

國立臺灣科技大學資訊工程系

一百零六學年度第一學期專題研究

總報告

基於 Git 的共享創作平臺 WeArt

研究組員

李大祥

張天佑

指導教授: 謝仁偉

中 華 民 國 106 年 1 月 4 日

致謝

在長達一年的專題研究中，學習了太多的知識、經驗和方法，收益良多。

首先要感謝的是我們專題指導老師 — 謝仁偉教授。老師在這漫長的時間裏給予了我們細致耐心的指導，準確地指出了我們專題學習中存在的很多問題，並提供了許多有效的解決方案。非常感謝教授一年來的教導。

在設計與開發 WeArt 期間，我們組員之間產生了不少爭議，在產品的設計理念、功能需求的取舍、時間衡量上有過很多的分歧。但通過共同商討、共同探究，商榷很久後幾乎都得到了比較完善的解決。也許每一個具有新理念的作品產生過程中都包含了各種各樣的曲折。最後作品能夠基本完成，要感謝我們組的成員的努力付出。

最後也感謝系上給予的支持。謝謝你們。

目錄

致謝

1 背景介紹.....	1
2 動機.....	1
2.1 傳統文學創作方法.....	1
2.2 網路文學創作方法.....	2
2.3 缺點與改進.....	2
3 專題特色.....	3
4 系統設計.....	4
4.1 系統架構.....	4
4.2 功能結構.....	4
5 資料庫設計.....	5
6 系統實現.....	7
6.1 開發環境部署.....	7
6.2 開發流程.....	8
6.3 實現效果.....	9
7. 結論.....	12
8. 參考文獻.....	12

1 背景介紹

近十年來，移動性、便攜式的產品大賣，以 iPhone 為代表的智能手機開啓了移動互聯網的時代，曾經佔總人口比例小眾的網民突然之間成為了大眾。PC 時代裏面臨的市場小、數據量少的問題，在億萬級的移動網路用戶以及數以百萬計的 APP 面前，都不再是問題。反而現在面臨的是用戶的需求得不到滿足，而不再是好產品找不到足夠大的市場。

開始這個專題前，正值共享理念的迸發，陸續出現了各種基於互聯網的共享資源平臺，這些平臺把人們閒置的資源——從汽車房子到普通的工具，轉化為新的工作機遇和收入，讓人們有機會成為微型創業者，同時重建自己與社區的連結。共享經濟的代表包括了早期的維基百科和 Github。維基百科能讓任何注冊的人參與到對同一篇文章的編輯和修改中，也能免費開放給普通人去建立新的頁面；Github 是一種主要用於控制代碼版本更迭的工具，提供了免費的 Git 遠程倉庫。

新時代裏另外一個代表是直播行業。直播行業的興盛發展使得往日裏看似高高在上的明星人物瞬間變得可以親近，甚至觀眾與明星之間的界限不再像以前那樣橫在那裏，讓人生畏。這代表了一個社會的潮流，即用最接近民衆的方式獲取最即時的用戶反饋，才能讓自己在市場競爭中獲得足夠多的主動權。而現代市場的恆定規律是“贏得市場即贏得一切的觀點”。

2 動機

2.1 傳統文學創作方法

在我們專題的設想中，我們由自己平時生活愛好而注意到文學領域。自古以來，文學創作這個行業就經久不衰。從詩詞歌賦到近現代的小說，以及

互聯網時代的博客，文學讓我們足不出戶的了解外面世界，認識志同道合的陌生人的思想，娛樂生活，豐富自己的思想世界。

傳統的紙筆寫作，一筆一劃的寫文字，耗時耗力，而且一旦思路不暢或者有了新的想法也可能會扔掉以前的書稿，但過了幾天可能又想找回來。除此之外，這種創作還可能面臨容易被盜竊知識財產、丟失遺棄正稿的風險，具備傳播緩慢的缺點。從作者開始寫書到正式出版要很久的時間，從正式出版到聞名天下又要耗時耗錢的宣傳。

2.2 網路文學創作方法

互聯網時代裏在網上寫小說並公開的網路文學的創作方法雖然一定程度避免了上述傳統文學創作的缺點，但網路文學作品受到作者個人因素影響極大。比如一位兼職作者換了一個繁忙的工作會導致沒時間繼續寫一本優秀的作品，某段時間作者心情不好會導致作品質量的下降等等。這些問題也一定程度上影響了網路文學創作的形象。

