

รายงาน  
Final Project



โดย

นายณนทพัทธ์ พงศ์พิศน์ 63130010051

นายธนุพัฒน์ วงหาญ 63130010316

เสนอ

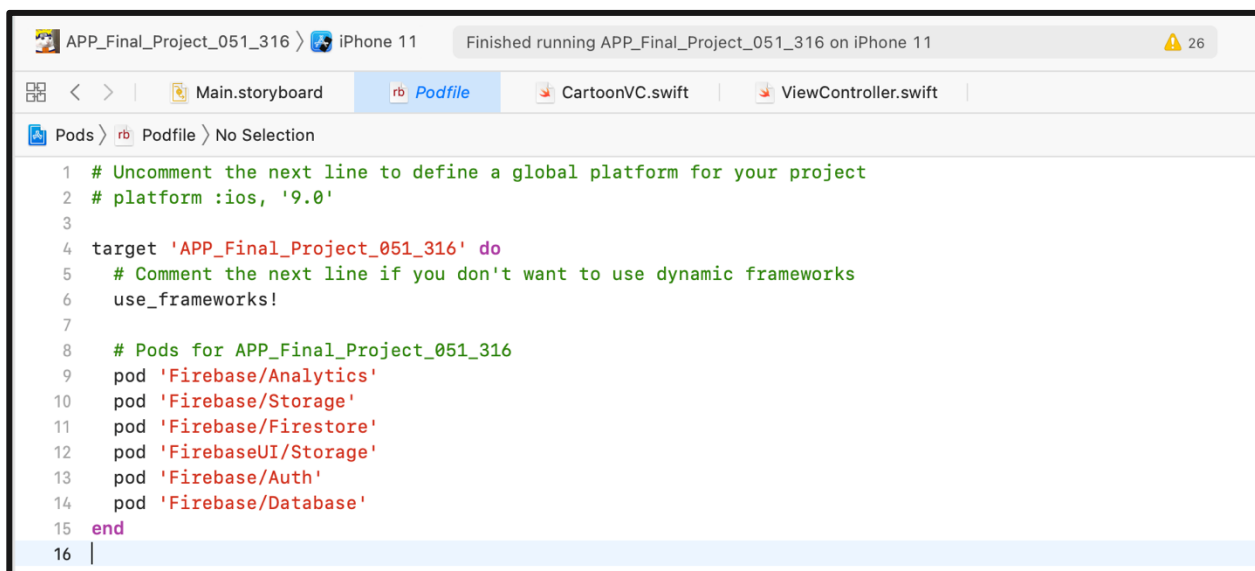
อาจารย์ สิทธิชัย วรโชติกำจร

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา CSC261 OBJECT ORIENTED PROGRAMMING  
ชั้นปีที่ 2 สาขาวิชานวัตกรรมการสื่อสารสังคม เอกคอมพิวเตอร์เพื่อการสื่อสาร  
ภาคเรียนที่ 1 ปีการศึกษา 2564  
มหาวิทยาลัยศรีนครินทรวิโรฒ

# Final Project

Key หลัก Innovation ใหม่ที่ได้ใช้ในโปรเจกต์

## นำเข้า Firebase



ทำการลง GoogleService-Info.plist ที่ได้ดาวน์โหลด จาก Firebase ลงใน Xcode หลังจากนั้นเราจะทำการ pod init ลงในโฟลเดอร์ของโปรเจกต์เราแล้ว ต่อมาใส่ pod ต่างๆที่จะใช้ในโปรเจกต์

