**D424 – Software Engineering**

**Task 4**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  University Course Tracker |
| **Student Name:** | Lance Roller |

**Table of Contents**

Part A:  
1.1. Cloud Solution Justification …………………………………………. Page 3  
1.2. Container Image Implementation ……………………………………Page 4

Part B:  
2.1. GitLab Implementation…………………………………………….… Page 4

Part C  
3.1. Panopto Video………………………………………………….……… Page 5

**Deployment Documentation**

**PART A1:**

**Cloud Solution Justification**

For the deployment of my .NET MAUI mobile application, I selected Google Play Store (Internal Testing) as the cloud solution provider. Later the application can go on a public track rather than an internal track if needed. Google Play was chosen because it is the official distribution platform for Android applications and offers a streamlined way to distribute, update, and manage mobile applications. The Internal Testing track was particularly useful as it allowed for controlled deployment, enabling selected testers to validate the app before a full public release.

One of the main advantages of using Google Play is its scalability, as it efficiently manages app distribution and updates without requiring additional infrastructure. Additionally, the platform ensures security by scanning applications for malware before they are made available to users. Another benefit is the testing flexibility, where different testing tracks, such as internal, closed, and open testing, allow for iterative improvements before full deployment. By using the Internal Testing track, selected testers could install and use the application while automatically receiving updates for new versions. Given that .NET MAUI is designed as a cross-platform framework, Google Play was the most suitable option for deploying the Android version of the application. For iOS deployment, a similar process would be required through Apple’s App Store.

**PART A2:**

**Container Image Implementation**

Regarding containerization, technologies such as Docker were not implemented in this project. The primary reason is that .NET MAUI applications are natively compiled mobile applications, meaning they do not require containerized environments like web-based applications do. Container images are primarily useful for backend services or cloud-based applications that need consistent execution across multiple platforms. Since the Google Play Store handles deployment directly, there was no need for a containerized approach. Instead, the application was built as an Android App Bundle (AAB) and uploaded directly to Google Play Console.

**PART B:**

**GitLab Implementation**

The project was developed and maintained in GitLab, with version control and commits managed throughout. It was compiled in Publish mode using Visual Studio, generating an Android App Bundle (AAB) for deployment via Google Play Console. After uploading the AAB to the Internal Testing section and configuring settings, testers were invited via email to install and test the app, receiving automatic updates for new versions. The GitLab repository was exported as a compressed file and submitted alongside the repository branch history and deployment URL.

***The GitLab Link can be found*** [***HERE***](https://gitlab.com/wgu-gitlab-environment/student-repos/lroll47/d424-software-engineering-capstone/-/tree/Working?ref_type=heads)***.***

**PART C:**

**Panopto Video**

A Panopto video was recorded to demonstrate the deployment process, covering AAB uploads, release management, tester invitations, and app installation. This deployment ensures seamless distribution to internal testers for validation before a potential public release.

***The Panopto Link can be found*** [***HERE***](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=563553ed-6779-4ed9-b9bc-b2a2013edcfc)