

เอกสารประกอบการติดตั้งและการใช้งาน Git

1.1 Setting up Git

ก่อนที่นักศึกษาจะเริ่มใช้งาน Git ในโครงการต่าง ๆ ได้นั้น สิ่งที่นักศึกษาต้องทำเป็นอันดับแรกคือ การติดตั้ง Git ขั้นตอนต่อไปนี้เป็นกระบวนการติดตั้ง Git

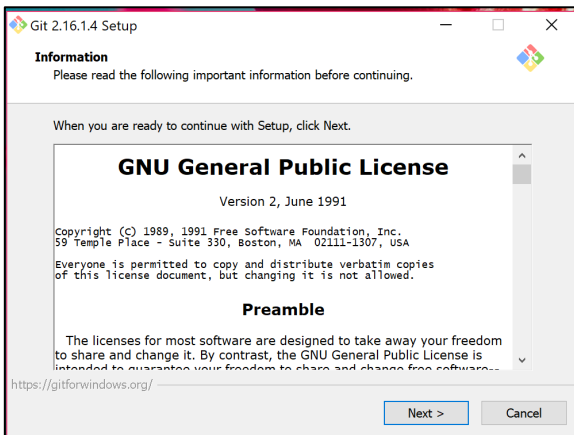
ขั้นตอนที่ 1: เข้าไปที่เว็บไซต์ <http://git-scm.com/>

ขั้นตอนที่ 2: คลิกที่ "Download for Windows"

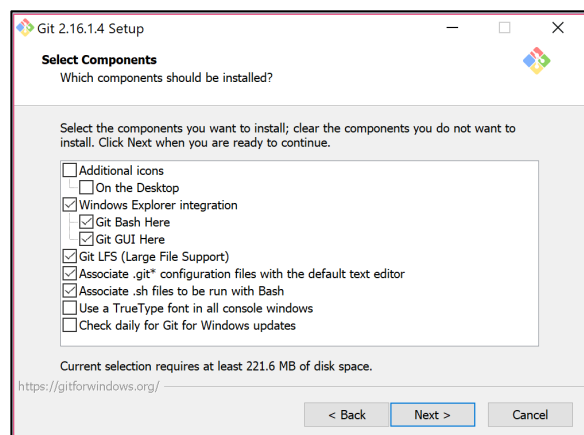
ขั้นตอนที่ 3: ดับเบิลคลิกไฟล์ที่ดาวน์โหลดมา

ขั้นตอนที่ 4: ทำตามขั้นตอนของ Git setup wizard ดังแสดงในภาพที่ (1)-(6)

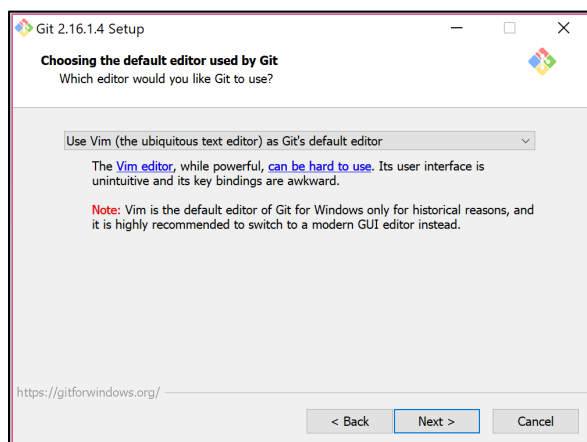
(1)



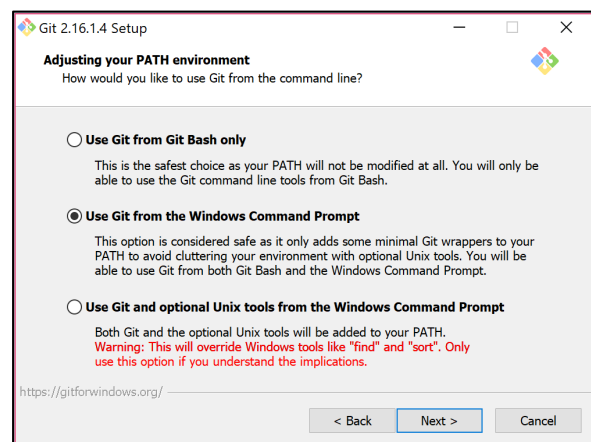
(2)