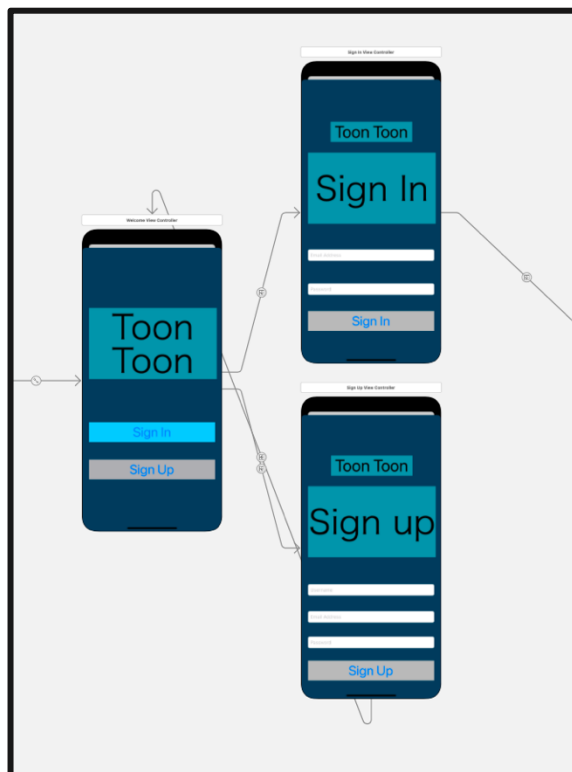
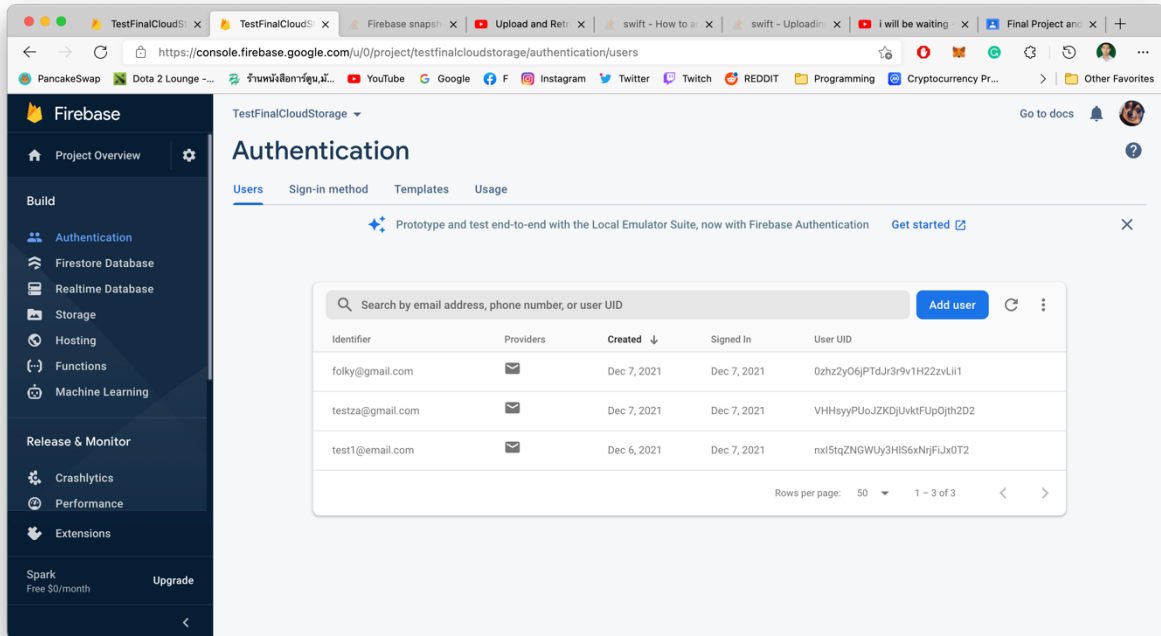
เช่น `pod 'Firebase/Auth'`, `pod 'Firebase/Storage'` แล้วทำการ `pod install` หลังจากนั้นก็ทำการ `import` Firebase และ `FirebaseApp.configure()` ลงไปใน AppDelegate

```
import UIKit
import Firebase

@main
class AppDelegate: UIResponder, UIApplicationDelegate {

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        FirebaseApp.configure()
        return true
    }
}
```

# ทำ Sign In, Sign Up, Log Out จาก Firebase Authentication



หน้าแรกคือหน้าที่มี ปุ่ม Sign In และ Sign Up จะมีฟังก์ชันที่ตรวจว่า CurrentUser ได้ Sign In ค้างไว้ไหม ถ้าตรวจได้ว่า Sign In ค้างไว้รอบหน้าที่เข้าแอปมา ก็จะไม่ต้อง Sign In ใหม่

```
override func viewDidLoad(_ animated: Bool) {
    super.viewDidLoad(animated)
    if Auth.auth().currentUser != nil {
        let vc=self.storyboard?.instantiateViewController(identifier: "alltoontabbar") as! UITabBarController
        self.view.window?.rootViewController=vc
    }
}
```