另外，在移動互聯網時代中，各種各樣的自媒體孕育而生，為了迎合大眾口味，這些視頻文字都會非常短小，其中觀點片面化較多，營養價值較小。那麼對於博客來說，不少人覺得博客的架構限制了文學作品的創作，追蹤不同人的博客對於一般用戶來說也是一件很麻煩的事情。

2.3 缺點與改進

誠然，當下流行的寫作網站能夠提供作家與讀者更自由的創作環境與更方便的閱讀途徑，為廣大熱愛文學的人們提供優質的平台。現在流行的小說網絡創作網站例如“起點中文網”已經引領了互聯網時代的文學創作，但同時我們在長期使用類似產品與參與寫作中也注意到不少缺點。比如這些創作平臺功能越來越煩雜，功能越來越臃腫，有的好作品卻無人問津以致作者最後放棄，有的好想法被寫成爛故事等等。此外，現有的創作平臺也把

作者與讀者分離成了兩個群體，實際上並沒有像直播產業那樣很好地利用互聯網及時緊密連接你我他的特性。

我們的初衷就是讓大家享受自由的創作，可以讓更多文學愛好者參與其中，由個人可以個人自主創作，也可以參與其他人的創作，即便彼此互不相識的陌生人也可以同時對同一章節進行創作。不僅讓原創者有更多靈感來源，也讓閱讀者擁有自己改變故事的能力。關於如何產生好的作品，這個權力我們還給了讀者，由讀者投票決定。

3 设计特色

上述幾種文學創作的方法都是基本遵循個人創作自己作品的原則，面臨的風險和缺點不一，他們的創作方法和作品就像圖 2 的椰子樹，只有一根光禿禿的樹幹往上走，堅持到最後才能看到葉子和結出果實。

而我們的平臺就像圖 1 的普通樹木，樹幹不長保證所有人能參與進來，樹叉很多，可以滿足不同人對於同一個故事創意的不同解釋與創作。同一個故事也可以有不一樣的結局和發展線。

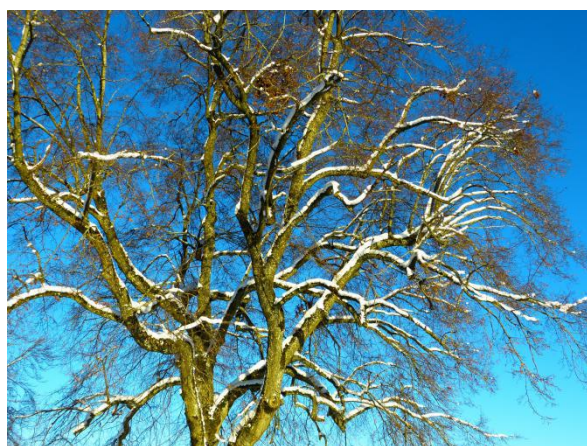


圖 1: [來源鏈接](#)



圖 2: [來源鏈接](#)

4 系統設計

4.1 系統架構

下面的圖是我們的系統架構示意圖。User 通過網路訪問我們的 Web Server，Web Server 裏面的 Python 程式會對請求做處理與響應，這些處理包括對 Web Server 裏面 Database 的讀寫以及對 Git Server 的二次請求。

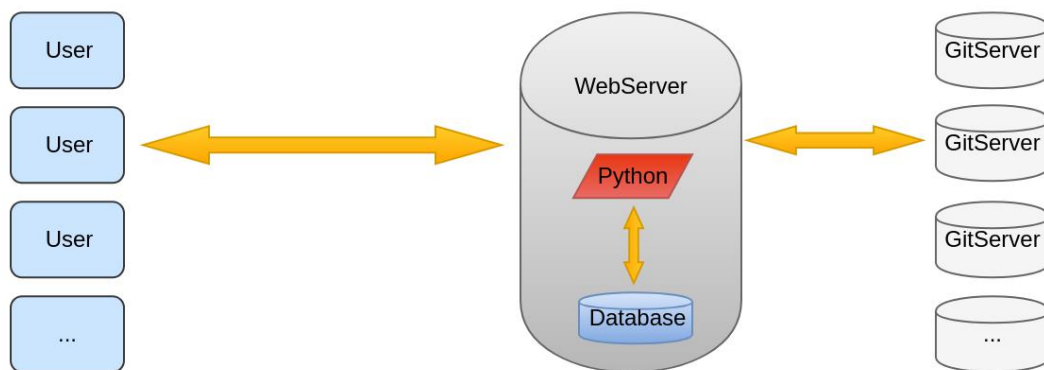


圖 3

Git Server 的部分是實現我們功能與目標的關鍵。它會存儲書的各個章節、各個版本，它扮演了 Github 的角色，它擔負着大量資料的存儲與版本控制的重要功能、特色功能。