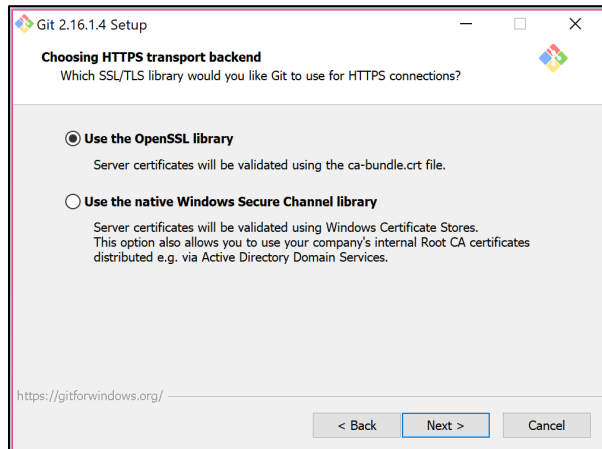
(3)



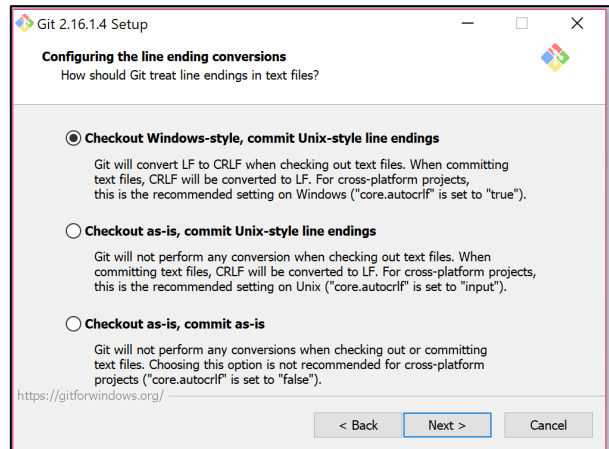
(4)



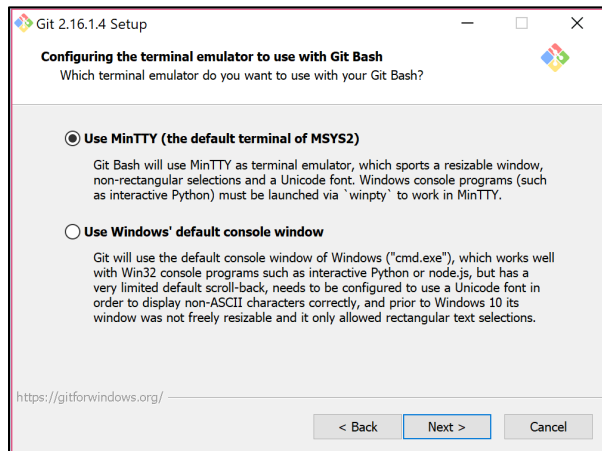
(5)



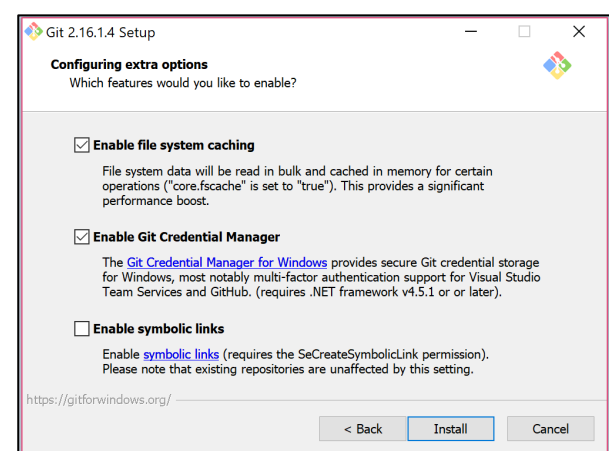
(6)



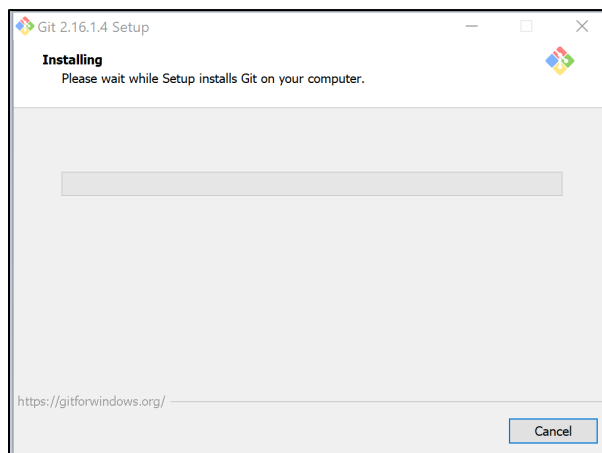
(7)



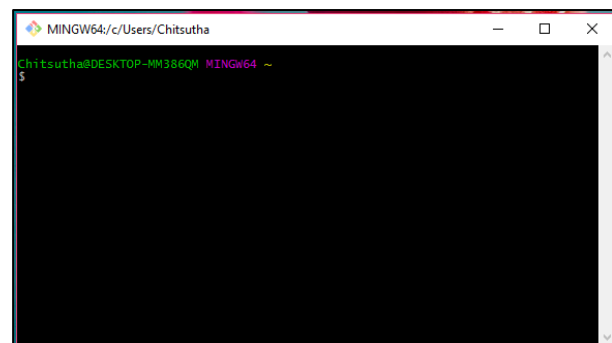
(8)