หน้าต่อมาเป็นหน้า Sign Up เมื่อเรากรอก Email, Password เรียบร้อยแล้วกดปุ่ม จะทำการ createUser แล้วนำไปเก็บใน Auth ของ Firebase

```
@IBAction func signUpBTN(_ sender: Any) {
    let auth = Auth.auth()

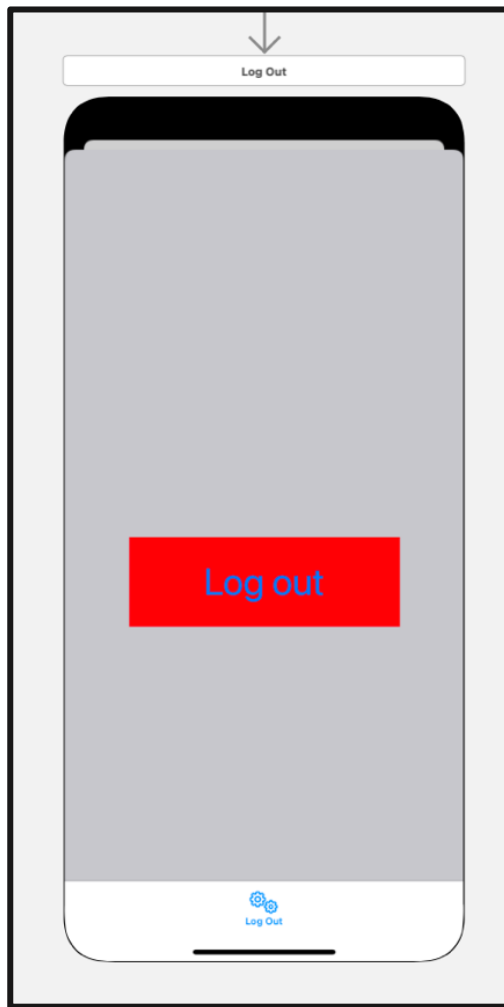
    auth.createUser(withEmail: emailFill.text!, password: passwordFill.text!) { AuthDataResult, Error in
        if Error != nil {
            self.present(Service.createAlertController(title: "Error", message: Error!.localizedDescription),
                animated: true, completion: nil)
            return
        }
        self.performSegue(withIdentifier: "backtologin", sender: nil)
    }
}
```

หน้าต่อมาเป็นหน้า Sign In เมื่อเรากรอก Email, Password เรียบร้อยแล้วกดปุ่ม จะทำการ signIn โดยใช้ข้อมูลจาก Sign Up ที่นำไปเก็บใน Auth ของ Firebase

```
@IBAction func SignInBTN(_ sender: Any) {
    let auth = Auth.auth()

    auth.signIn(withEmail: emailFill.text!, password: passwordFill.text!) { AuthDataResult, Error in
        if Error != nil {
            self.present(Service.createAlertController(title: "Error", message: Error!.localizedDescription),
                animated: true, completion: nil)
            return
        }
        self.performSegue(withIdentifier: "passloginsus", sender: nil)
    }
}
```

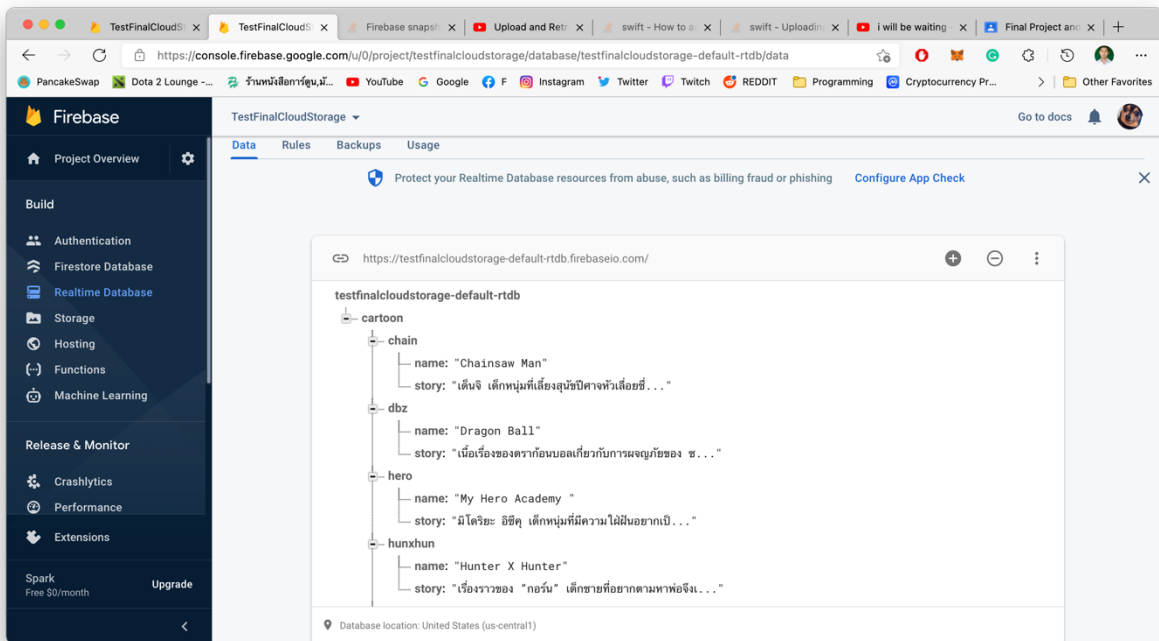
หน้าต่อมาเป็นหน้า Log Out เมื่อเรากดปุ่ม จะทำการ `signOut`



```
@IBAction func logOutBTN(_ sender: Any) {
    let auth=Auth.auth()

    do {
        try auth.signOut()
        self.dismiss(animated: true, completion: nil)
    } catch let SignOutError {
        self.present(Service.createAlertController(title: "Error", message:
            SignOutError.localizedDescription), animated: true, completion: nil )
    }
}
```

