used with all the information related to the table by call for only once.

Even though the network bandwidth today is satisfactory, however there are some users are still using low speed internet access, especially mobile users and overseas users. To make sure users who connected via low speed internet connections, the page size of each time loading need to be reduced as little as possible. The use of CSS style sheet file will format the style of pages by a same file. So the CSS file can be cached for fast loading and apply to every page for its style. The system uses CSS for style design, so the formatting of each page only needs to call tags in a same CSS file, without separate codes, and can be loaded much faster

As the design of CSS is followed the guidance of W3C standard [62], it is very multifunctional. The traditional place of using images in formatting can be replaced in the latest standard of CSS, such as the background of table header and links with “mouse over action” can be did by CSS description instead of use images and JavaScript at before, and the page size at each time load will be reduced.

### 5.2.9 Table sort

There are many table will be displayed for users, such as project list, task list, directory list and file list. Some of the tables contains sortable attributes like priority, start time and version code. If the contents in the table can be sorted, user may feel easy in organising and scheduling his/her works. However, the existing HTML has not provide features in sorting table dynamically, even though the SQL command of database has parameter like “order by”,

it still request page refresh to make changes become effective, and the performance will be reduced by the times of database queries. As researched online, an open-source jQuery<sup>22</sup> plugin called Tablesorter [63] solves this problem perfectly. By using the open-source JavaScript library, the tables displayed in the system can be sorted by any column of data. User can sort file name, version code, date and whatever he/she would like to sort, and the change of sorting results will be displays at real-time, without page refresh.

### 5.2.10 Migration and modification

This system has been defined as can be used at any PHP and MySQL ready servers. However, running environment of the servers may be varying. Small company may run their MySQL database service at the same server of PHP running environment, but large companies may have separated PHP and MySQL server. Some of the system operator may like call the system as just WVCS, and some may prefer to call it begin with company name. To solve the problem of running environment difference, a configuration file has been designed and implemented for stores all the configuration information, such as database connection details, system name, login basis, cookie validate time, system time zone and terms and conditions. This design makes the administrators of the system can configure the system to run at their local servers much easier. They only need to change the values by requirements in the configuration file, without modification of any programming part.

Property	Example value	Description
db_server	localhost	Database server name or IP address.
db_name	wvcs	Database name.
db_username_read	wvcs_read	User name of database read-only account.
db_password_read	york	Password of database read-only account.
db_username_write	wvcs_write	User name of database updating account.
db_password_write	york	Password of database updating account.
administrator_email	sy595@cs.york.ac.uk	E-mail address of system administrator.
system_name	Web-based Version Control System	Name of the version control system.
system_name_short	WVCS	Short name of the system.
system_version	1.0	System version number.
login_by	email	Login by verify which information of user.
after_login_redirect	summary.php	Jump to page after login.
cookie_valid	60	Cookie lifetime (days).
date_default_timezone	Europe/London	Time zone applied to the system.

<sup>22</sup> jQuery is JavaScript library with great functions for dynamic effects of web pages [88].

e_set		
terms_conditions	WVCS Terms and Conditions...	Content of terms and conditions (with HTML tags).

**Table 2 - Configuration file information**

To make the system can be easier customised via secondary development, the programme has been designed and implemented by a library and style separately mode. The files of libraries and styles have been placed into two folders which name is libraries and styles. If the customer only needs to change the styles and formatting of the system, it only needs to modify the files in styles folder; if the customer needs to change features of the system, it needs to change both files in root directory and libraries folder. The code of files in both these two folder have been well commented, so it may much easier to find out which part of code is the place need to modify.

### 5.3 Database model

According to the requirements analysis of database, the database tables will be designed in the third normal form (3NF) of database normalization. The property of file and file changes will be recorded separately in the database, also the directory and directory changes, task and task histories. That is to make sure the table always is able in 3NF without redundancy to avoid “update anomaly”.

#### 5.3.1 Entity-relationship modelling

There are nine tables in the database design: user, project, task, task\_history directory, directory\_change, file and file\_change. Each of them has been linked by their “auto increase” primary keys and foreigner keys referrals.

The syntax of notations in the ERD is UML style notations.