(9)



(10)



Git ที่นักศึกษาได้ติดตั้งสำเร็จเรียบร้อยแล้วนั้นสามารถถูกเข้าถึงผ่าน Command line หรือ Git Bash ได้ ซึ่งเป็นวิธีการที่ไม่สะดวกนัก เราสามารถใช้ Git GUI เข้ามาช่วยให้เราสามารถใช้งาน Git ได้อย่างสะดวกสบายมากขึ้น เช่น SourceTree ซึ่งเป็นเครื่องมือที่ได้รับความนิยมตัวหนึ่ง

1.2 คำสั่งพื้นฐานในการใช้งาน Git

คำสั่ง	การใช้งาน
git init	คำสั่งเริ่มต้นใช้งาน Git ให้เข้ามาควบคุมเวอร์ชันของไฟล์และโฟลเดอร์ในตำแหน่งที่อยู่ปัจจุบัน โดยจะสร้างโฟลเดอร์ .git ขึ้นมา ซึ่งเป็น Hidden folder โดยที่โฟลเดอร์นี้จะเก็บข้อมูลต่าง ๆ ของ Git repository เอาไว้
git status	คำสั่งที่ใช้ในการดูว่ามีเปลี่ยนแปลงเกิดขึ้นหรือไม่
git add <file name>	เพิ่มไฟล์หรือโฟลเดอร์ที่สร้างขึ้นใหม่ภายใต้ Working directory เข้าไปยัง Git repository
git add .	เพิ่มไฟล์หรือโฟลเดอร์ทั้งหมดที่อยู่ภายใต้ Working directory เข้าไปยัง Git repository
git add *.นามสกุลของไฟล์	เพิ่มไฟล์หรือโฟลเดอร์ทั้งหมดที่นามสกุลตรงกับที่ระบุที่อยู่ภายใต้ Working directory เข้าไปยัง Git repository
git commit -m '<comment message>'	เป็นคำสั่งเพื่อยืนยันไฟล์ที่มาจากคำสั่ง add เพื่อเข้าไปเก็บใน Git repository จริงๆ พร้อมกับการใส่คำอธิบายการเปลี่ยนแปลงที่เกิดขึ้น
git log	เป็นคำสั่งเพื่อดูประวัติการ Commit ต่าง ๆ ใน Git repository
git summary	ดูสถานะปัจจุบันของ Git
git checkout <branch name>	เป็นคำสั่งสำหรับเปลี่ยนการใช้งานไปยัง Branch นั้น ๆ
git branch <branch name>	เป็นคำสั่งสำหรับการสร้าง Branch ขึ้นมาใหม่ ซึ่งสร้างเพื่อนำไปใช้งานตามที่เราต้องการ เช่น ทำการแก้ไขส่วนต่าง ๆ โดยไม่ไปกระทบคนอื่น ๆ
git branch -d <branch name>	เป็นคำสั่งสำหรับลบ Branch ที่ต้องการ
git merge <branch name>	เป็นคำสั่งสำหรับการ Merge ไฟล์และโฟลเดอร์จาก <branch name> กับ Current branch หรือ Master (จากตัวอย่าง)
git remote add origin <url ของ GitHub>	เพิ่ม Remote repository เข้าสู่ GitHub server โดยใช้ชื่อของ Remote repository ว่า Origin
git push	ส่งไฟล์และโฟลเดอร์ที่ Commit เรียบร้อยไปที่ Local repository ไปยัง Remote repository
git push -u origin master	ส่ง Master ขึ้นไปเก็บบน Remote repository ที่ชื่อ Master และให้ Git จำพารามิเตอร์ไว้ด้วย เพื่อที่เราจะได้ไม่ต้องใส่ข้อมูลทุกครั้งที่ Push ข้อมูลขึ้น Git

คำสั่ง	การใช้งาน
git pull origin master	ดึงไฟล์และโฟลเดอร์ต่าง ๆ ที่มีการเปลี่ยนแปลงที่ Remote repository ลงมาที่เครื่องของผู้ใช้
git diff HEAD	ใช้ตรวจสอบ Repository ว่ามีการเปลี่ยนแปลงอะไรที่แตกต่างจากสิ่งที่เคย Commit ไว้แล้วบ้าง
git diff --staged	ใช้ตรวจสอบ Repository ว่ามีการเปลี่ยนแปลง Staged อะไรที่แตกต่างจากสิ่งที่เคย Commit ไว้แล้วบ้าง
git reset <file name>	สำหรับการยกเลิก Staged ของ file ที่เราต้องการออกไป ก่อนที่จะ commit (Unstaging นั่นเอง)
git checkout -- <file name>	เปลี่ยนไฟล์หรือโฟลเดอร์ที่สนใจกลับไปสู่สถานะที่มีการ Commit ครั้งสุดท้าย
git rm <file name>	สำหรับลบไฟล์และโฟลเดอร์ต่าง ๆ ออกไปจาก Git repository และจาก Storage อีกด้วย
touch <filename>	สร้างไฟล์ที่ Working directory แต่ยังไม่ได้ Add เข้าไปใน Repository