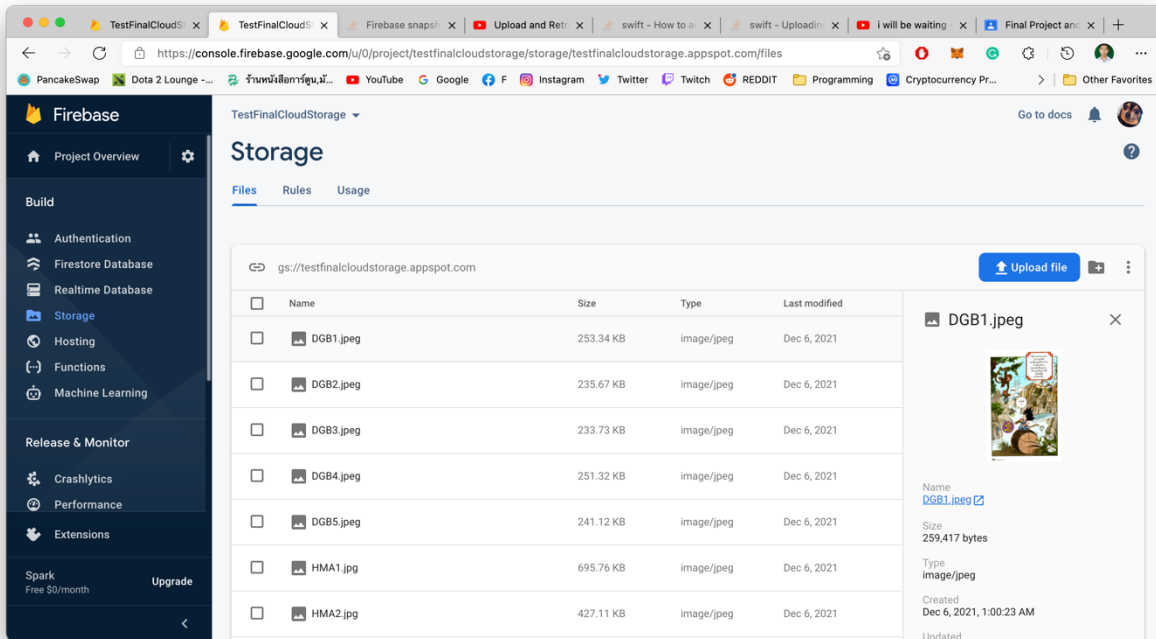
## การใช้ Realtime Database ของ Firebase



เราสร้างข้อมูลลงไปใน Database ของ Firebase หลังจากนั้นก็สร้าง `func getName()` เพื่อที่จะดึงข้อมูลจาก Database ออกมาเพื่อนำไปใส่ใน `UILabel`

```
func getName() {
    //แสดงชื่อและเรื่องย่อของ Toriko
    if bg == "torikobg.jpeg"{
        ref.child("cartoon").child("toriko").child("name").observeSingleEvent(of: .value, with: { snapshot in
            let name = snapshot.value as! String
            self.showname.text! = name
        })
        ref.child("cartoon").child("toriko").child("story").observeSingleEvent(of: .value, with: { snapshot in
            let story = snapshot.value as! String
            self.showStory.text! = story
        })
    }
    //แสดงชื่อและเรื่องย่อของ Chainsaw Man
    if bg == "chainbg.jpeg"{
        ref.child("cartoon").child("chain").child("name").observeSingleEvent(of: .value, with: { snapshot in
            let name = snapshot.value as! String
            self.showname.text! = name
        })
        ref.child("cartoon").child("chain").child("story").observeSingleEvent(of: .value, with: { snapshot in
            let story = snapshot.value as! String
            self.showStory.text! = story
        })
    }
}
```