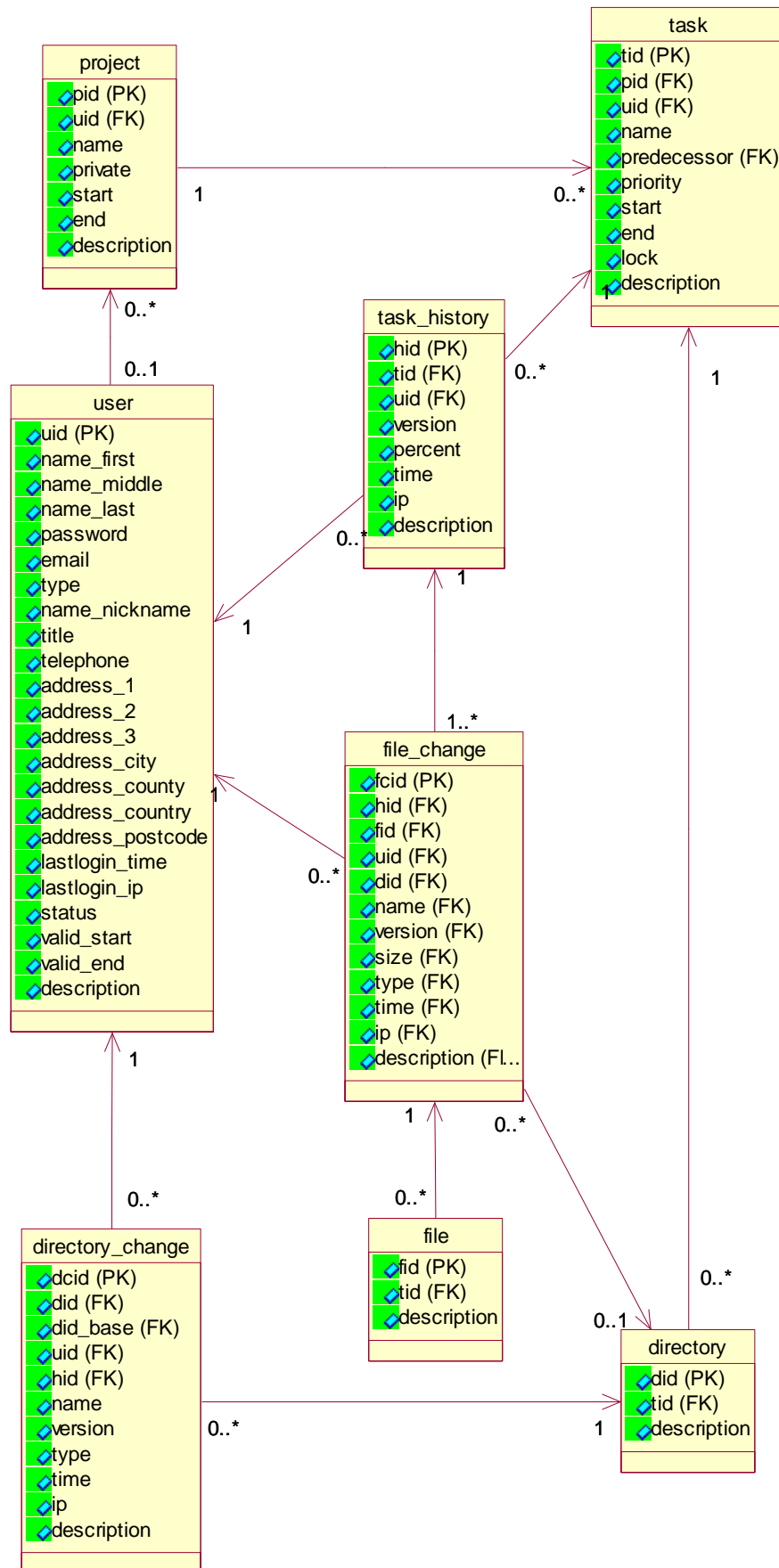


Figure 6 - Entity-relationship diagram

### 5.3.2 Attribute details of entities

In database table, all the fields<sup>23</sup> related to ID and numbers are all in integer type. Other text fields are in varchar type, with length of the maximum prediction the value would be. ID of each table is always in the first field. The details of the eight tables' attributes are listed below.

#### a. User

Field	Type	Null	Default	Comments
uid	int(8)	No		ID of user
name_first	varchar(100)	No		first name
name_middle	varchar(100)	Yes	NULL	middle name
name_last	varchar(100)	No		last name
password	varchar(50)	No		nick name
email	varchar(100)	No		email address
type	int(1)	No	1	type of user (0=admin, 1=regular)
name_nickname	varchar(50)	No		nick name
title	varchar(30)	Yes	NULL	title
telephone	varchar(30)	Yes	NULL	contact number
address_1	varchar(60)	Yes	NULL	address line 1
address_2	varchar(60)	Yes	NULL	address line 2
address_3	varchar(60)	Yes	NULL	address line 3
address_city	varchar(40)	Yes	NULL	city
address_county	varchar(40)	Yes	NULL	county
address_country	varchar(40)	Yes	NULL	country
address_postcode	varchar(15)	Yes	NULL	post code
lastlogin_time	datetime	Yes	NULL	last log-in time
lastlogin_ip	varchar(50)	Yes	NULL	last log-in IP address
status	int(1)	No	1	status (0=disabled, 1=active)
valid_start	datetime	No		register time or valid from time
valid_end	datetime	Yes	NULL	invalid time
description	varchar(8000)	Yes	NULL	description

<sup>23</sup> Field is another name of attribute in database entity.