從圖 3 我們看到，每一個 User 發送的請求都將在 Web Server 中做處理，處理後的請求轉換為 Git 指令與 Git Server 交互操作，之後 Git Server 將返回數據給 Web Server 對 Database 更新，同時刷新 User 的可視界面。

4.2 功能結構

圖 4 是我們系統功能結構的示意圖，演示了不同角色可能需要面臨的需求。

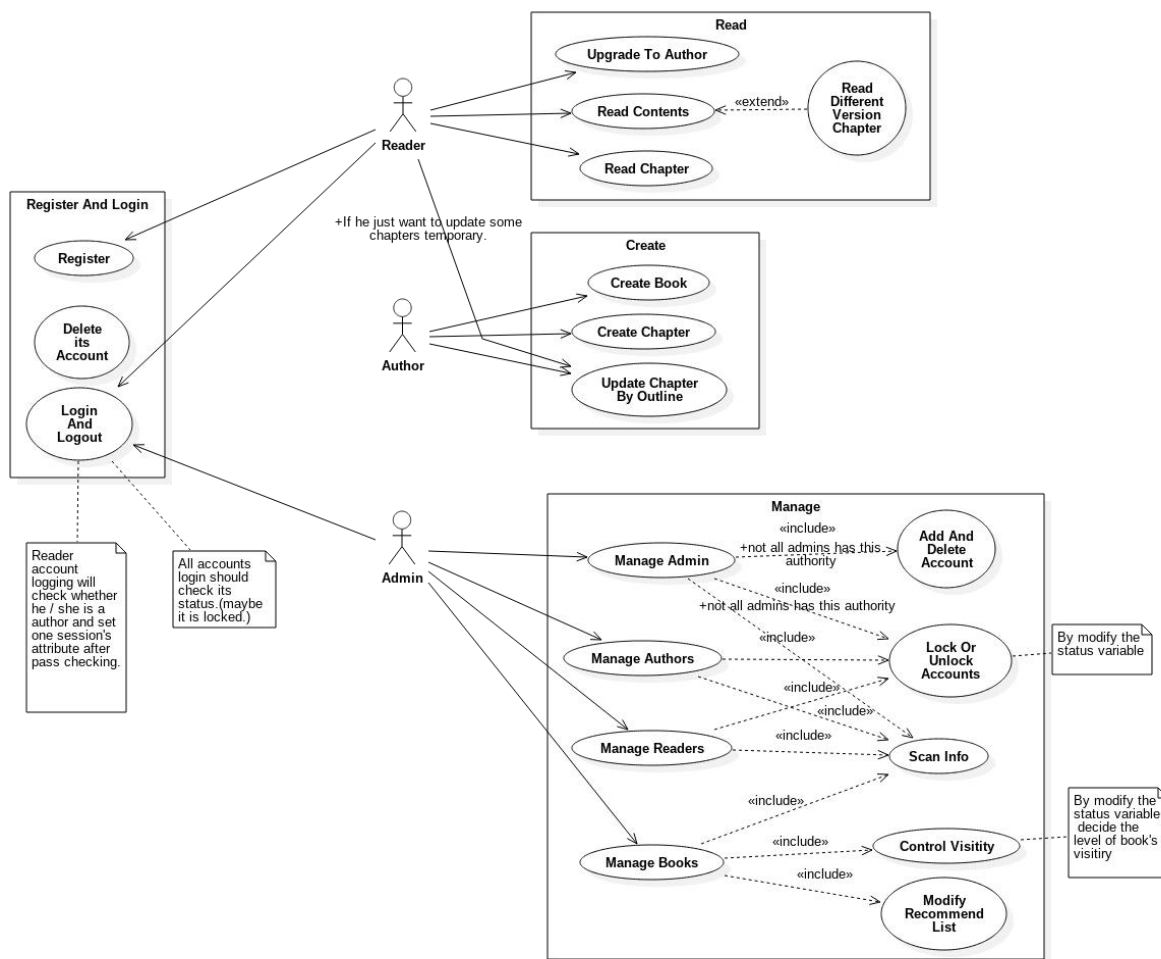


圖 4

例如一個用戶開始時候只是想看書，後來這個讀者慢慢看到有些感興趣的或者感覺作者沒寫好的章節想動手寫作，這個時候他/她可以選擇啓用作者功能，然後參與到某一本書或者某一個章節的寫作中。

