

Configuration Manual

MSc Research Project
Cloud Computing

Sharannya Nair
Student ID: x21154520

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name:Sharannya Nair.....

Student ID:x21154520.....

Programme:Cloud Computing..... **Year:**2022.....

Module:MSc Research Project.....

Lecturer:Vikas Sahni.....

Submission Due Date:15/12/2022.....

Project Title: Data security using a hybrid cryptographic approach in mobile cloud computing

Word Count:1199..... **Page Count:**10.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:Sharannya Nair.....

Date: ...15 December 2022.....

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Data security using a hybrid cryptographic approach in mobile cloud computing

Sharannya Nair
Student ID: x21154520

1 Introduction

1.1 Motivation of the document

This document manual describes requirement, deployment instructions for Data security using hybrid cryptography approach in mobile cloud computing, according to National College of Ireland's project module handbook.

1.2 Configuration module structure

Section	Purpose
Basic Information	This section gives the information about basic configuration
Configuration required for development	Setup description required for implementation
Deployment for solution	This section describes the implementation procedure

2 Basic Information

The application is developed and tested using macOS version 11.6.4 and azure cloud services.

3 Configuration required for development

3.1 Pre-Requisite for the project implementation

- Install JDK (openjdk version "11.0.12" 2021-07-20)
- Install Visual Studio Community 2022 ([WebAssembly Web API](#) should be installed)
- Install Android Studio Dolphin 2021.3.1 (Android SDK- version 28) (Gradle version – 5.6.4 required)
- Azure cloud account
- Windows/Mac system

3.2 Code Repository

The Zip file of the code is attached

3.3 Running Web API

To run the web API, open `Cryptography_Api/Cryptography_Api.sln` in Visual Studio 2022, as shown in Fig 1. The Controller package contains the class “UserController”. This class contains the endpoint implementation of the API. The list of Web API endpoints is as given below:

1. Register API – endpoint used to register user
2. Login API – endpoint used to login user
3. Upload API – endpoint used to upload files to the cloud
4. Download API – endpoint used to download file from the cloud
5. GetFiles API – endpoint used to fetch file details of the user from the cloud

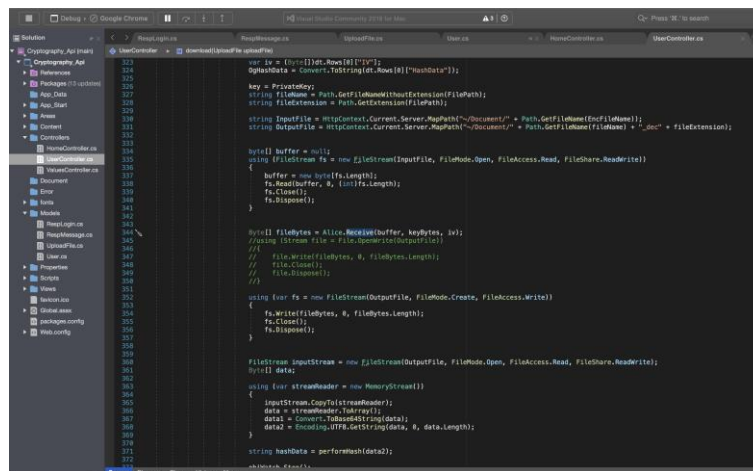


Fig 1: Web API workspace

3.4 Running mobile application

Steps to run cryptography mobile application

1. Open folder `CryptographyApp` in Android Studio IDE, as shown in Fig 2
2. Build and run application to generate an apk file
3. Apk file is provided in the code repository folder “`cryptography.apk`”
4. Use `PackageInstaller` in your mobile device and install app or can install from terminal of your system
5. To install apk from the terminal go to Android SDK folder, go to platform-tools folder. Open path in a terminal and type “`adb install cryptography.apk`”