Table 3 - Attribute details of user table

**b. Project**

Field	Type	Null	Default	Comments
pid	int(8)	No		ID of the project
uid	int(8)	Yes	NULL	ID of group leader
name	varchar(250)	No		project name
private	int(1)	No	0	identify private project or not
start	datetime	Yes	NULL	start time
end	datetime	Yes	NULL	due time
description	varchar(8000)	Yes	NULL	description

Table 4 - Attribute details of project table

**c. Task**

Field	Type	Null	Default	Comments
tid	int(8)	No		ID of the task
pid	int(8)	No		ID of project which the task belong to
uid	int(8)	No		ID of user which responsible to
name	varchar(500)	No		task name
predecessor	int(8)	Yes	NULL	ID of predecessor task
priority	int(1)	No	1	task priority
start	datetime	Yes	NULL	start time
end	datetime	Yes	NULL	due time
lock	int(1)	No	0	status of locked or not
description	varchar(8000)	Yes	NULL	description

Table 5 - Attribute details of task table

**d. Task\_history**

Field	Type	Null	Default	Comments
hid	int(12)	No		ID of this task history
tid	int(10)	No		ID of task which the history related to
uid	int(8)	No		ID of user who updated the task history
version	int(5)	No		version number of the history

Field	Type	Null	Default	Comments
percent	int(3)	Yes	NULL	percentage finished predict
time	datetime	No		time of the history submitted
ip	varchar(30)	Yes	NULL	IP log
description	varchar(8000)	Yes	NULL	description

Table 6 - Attribute details of task\_history table

*e. Directory*

Field	Type	Null	Default	Comments
did	int(11)	No		ID of the directory
tid	int(8)	No		ID of task which this directory belong to
description	varchar(8000)	Yes	NULL	description

Table 7 - Attribute details of directory table

*f. Directory\_change*

Field	Type	Null	Default	Comments
dcid	int(12)	No		ID of the directory modification
did	int(11)	No		ID of which directory is related to
did_base	int(11)	Yes	NULL	Upper level directory ID
uid	int(8)	No		ID of user who did this modification
hid	int(12)	No		ID of task change history record which the directory related to
name	varchar(259)	No		directory name
version	int(5)	No		version of this modification
type	int(1)	No		type of operation of the modification
time	datetime	No		time of commit
ip	varchar(30)	Yes	NULL	IP record of the modification
description	varchar(8000)	Yes	NULL	description

Table 8 - Attribute details of directory\_change table

*g. File*

Field	Type	Null	Default	Comments
fid	int(11)	No		ID of the file



Field	Type	Null	Default	Comments
tid	int(10)	No		ID of task which this file belong to
description	varchar(8000)	Yes	NULL	description

Table 9 - Attribute details of file table

#### h. File\_change

Field	Type	Null	Default	Comments
fcid	int(12)	No		ID of the file modification
hid	int(12)	No		ID of task change history record which this file related to
fid	int(11)	No		ID of the file record
uid	int(8)	No		ID of user who did this modification
did	int(11)	Yes	NULL	ID of which directory the file modification belong to
name	varchar(259)	No		file name in this modification
version	int(5)	No		version number
size	int(25)	No		file size
type	int(1)	No	0	type of operation
time	datetime	No		time record of the modification
ip	varchar(30)	Yes	NULL	IP log
description	varchar(8000)	Yes	NULL	description

Table 10 - Attribute details of file\_change table

## 5.4 Interface design

According to the requirement analysis of platform compatibility, there are lots of users still using screen with 1024 pixel wide. To make sure this kind of user can view pages without horizontal scrolling, after subtracting the width of browser border and vertical scroll bar on the right, the page width is sets at 960 pixels.

Generally, all the pages are divided into three parts: header, main body and footer. Header is responsible for the display of navigation bar (top part of page) and defining page properties; footer is used to presenting copyright information or other sentence need to be placed in the end of page.