例如管理員應該有控制用戶賬戶權限、管理書本的可看狀態以及推薦書的功能。

5 資料庫設計

根據功能所需我們利用 Mariadb 建立圖 5 中所示的九張表：

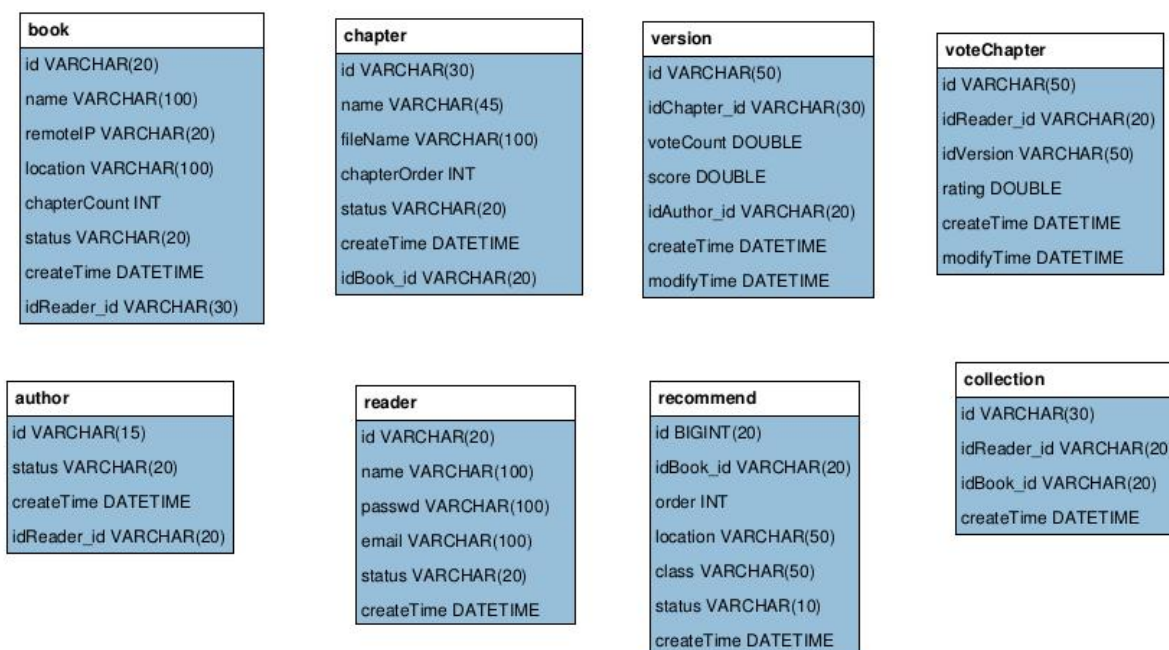


圖 5

(1) voteChapter: 該表中屬性有章節 ID，讀者 ID，章節版本號，投票分數，創建及更新時間。

(2) version: 該表中屬性有版本 ID，章節號，投票次數，分數，創作者 ID，創建及更新時間。

(3) chapter: 該表中屬性有章節 ID，章節名稱，章節全名，章節序號，當前狀態，創建及更新時間。

(4) author: 当启用作者功能后，該表中会储存作者 ID，狀態，創建及更新時間。

(5) reader: 該表中屬性有普通用戶 ID，姓名，密碼，電子郵件，用戶狀態，創建及更新時間。

(6) book: 該表中屬性書籍 ID，書籍名稱，存儲該本書的 Git Server 的 IP 地址，具體存儲在 Git Server 裏的檔案夾位置，章節數量，當前狀態，創建及更新時間。

(7) recommend: 該表中屬性有書籍 ID, 次序, 文件夾位置, 類別, 當前狀態, 創建時間。

(8) collection: 收藏 ID, 普通用戶 ID, 書籍 ID, 創建時間。

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
password	varchar(128)	NO		NULL	
last_login	datetime(6)	YES		NULL	
is_superuser	tinyint(1)	NO		NULL	
username	varchar(150)	NO	UNI	NULL	
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	NO		NULL	
email	varchar(254)	NO		NULL	
is_staff	tinyint(1)	NO		NULL	
is_active	tinyint(1)	NO		NULL	
date_joined	datetime(6)	NO		NULL	

