# Section 1

Authentication

Database

# Section 2

Authentication

Google Authentication –

<https://www.nuget.org/packages/Microsoft.AspNetCore.Authentication.Google>

Requires no interface

Provides Interface on Google Authentication Apps

<https://www.nuget.org/packages/Google.Apis.Oauth2.v2/>

Requires no interface

Provides Interface on Google APIs Client Library

<https://www.nuget.org/packages/GoogleAuthenticator>

Requires QRCoder Interface and secret key management

Provides QR code creator and verification interfaces

Email Sending

<https://www.nuget.org/packages/AWSSDK.SimpleEmail>

Requires no interface

Provides Sending Email Interface

Database –

Google Cloud Database

Requires no interface

Provides interface to interact with google services

SQL Server

Requires no interface

Provides interface to perform SQL commands

Azure Database

Requires no interface

Provides interface to interact with azure services

# Section 3

Authentication

Best component for this project is GoogleAuthenticator (<https://www.nuget.org/packages/GoogleAuthenticator>) as it contains commands only for authentication and no other google authentication measures as well as acting as a wrapper for the google authentication we needed for this project. This also links with our other components well because it does not need any extra setup to work, just managing the secret key in the database and generates a QR code for us to use. It is maintained and has a large user base. It is open source so the codebase can be customised and kept up to date.

Database

Best component for this project is using an SQL server, this was because we could customise everything to suit our needs and do not have to rely on google and azure. This would require the maintenance be on our end but given the nature of the data that is being stored a bank would like to have data on site. This also allows us to ensure that this will work with all of our components that we are using. This also has a nice integration into the IDE that we are using (Visual Studio)

# Section 4

The components diagram needed to be changed as the use of the SQL server removes the server component as well as having to include QR coder as a component that is accessed by the authentication service