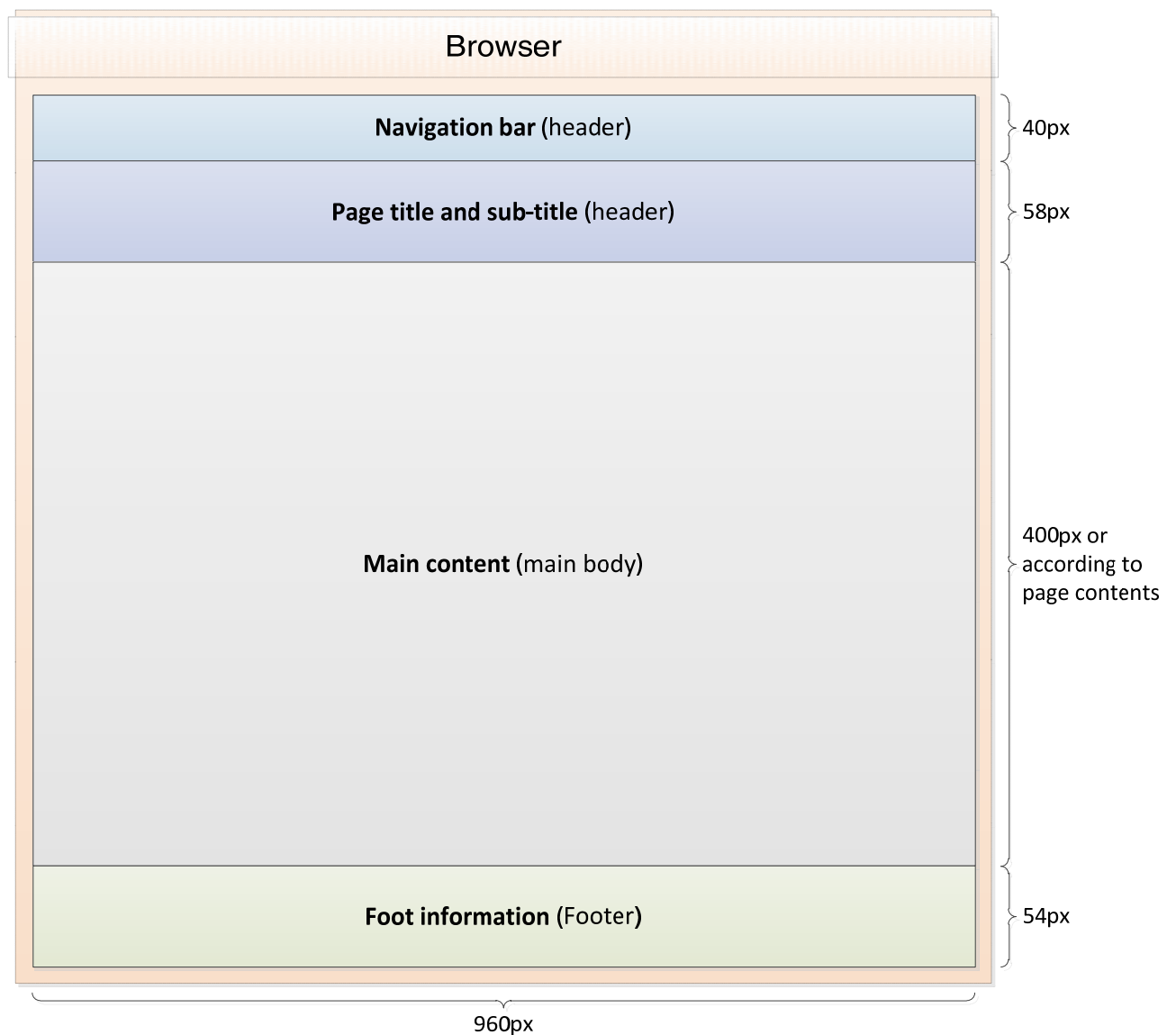


Table 11 - Page layout

[Colour matching](#)

[Style table](#)

## 5.5 Accessibility

WCAG 1.0 <http://www.w3.org/TR/WCAG10/>

H

P

Table with Thead

Image with alt

Less javascript use

White background high contrast

Use OL UL DL LI for list

Title attribute clear

比如按钮 用户需要反馈 颜色

## 5.6 Prototype design

[Prototypes here](#)

## 5.7 Prototype evaluation

[Prototype evaluation table here](#)

## 5.8 Prototype re-design

[Re-designed prototypes here](#)

## 5.9 Two layer PHP architecture

In the design of this system, the PHP files have been divided into two parts: function libraries and controllers. In controller part, the block of PHP code will be placed in the head of each PHP file, before output of HTML code. This architecture

## 5.10 Compatibility

## 5.11 CSS classes multiple use

twitter

To avoid the CSS file too large and can be load quickly without redundancy, in the system design, the use of CSS has been designed as fully multiple use mode. The CSS multiple use is a lesser known way to reduce CSS file size effectively, which means more than one CSS classes could be referenced in one HTML element at same time by split the class names by a space [64]. For example.

## **6. Evaluation**

### **6.1 Testing of version control**

### **6.2 Testing of administration**

### **6.3 Compatibility testing**

### **6.4 Stress testing and response time testing**

## 7. Conclusion

This project is a very interesting design about tracking works in computer based group and individual projects. “Task” as a new tracking unit has been firstly proposed and implemented in this area. The new fully web based interface and lightweight functions provides user a relaxed way in doing version control without excessive worry.

I am happy that I can learn lots of knowledge from the whole progress of the project. My personal capacity has been improved by constantly trying of solving problems during doing literature review, programming and writing project report.