圖 6

(9) auth_user: 管理員 ID, 密碼, 最後一次登錄時間, 是否是超級管理員用戶, 昵稱, 名, 姓, 注冊郵箱, 是否是員工, 是否可用, 創建時間。本表是通過 Django 的命令創建的, 主要用於管理員的資料存儲。

6 系統實現

6.1 開發環境部署

基本開發環境:

作業系統	Ubuntu16.04 & macOS
程式語言	Python2.7, HTML, CSS, JavaScript
Library & Framework	Django1.8, Bootstrap3.0, jQuery

這裏的部分是羅列了一些可能需要特別注意的軟體或者框架安裝：

安裝機器	安裝命令類型	
Git Server	apt	git2.11 以上版本
Web Server	apt	Nginx, Python2.7, Mariadb, firewalld, sshpass, libmysqlclient-dev
	pip	django, gitpython, paramiko, itsdangerous, mysqlclient, django_extensions, django_dia

另外一些環境部署時候的注意事項：

(1) 需要確保 Web Server 与 Git Server 是可以無密碼通過 SSH 连通的。

(2) 使用此命令建立 Django 的 Database 中的緩存表：

```
python manage.py makemigrations  
python manage.py migrate
```

(3) 使用此命令創建管理員帳號：

```
python manage.py createsuperuser
```

(4) 使用此命令開啓 Web Server：

```
python manage.py runserver 0.0.0.0:8000
```

(5) 如果要採取企業級服務器的標準，則需要使用 Nginx + Supervisor + uWSGI 聯合調整相關配置。

6.2 開發流程

開發流程基本按照功能的重要性排列，越是重要的功能越是會先開發。圖 6 是使用甘特圖表示我們本學期主要的開發過程。

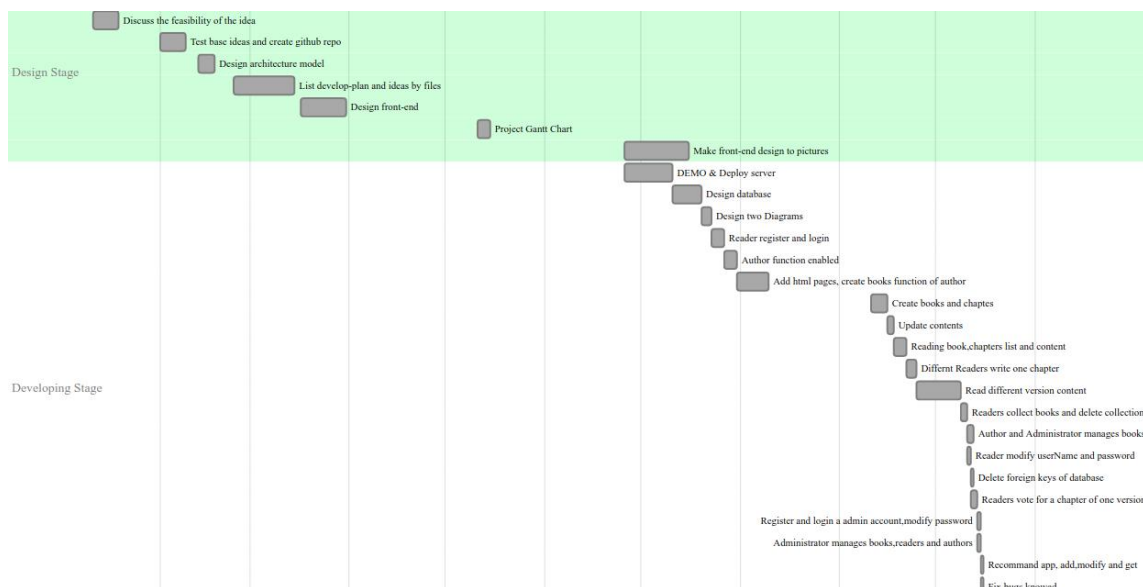


圖 7

6.3 實現效果

我們的實現效果如下面幾張圖所示。圖 8 是首頁的截圖效果，會有一個簡單的介紹使用以及推薦書籍列表。

圖 9 和圖 10 是閱讀章節界面，同時可以參與對這一章節的投票以及參與對這一章節的寫作，這裏的寫作不用開啓作者功能。同一個章節會列出來所有參與這個章節寫作的作者們，點擊作者名字可以查看不同作者的版本內容。當這個頁面第一次進入時，展示的是讀者投票最高的一個作者寫的版本。