1.3 การสร้าง Repository ใน GitHub

การสร้าง Repository ไว้ที่ GitHub server ช่วยให้สมาชิกในทีมพัฒนาซอฟต์แวร์สามารถเข้าถึงงานเวอร์ชันล่าสุด และประสานงานกันได้ง่าย ขั้นตอนในการสร้าง Repository บน GitHub มีดังต่อไปนี้

ขั้นตอนที่ 1: เข้าไปที่เว็บไซต์ <http://github.com>

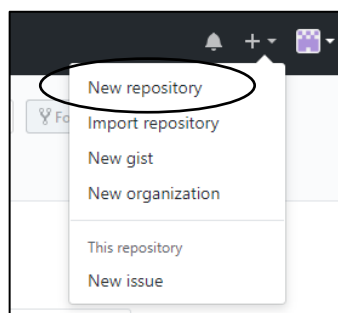
ขั้นตอนที่ 2: Log-in เข้าสู่ Account ของตนเอง

ขั้นตอนที่ 3: กดที่เครื่องหมาย + แล้วเลือก New Repository ดังแสดงในภาพที่ (1)

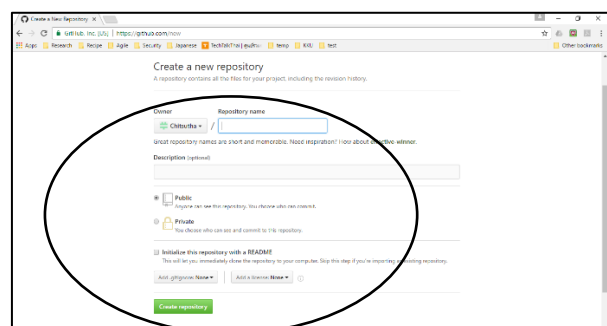
ขั้นตอนที่ 4: ตั้งชื่อ Repository และใส่คำอธิบายในช่อง Description เลือก Public หากต้องการสร้างไฟล์ README ให้เลือก Initialize this repository with a README ด้วย แล้วกดปุ่ม Create Repository ดังแสดงในภาพที่ (2)

ขั้นตอนที่ 5 เมื่อสร้าง Repository สำเร็จ จะปรากฏหน้าหลักของ Repository ที่สร้างขึ้น

(1)



(2)



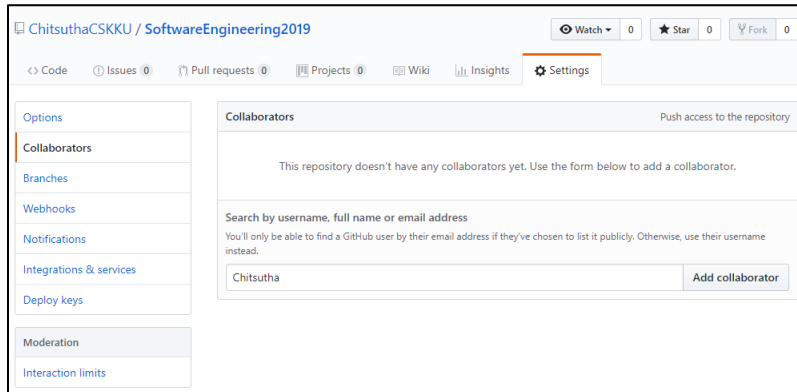
1.4 การเพิ่ม Collaborator ใน GitHub

การเพิ่ม Collaborator เข้ามาใน Repository ช่วยให้ทีมพัฒนาซอฟต์แวร์สามารถทำงานบนโปรเจกต์เดียวกันได้

ขั้นตอนที่ 1: ที่หน้าหลักของ Repository ที่ต้องการเพิ่มทีมงานเข้ามา ให้เลือกที่แท็บ Settings

ขั้นตอนที่ 2: เลือก Collaborator

ขั้นตอนที่ 3: ใส่ Username หรือ Email ของ Collaborator ที่ต้องการจะเพิ่มเข้ามาในโครงการ แล้วคลิกปุ่ม Add collaborator



ขั้นตอนที่ 4: ทีมงานที่ถูก Add เข้ามาจะต้องตอบรับคำเชิญ จึงจะสามารถเข้าถึง Repository ที่กำหนดได้