After the evaluation, the system has been verified as running without obvious bugs. However, there are some work still needs to do in the future in making sure the system is able in continue serving users in the future. There are:

### 7.1 Easier local storage

### 7.2 Multi task assignment

### 7.3 Diff storage and analysis

[65]

### 7.4 Automatic merging

### 7.5 Branching support

### 7.6 Distributed system design

### 7.7 Multi-level administration

## References

- [1] G. Stamatellos, *Computer Ethics: A Global Perspective*, Jones and Bartlett, 2007.
- [2] UK Parliament, *Data Protection Act 1998*, London, 1998.
- [3] ACM Council, "ACM Code of Ethics and Professional Conduct," 16 Oct 1992. [Online]. Available: <http://www.acm.org/about/code-of-ethics>. [Accessed 2 Sep 2011].
- [4] B. Collins-Sussman, F. W. Brian and C. M. Pilato, *Version Control with Subversion*, O'Reilly, 2004.
- [5] B. Danella, "Rapid Subversion adoption validates enterprise readiness and challenges traditional software configuration management leaders," 15 May 2007. [Online]. Available: [http://www.open.collab.net/news/press/2007/svn\\_momentum.html](http://www.open.collab.net/news/press/2007/svn_momentum.html). [Accessed 10 July 2011].
- [6] D. Price, "CVS v1.11.23 Manual," Ximbiot LLC, 8 May 2008. [Online]. Available: <http://ximbiot.com/cvs/manual/cvs-1.11.23/cvs.html>. [Accessed 1 July 2011].
- [7] The Apache Software Foundation, "Apache Subversion Features," [Online]. Available: <http://subversion.apache.org/features.html>. [Accessed 1 July 2011].
- [8] T. Oakden, "None Concurrent Access in Version Control," 12 Oct 2009. [Online]. Available: <http://forum.unity3d.com/threads/36536-None-concurrent-access-in-version-control>. [Accessed 22 July 2011].
- [9] P. Roy, "Understanding Subversion's Problems," 9 Mar 2011. [Online]. Available: <http://ventspace.wordpress.com/2011/03/09/understanding-subversions-problems/>. [Accessed 22 July 2011].
- [10] Jbcrouigneau, "Task Oriented Development and Validation Space," 24 Apr 2009. [Online]. Available: <http://www.svnforum.org/threads/36840-Task-oriented-development-and-Validation-space?s=cf3b028492de3003320a35e609f4777b>. [Accessed 22 July 2011].
- [11] H. Gantt, *Work, Wages and Profit*, New York: The Engineering Magazine, 1910.
- [12] Shawn, "Which is More Popular (Currently, by Recent Install Base) SVN or CVS?," 23 Apr 2009. [Online]. Available: <http://stackoverflow.com/questions/782375/which-is-more-popular-currently-by-recent-install-base-svn-or-cvs>. [Accessed 23 July 2011].

- [13 C. Duan, "Understanding Git Conceptually," 17 Apr 2010. [Online]. Available:  
] <http://www.eecs.harvard.edu/~cdan/technical/git/>. [Accessed 23 July 2011].
- [14 T. Spencer, "Setup a Subversion Server in 4 Minutes," 2 Mar 2007. [Online]. Available:  
] <http://www.tonyspencer.com/2007/03/02/setup-a-subversion-server-in-4-minutes/>.  
[Accessed 23 July 2011].
- [15 "Setting Up Subversion," July 2006. [Online]. Available:  
] <http://systrhead.net/texts/200607subver.php>. [Accessed 23 July 2011].
- [16 D. Thomas and A. Hunt, Pragmatic Version Control Using CVS, Pragmatic Bookshelf,  
] 2003.
- [17 Google Inc., "Top ten advantages of Google's cloud," 2011. [Online]. Available:  
] <http://www.google.com/apps/intl/en/business/cloud.html>. [Accessed 24 July 2011].
- [18 L. L. Peterson and S. B. Davie, Computer Networks : A Systems Approach, Amsterdam;  
] London: Morgan Kaufmann, 2007.
- [19 Oracle, "Java Servlet Technology Overview," [Online]. Available:  
] <http://www.oracle.com/technetwork/java/overview-137084.html>. [Accessed 10 July  
2011].
- [20 M. Hall, "Building Web Apps in Java: Beginning & Intermediate Servlet & JSP Tutorials,"  
] 2011. [Online]. Available: [http://courses.coreservlets.com/Course-](http://courses.coreservlets.com/Course-Materials/csajsp2.html)  
[Materials/csajsp2.html](http://courses.coreservlets.com/Course-Materials/csajsp2.html). [Accessed 10 July 2011].
- [21 TechyShell.com, "ASP – Its Advantages and Disadvantages," 27 May 2009. [Online].  
] Available: <http://www.techyshell.com/internet/asp-its-advantages-and-disadvantages/>.  
[Accessed 13 July 2011].
- [22 The PHP Group, "The PHP License, version 3.01," 2010. [Online]. Available:  
] [http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt). [Accessed 11 July 2011].
- [23 Z:WAMP Group, "Z:WAMP Server Pack," 7 Nov 2010. [Online]. Available:  
] <http://zwamp.sourceforge.net/>. [Accessed 21 July 2011].
- [24 R. Bourdon, "WampServer," 24 Dec 2010. [Online]. Available:  
] <http://www.wampserver.com/en/>. [Accessed 21 July 2011].
- [25 E. Group, "EasyPHP," 2011. [Online]. Available:  
] <http://www.easyphp.org/introduction.php>. [Accessed 12 July 2011].
- [26 B. Shire, "PHP and Facebook," 3 May 2007. [Online]. Available:



- ] <http://www.facebook.com/blog.php?post=2356432130>. [Accessed 10 7 2011].
- [27 Daniel, "Benefits Of MySQL," 20 Nov 2010. [Online]. Available:  
] <http://benefitof.net/benefits-of-mysql/>. [Accessed 13 July 2011].
- [28 A. Cooper, R. Reimann and D. Cronin, About Face 3: The Essentials of Interaction Design,  
] Wiley, 2007.
- [29 H. Sharp, Y. Rogers and J. Preece, Interaction Design : Beyond Human-computer  
] Interaction, Chichester: Wiley, 2007.
- [30 J. Nielsen, "Usability 101: Introduction to Usability," [Online]. Available:  
] <http://www.useit.com/alertbox/20030825.html>. [Accessed 24 7 2011].
- [31 M. Miller, Cloud computing: Web-based applications that change the way you work and  
] collaborate online, Que, 2008.
- [32 Google Inc., "Boost productivity with Google-powered collaboration apps," 2011.  
] [Online]. Available: <http://www.google.com/apps/intl/en/business/collaboration.html>.  
[Accessed 27 July 2011].
- [33 S. Pichai, "Introducing the Google Chrome OS," 7 July 2009. [Online]. Available:  
] <http://googleblog.blogspot.com/2009/07/introducing-google-chrome-os.html>.  
[Accessed 27 July 2011].
- [34 S. G. Cohen and D. E. Bailey, "What Makes Teams Work: Group Effectiveness Research  
] from the Shop Floor to the Executive Suite," *Journal of Management*, vol. 23, no. 3, pp.  
239-290, 1977.
- [35 W. Chisholm, G. Vanderheiden and I. Jacobs, "Web Content Accessibility Guidelines  
] 1.0," 5 May 1999. [Online]. Available: <http://www.w3.org/TR/WCAG10/>. [Accessed 27  
July 2011].
- [36 G. Adams-Spink, "New guidelines boost web access," 22 Dec 2008. [Online]. Available:  
] <http://news.bbc.co.uk/1/hi/technology/7789622.stm>. [Accessed 24 July 2011].
- [37 S. Ahern, D. Eckles, N. Good, S. King, M. Naaman and R. Nair, "Over-Exposed? Privacy  
] Patterns and Considerations in Online and Mobile Photo Sharing," in *CHI '07: Proc. of  
the SIGCHI Conf., Human Factors in Computing Systems*, 2007.
- [38 J. Nielsen, "Response Times: The 3 Important Limits," 1993. [Online]. Available:  
] <http://www.useit.com/papers/responsetime.html>. [Accessed 27 July 2011].
- [39 R. B. Miller, "Response time in man-computer conversational transactions.," in *AFIPS*

- ] *Fall Joint Computer Conference*, 1968.
- [40 C. Heng, "Designing Your Website for Browser and Platform Compatibility," 5 Sep 2008.  
] [Online]. Available: <http://www.thesitewizard.com/archive/compatibility.shtml>.  
[Accessed 27 July 2011].
- [41 W3schools.com, "Browser Statistics," June 2011. [Online]. Available:  
] [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp). [Accessed 29 July 2011].
- [42 W3school.com, "Browser Display Statistics," Jan 2011. [Online]. Available:  
] [http://www.w3schools.com/browsers/browsers\\_display.asp](http://www.w3schools.com/browsers/browsers_display.asp). [Accessed 29 July 2011].
- [43 R. Adams, "JavaScript is Good, But Should Not be Relied Upon," 19 July 2009. [Online].  
] Available: <http://wblinks.com/notes/javascript-is-good-but-should-not-be-relied-upon>.  
[Accessed 29 July 2011].
- [44 CloudTweaks, "Cloud Computing – Demystifying SaaS, PaaS and IaaS," 3 May 2010.  
] [Online]. Available: <http://www.cloudtweaks.com/2010/05/cloud-computing-demystifying-saas-paas-and-iaas/>. [Accessed 20 June 2011].
- [45 OpenCrowd, "Software as a Service (SaaS)," 2010. [Online]. Available:  
] <http://cloudtaxonomy.opencrowd.com/taxonomy/software-as-a-service/>. [Accessed 20 June 2011].
- [46 S. Schwartz, "Just what is 'SaaS' and are some vendors abusing the term?," 21 Mar  
] 2011. [Online]. Available:  
<http://blog.connectedplanetonline.com/unfiltered/2011/03/21/just-what-is-saas-and-are-some-vendors-abusing-the-term/>. [Accessed 30 June 2011].
- [47 J. Brodtkin, "Gartner: Seven cloud-computing security risks," 2 July 2008. [Online].  
] Available: <http://www.infoworld.com/d/security-central/gartner-seven-cloud-computing-security-risks-853?page=0,0>. [Accessed 30 June 2011].
- [48 GPS Systems, "10 Reasons Why You Need SaaS Web-based Telematics," 12 Dec 2009.  
] [Online]. Available: <http://gpssystem.net/10-reasons-saas-webbased-telematics/>.  
[Accessed 30 June 2011].
- [49 W. Royce, "Managing the Development of Large Software Systems," in *IEEE WESCON*  
] 26, 1970.
- [50 S. McConnell, *Rapid Development: Taming Wild Software Schedules*, Redmond:  
] Microsoft Press, 1996.