圖 11 是讀者和作者的個人中心。這裏有讀者的收藏書籍列表，有修改昵稱和密碼的界面，還有完整的作家專區功能。當然只有當用戶選擇啓用了作者功能，作家專區才會開放相關創建書、編輯章節等界面。



熱門精選

序號	書名	作者昵稱	章節數量
1	123	0	1



圖 8



圖 9



圖 10

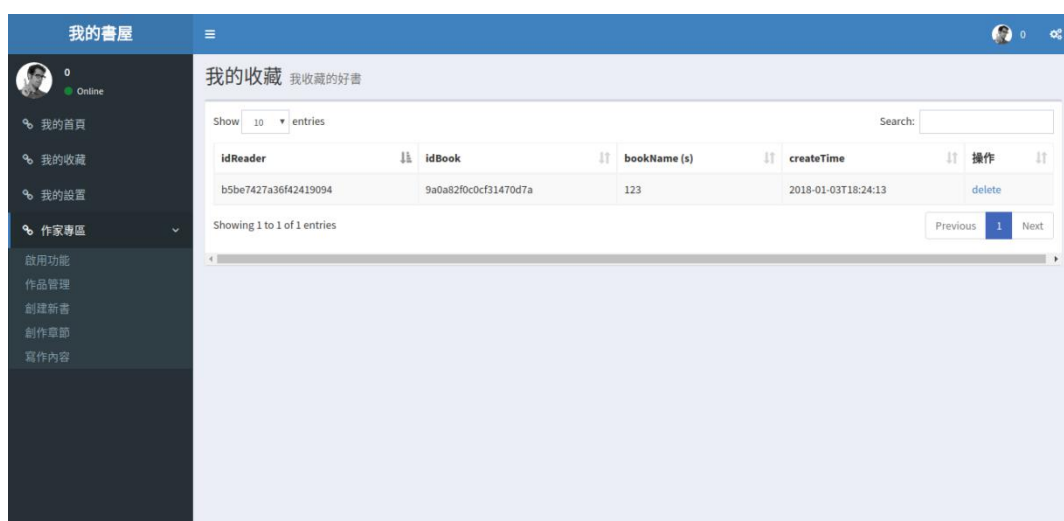


圖 11

7 結論

在最初的設想中，我們希望創造一個新的空間，讓好作品能被更多人知道，讓一件作品不會因為作者臨時有事放棄而終結，讓好的想法可以不被作者寫作功力限制。受到維基百科、共享單車、Github 的啟發，我們決定讓這個平臺可以讓多人參與寫同一件作品，我們設計了一個投票機制保證讀者決定好作品而不是資本決定，我們補充了推薦書籍的機制避免好作品被埋沒。現如今我們的作品已初步實現我們所設想的共同創作理念。

如果在未來如果有機會，需要改進的是這樣的共享平台是否從中英文創作拓展到其他語言以及繪畫領域。另外在技術方面，可能還需要類似 Celery 這種技術去對耗時比較久的寫入操作進行序列化，以防止多人對同一本書讀寫操作時出錯；還有需要 Redis 這種技術去做一個 Database 的緩存，以加快讀取 Database 的速度和減少網頁訪問時間；對於 Git Server 層，需要額外開發一套監控系統去監控每個 Git Server 的存儲情況以及 Web Server CPU 以及存儲空間的情況。

最後也再次感謝指導老師的教導和組員的努力付出，謝謝你們。

8 參考文獻

<https://github.com/almasaeed2010/AdminLTE>
<http://toyroom.bruceli.net/tw/2011/02/04/install-git-server-on-ubuntu-linux.html>
<http://www.cnblogs.com/baiyangcao/p/gitpython.html>
<https://git-scm.com/book/>
<https://www.linux.com/learn/how-run-your-own-git-server>
<https://www.djangoproject.com/>
<http://www.runoob.com/django/django-tutorial.html>
<http://www.cnblogs.com/xkops/p/6234554.html>
<https://www.digitalocean.com/community/tutorials/how-to-serve-django-applications-with-uwsgi-and-nginx-on-centos-7>
<https://www.vultr.com/docs/how-to-install-django-on-centos-7>
<https://stackoverflow.com/questions/2497051/how-can-i-show-the-contents-of-a-file-at-a-specific-state-of-a-git-repo>