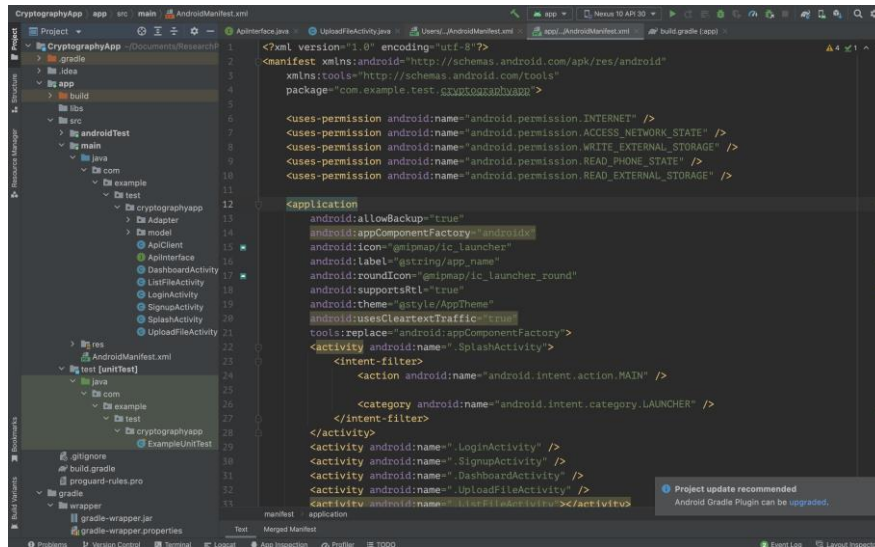


Fig 2: Android Studio workspace

Below are the screen flows of the CryptographyApp application

1. **Registration Screen:** Once the Cryptography is launched, the Splash screen will be displayed. From the splash screen, the user will be redirected to the registration screen if a user is not registered, as shown in Fig 3. User information like name, email address, password and and mobile number will be required to complete registration process.

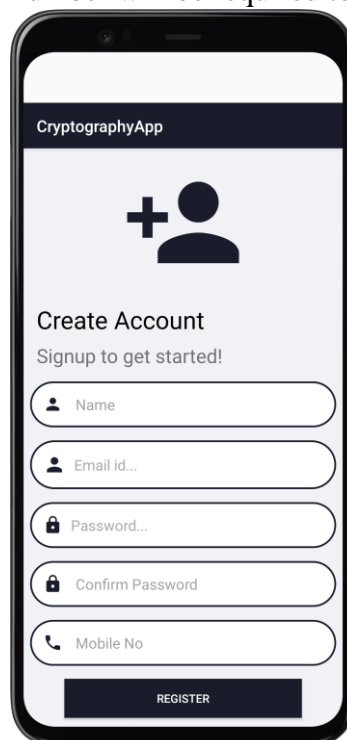


Fig 3: Registration Screen

2. **Login Screen:** If user is registered, the application will be navigated to login screen. The user needs to enter email id and password to login into the application, as shown in Fig. 3.

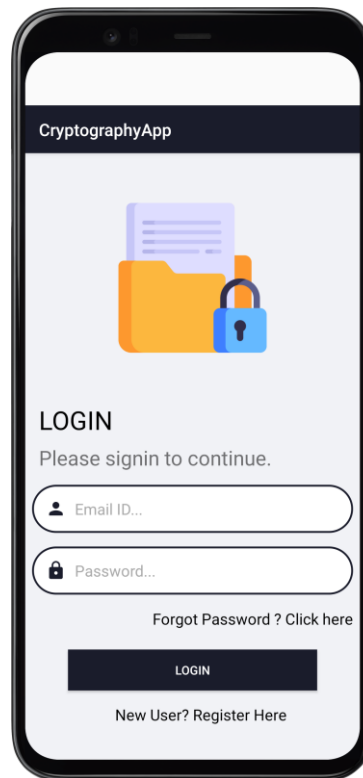


Fig 3: Login Screen

3. Dashboard Screen: Once the user is logged in, the application is redirected to the Dashboard Screen. This screen displays two options, one to upload files and another to view uploaded files.