- [51 B. W. Boehm, "A Spiral Model of Software Development and Enhancement," *Computer*, pp. 61-72, May 1988.
- [52 P. Smith, "File:Waterfall model (1).svg," 8 Mar 2009. [Online]. Available: [http://en.wikipedia.org/wiki/File:Waterfall\\_model\\_\(1\).svg](http://en.wikipedia.org/wiki/File:Waterfall_model_(1).svg). [Accessed 30 June 2011].
- [53 C. M. McClendon, L. Regot and G. Akers, "What is Prototyping?," 26 May 1999. [Online]. Available: <http://www.umsl.edu/~sauterv/analysis/prototyping/proto.html>. [Accessed 30 June 2011].
- [54 P. P.-s. Chen, "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems*, pp. 9-36, March 1976.
- [55 S. W. Ambler, "Data Modeling 101," [Online]. Available: <http://www.agiledata.org/essays/dataModeling101.html>. [Accessed 30 June 2011].
- [56 E. F. Codd, "Further Normalization of the Data Base Relational Model," IBM Research Report, New York City, 1971.
- [57 R. J. Rustin, *Data Base Systems: Courant Computer Science Symposia Series 6*, New York City: Prentice-Hall, 1972.
- [58 E. F. Codd, *The Relational Model for Database Management*, Addison-Wesley, 1990.
- [59 C. Zaniolo, "A New Normal Form for the Design of Relational Database Schemata," *ACM Transactions on Database Systems*, vol. 3, no. 7, 1982.
- [60 R. Elmasri, *Fundamentals of Database Systems*, London: Addison-Wesley, 2007.
- [61 E. Larsen, "Ram disk 500 times faster than hard drive," 22 Mar 2007. [Online]. Available: <http://www.computeractive.co.uk/pcw/news/1922962/ram-disk-500-times-faster-hard-drive>. [Accessed 1 Aug 2011].
- [62 W3Schools.com, "CSS Tutorial," 2011. [Online]. Available: <http://www.w3schools.com/css/>. [Accessed 1 Aug 2011].
- [63 C. Bach, "Tablesorter: Flexible client-side table sorting," 17 Mar 2008. [Online]. Available: <http://tablesorter.com/docs/>. [Accessed 1 Aug 2011].
- [64 J. Kyrnin, "Use Multiple CSS Classes on a Single Element," 2011. [Online]. Available: <http://webdesign.about.com/od/css/qt/tipcsmulticlas.htm>. [Accessed 26 July 2011].