## การใช้ Storage ของ Firebase



เรารูปภาพเข้าไปใน Storage ของ Firebase หลังจากนั้นก็สร้าง

```
let storage = Storage.storage()
let storageRef = storage.reference()
```

เพื่อนำมาดึงข้อมูลใส่เข้าไปใน UIImageView โดยใช้

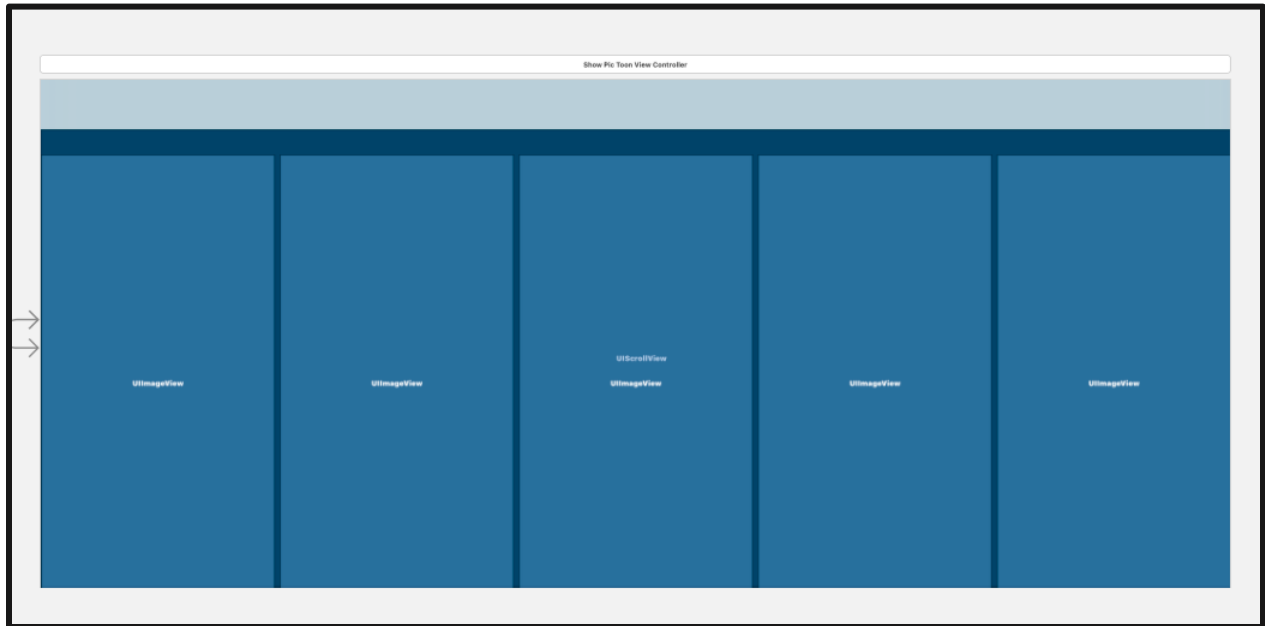
```
let ref1 = storageRef.child(pic[0])
```

ในการค้นหาลงใน Storage

```
override func viewDidLoad() {
    super.viewDidLoad()
    scrollView.contentSize = CGSize(width: 2100, height: 0)
    let storage = Storage.storage()
    let storageRef = storage.reference()

    let ref1 = storageRef.child(pic[0])
    let ref2 = storageRef.child(pic[1])
    let ref3 = storageRef.child(pic[2])
    let ref4 = storageRef.child(pic[3])
    let ref5 = storageRef.child(pic[4])
    Pic1.sd_setImage(with: ref1)
    Pic2.sd_setImage(with: ref2)
    Pic3.sd_setImage(with: ref3)
    Pic4.sd_setImage(with: ref4)
    Pic5.sd_setImage(with: ref5)
}
```

## การใช้ Scrolling View



มีการใช้ `UIScrollView` เพื่อเลื่อนรูปภาพการ์ตูนที่ดึงมาจาก `Storage` ของ `Firebase` โดยการใช้ `[String]()` เพื่อเช็คข้อมูลจาก `Storage` ถ้าข้อมูลตรงกันก็จะแสดงรูป และ ใช้ `scrollView.contentSize = CGSize(width: 2100, height: 0)` เพื่อปรับระดับการ Scrolling