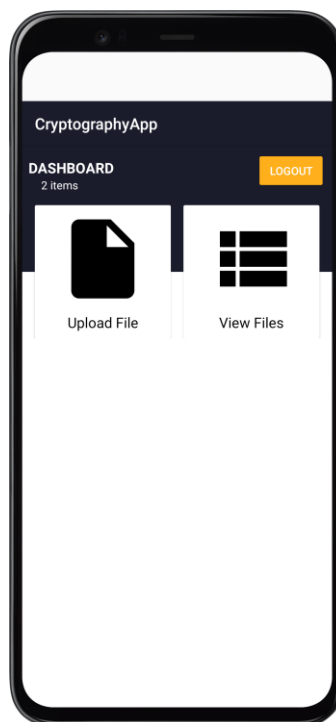


Fig 3: Dashboard Screen

4. Upload Screen: On clicking on the Upload file option, we can upload the file. The select button allows you to upload files from your mobile device. The file type can be text, pdf, word, jpg, etc. Once the upload of the file is successful, a key will be sent to the registered email id. This key can be used to decrypt file.

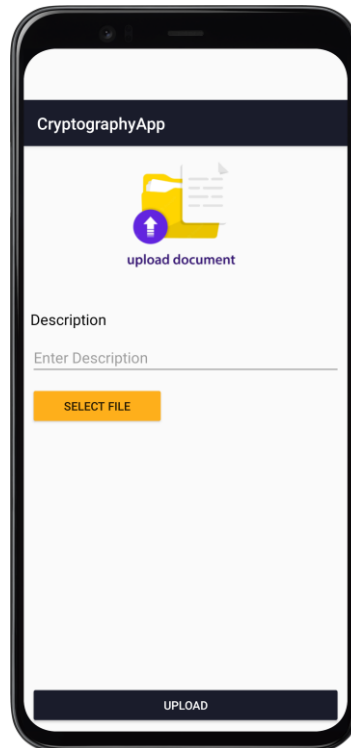


Fig 4: Upload Screen

5. View files Screen: This screen acts as an interface to view all files uploaded by the user in the cloud. Here, the user can use the key used in the email to decrypt and download the file from the cloud.

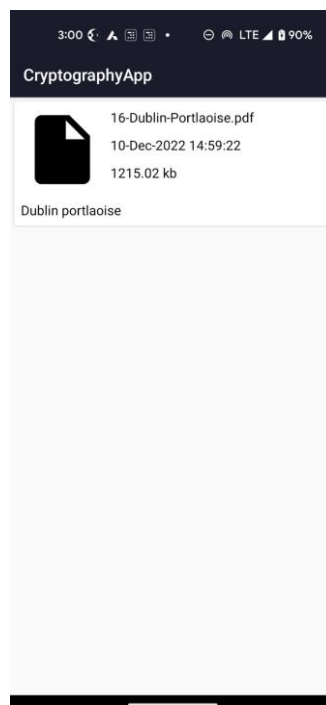


Fig 4: View files screen

6. Download files Screen: Once user selects the file, the download file pop up displayed as shown in Fig 5. Once the user enters the key, a pop-up screen is displayed as shown in Fig 6. The pop up displays the time taken of the file decryption and download option to download file and view files.

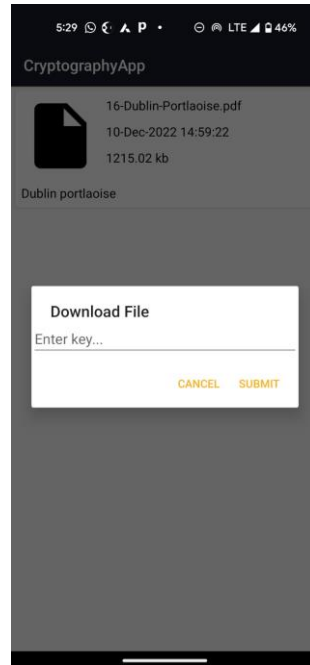


Fig 5: Download files pop up screen

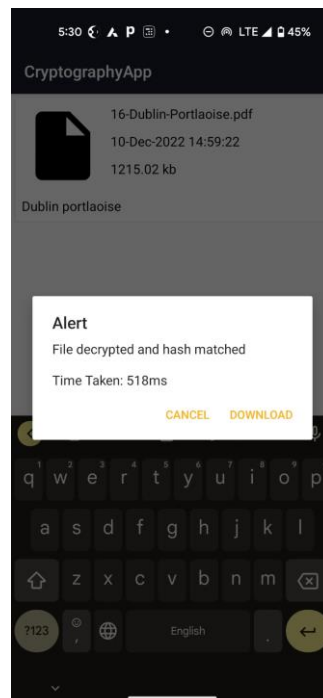


Fig 6: Download success screen

3.5 Programming languages

The programming languages required for application are as follows:

1. Java – Android framework (frontend development)
2. ASP.NET (middleware development)
3. SQL for database

4 Deployment of Services

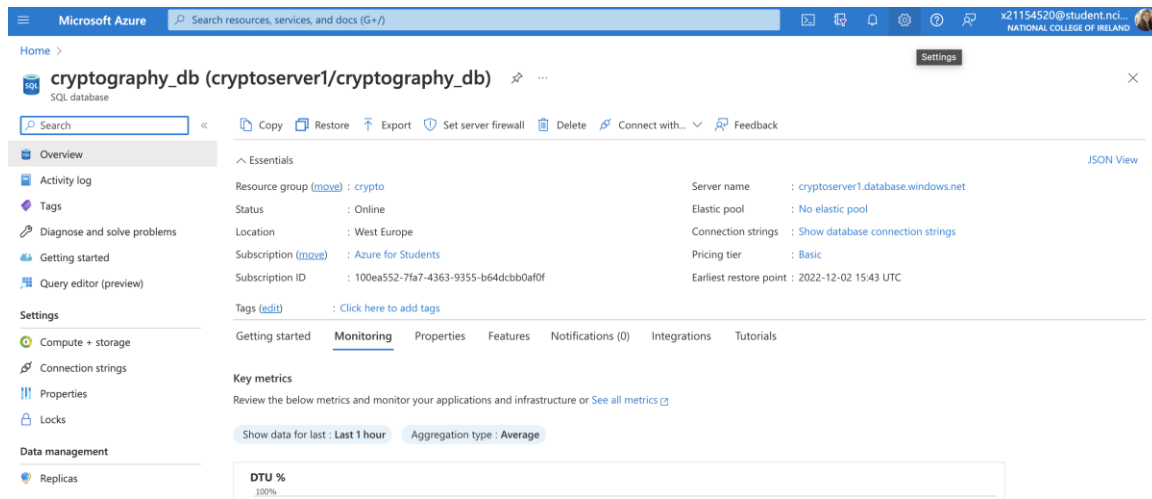
4.1 Creation of Azure App Service

Azure App service is used to host middleware API. Web service deployment is done here.
Steps to host web services:

- Go to Azure dashboard
- Go to Azure App service and create one service
- Select Runtime stack as ASP.NET 4.8
- After creating App Service, download “Get Publish profile”
- The publish profile is used in your code structure to publish to the web server(azure)
- Once the publish profile is copied in the workspace folder in Visual Studio. Right click on the project, select publish option. The App service layer will be published to the azure account. The publish settings contains the account details and app service details.
- Once published, copy the domain URL

4.2 Creation of Azure SQL Database

To create a database, go to SQL database and create an instance
Once database instance is created. Connection string is details in code base to connect to DB.



References

Beloglazov, A. and Buyya, R. (2015). Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds, *Concurrency and Computation: Practice and Experience* 27(5): 1310–1333.

Feng, G. and Buyya, R. (2016). Maximum revenue-oriented resource allocation in cloud, *IJGUC* 7(1): 12–21.

Gomes, D. G., Calheiros, R. N. and Tolosana-Calasan, R. (2015). Introduction to the special issue on cloud computing: Recent developments and challenging issues, *Computers & Electrical Engineering* 42: 31–32.

Kune, R., Konugurthi, P., Agarwal, A., Rao, C. R. and Buyya, R. (2016). The anatomy of big data computing, *Softw., Pract. Exper.* 46(1): 79–105.