- [65 K. Azad, "A Visual Guide to Version Control," 27 Sep 2007. [Online]. Available:  
] <http://betterexplained.com/articles/a-visual-guide-to-version-control/>. [Accessed 30 8  
2011].
- [66 Oracle, "JavaServer Pages Technology," [Online]. Available:  
] <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>. [Accessed 2 July  
2011].
- [67 S. Chacon, "Git - The Fast Version Control System," 2011. [Online]. Available: <http://git-scm.com/>. [Accessed 23 July 2011].
- [68 J. Loeliger, Version Control with Git, O'Reilly, 2009.  
]
- [69 R. M. Hinden and B. Haberman, "RFC 4193 : Unique Local IPv6 Unicast Addresses," Feb  
] 2005. [Online]. Available: <http://tools.ietf.org/html/rfc4193>. [Accessed 24 July 2011].
- [70 Y. Rekhter, R. G. Moskowitz, D. Karrenberg, G. J. d. Groot and E. Lear, "RFC 1918 :  
] Address Allocation for Private Internets," Feb 1996. [Online]. Available:  
<http://tools.ietf.org/html/rfc1918>. [Accessed 24 July 2011].
- [71 P. Mockapetris, "The Domain Name System," in *Proceedings of the IFIP 6.5 Working  
] Conference on Computer Message Services*, Nottingham, 1984.
- [72 P. Mockapetris, J. Postel and P. Kirton, "Name Server Design for Distributed Systems," in  
] *Proceedings of the Seventh International Conference on Computer Communication*,  
Sidney, 1984.
- [73 P. Mockapetris, "RFC 882 : Domain Names - Concepts and Facilities," Nov 1983.  
] [Online]. Available: <http://tools.ietf.org/html/rfc882>. [Accessed 24 July 2011].
- [74 P. Mockapetris, "RFC 883 : Domain Names - Implementation and Specification," Nov  
] 1983. [Online]. Available: <http://tools.ietf.org/html/rfc883>. [Accessed 24 July 2011].
- [75 D. Parrack, "New Twitter.com UI is faster, better," 15 Sept 2010. [Online]. Available:  
] [http://tech.blorge.com/Structure:%20/2010/09/15/new-twitter-com-ui-is-faster-  
better/](http://tech.blorge.com/Structure:%20/2010/09/15/new-twitter-com-ui-is-faster-better/). [Accessed 25 July 2011].
- [76 Microsoft Corporation, "ASP.NET Web Pages," 2011. [Online]. Available:  
] <http://www.asp.net/web-pages>. [Accessed 2 July 2011].
- [77 A. Sami, "What is New in ASP.NET 4.0, Visual Studio 2010 IDE," 13 Jan 2010. [Online].  
] Available: [http://www.codeproject.com/KB/aspnet/Whatis\\_New\\_ASP\\_Net\\_4.aspx](http://www.codeproject.com/KB/aspnet/Whatis_New_ASP_Net_4.aspx).

[Accessed 2 July 2011].

- [78 C. D. Knuckles and D. S. Yuen, Web Applications: Concepts & Real World Design, Hoboken, N.J.: John Wiley & Sons, Inc., 2005.
- [79 D. Robinson and A. L. K. Coar, "RFC 3875: The Common Gateway Interface (CGI) Version 1.1," Oct 2004. [Online]. Available: <http://tools.ietf.org/html/rfc3875>. [Accessed 10 7 2011].
- [80 The Apache Software Foundation, "Apache HTTP Server Project," 2011. [Online]. Available: <http://httpd.apache.org/>. [Accessed 13 July 2011].
- [81 Igor Sysoev, "Nginx," [Online]. Available: <http://nginx.org/en/>. [Accessed 13 July 2011].
- [82 Microsoft Corporation, "IIS: Overview," 2011. [Online]. Available: <http://www.iis.net/overview>. [Accessed 13 July 2011].
- [83 J. Lee and B. Ware, Open source Web development with LAMP using Linux, Apache, MySQL, Perl, and PHP, Boston: Addison-Wesley, 2003.
- [84 Oracle Corporation, "MySQL Standard Edition," 2010. [Online]. Available: <http://www.mysql.com/products/standard/>. [Accessed 13 July 2011].
- [85 TutorialsPoint.COM, "Database - Second Normal Form (2NF)," 2011. [Online]. Available: <http://www.tutorialspoint.com/sql/second-normal-form.htm>. [Accessed 22 July 2011].
- [86 F. Spillers, "How Usable is Jakob Nielsen?," 7 Apr 2004. [Online]. Available: [http://experiencedynamics.blogs.com/site\\_search\\_usability/2004/04/how\\_usable\\_is\\_j.html](http://experiencedynamics.blogs.com/site_search_usability/2004/04/how_usable_is_j.html). [Accessed 21 July 2011].
- [87 The PHP Group, "Session Handling," 20 July 2011. [Online]. Available: <http://www.php.net/manual/en/book.session.php>. [Accessed 31 July 2011].
- [88 "Documentation of jQuery," 2010. [Online]. Available: [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page). [Accessed 1 Aug 2011].
- [89 T. Chahine, "Low-fidelity prototyping – an overview of tools," 8 May 2011. [Online]. Available: <http://aixd2011.com/2011/05/08/low-fidelity-prototyping-an-overview-of-tools/>. [Accessed 30 June 2011].
- [90 G. Booch, J. Rumbaugh and I. Jacobson, The Unified Modeling Language User Guide, London: Addison-Wesley, 2005.

[91 T. M. Connolly, Database systems : a practical approach to design, implementation, and  
] management, Harlow: Addison-Wesley, 2002.

## Appendices

### A. Set-up guide

#### A.1 Environment requirement

#### A.2 Step-by-step guide of installation at server side

### B. Questionnaire for requirement analysis

### C. Interface samples

### D. File structure

### E. Key source code

#### D.1 Libraries

#### D.2 Styles

#### D.3 Pages