COS 301 HighTech Team - SRS Document

Theoveshan Naidu
16148861

Janaki Patil
16006110

Zi Xin Zhang
15192556

Alexandros Petrou
15291792

Tristan Sander-hughes
17071390

2 May 2019

Contents

1 Introduction **3**
1.1 Definitions,Acronyms . . . 3

2 User Characteristics **3**
2.1 Intended users . . . 3

3 Technology and Constraint **4**
3.1 Technology Decisions . . . 4
3.1.1 Ionic Framework . . . 4
3.1.2 Typescript . . . 4
3.1.3 Github . . . 4
3.2 Constraints . . . 4
3.2.1 Android API levels . . . 4
3.2.2 Open Source / Free . . . 4

4 Function Requirements **5**
4.1 Use-case . . . 5
4.2 Functional requirements . . . 6

5 Traceability Matrix **7**

6 Quality Requirements **7**

7 Diagrams **9**
7.1 UML use case diagrams . . . 9
7.2 Domain Model . . . 10
7.3 Deployment Diagram . . . 11

8 Architectural Design **11**

9 Architectural Design Diagrams **13**
9.1 Architectural Design Diagram for admin portal . . . 13
9.2 Architectural Design Diagram for mobile application . . . 14

10 Conclusion **14**

11 References **15**

1 Introduction

In the modern plumbing industry, a continuous source of inconvenience arises for plumbers to log information on the geysers and components installed on various jobs by means of written material on a day-to-day basis. Considering this the goal of this project is to design a new database application system that greatly reduces the inconvenience of having to log information about installed geysers by taking advantage of the prevalence of mobile devices in the modern world. In order to do this, we will shift the logging of details from the paper-based method itself to a mobile application, detailed below.

1.1 Definitions,Acronyms

This section provides definitions of all items, acronyms and abbreviations to interpret the SRS document properly.

1. Certificate of Compliance - A certificate to prove that the geyser installation complies with regulation standards.
2. App - Application

2 User Characteristics

2.1 Intended users

The are different kinds of users that will be interacting with our system. The intended users of the software are:

Plumber- Geyser Installers

The most interactive part of the app consists of what the installer will be doing. The installer would be a user that is installing the geyser and associated parts, they require a fast and easy way to compile all related information files and images onto a database in order to streamline future adjustments and insurance claims.

Admin

The admin will be the one who will be adding a new case (request to install or repair a geyser), closing the cases, adding and deleting new employee. They are the core of this system since they perform most of the data processing.

Homeowners

The homeowners are the one who will be giving feedback on the service that has been provided to them(such as fixing or installation of geyser).

3 Technology and Constraint

3.1 Technology Decisions

3.1.1 Ionic Framework

We chose Ionic Framework because is a free open source mobile development framework that allows us to create cross-platform applications that are native to the platform. Ionic allows us to create applications with Javascript/Typescript, HTML and CSS. We wanted that functionality because it would allow us to not only port our application into a web-app, but also a desktop environment using Electron Framework all using a single codebase.

3.1.2 Typescript

Typescript is a typed superset of Javascript that compiles into plain Javascript. We choose this language because it integrates well with Ionic, and it allows us to write typed code which helps us catch errors at compile time, and also helps us debug our application much easier than a dynamic language. Typescript is also free and open source with a large community that would help us get libraries and help that we might need.

3.1.3 Github

We chose to use Github because it is a free centralized area what we can all collaborate using the Git Work-Flow that allows us all to work on the same project effectively and with as less conflict as possible. It has many different tools that allows us to review each others work, and also help.

3.2 Constraints

3.2.1 Android API levels

We are limited to only working with Android devices and because android devices each all have different API levels that allow for different features and functionality, we had to make sure that our application was limited to only using the API level of the lowest common android.

3.2.2 Open Source / Free

We are limited in only using open source or free software and framework to build our application, propriety software would be too expensive, or a limitation should this application wish to do something that is against their software policies.

4 Function Requirements

4.1 Use-case

UC1 The plumber login with its credentials.
UC2 The plumber scans the barcode that is placed on the geyser.
UC3 The plumber uploads hot water system images.
UC4 The plumber updates the details if repairs are done on the geyser.
UC5 The admin login with his/her credentials.
UC6 The admin gets signedup if they are a new employee.
UC7 The admin enters caller's details .
UC8 The admin enters case details.
UC9 The admin closes the cases that are completed.
UC10 The admin can view statistics on number of cases handled per month.
UC11 The admin can view statistics of the geysers that are mostly preferred or installed.
UC12 The admin adds new employee .
UC13 The admin deletes an old employee.
UC14 The admin can search for a specific plumber and view the cases that he has worked on.
UC15 The admin can search for a specific geyser details.
UC16 The admin can search for a specific case number.
UC17 The plumber can view his/her "to-do" list.
UC18 The plumber updates the case status from pending to complete upon completing his/her work at a site.
UC19 The admin then reviews all completed cases and changes the case status from complete to closed.
UC20 The user gives feedback on the service provided to them.

4.2 Functional requirements

Authentication subsystem

1 The app shall provide authentication functionality.

1.1 The app should allow the users to scan barcode on the geyser in order to proceed.

1.2 The app and website should differentiate between the different types of users (admin and plumber).

Data Capturing subsystem

2 The app and the website shall allow various details to be captured.

2.1 The website must capture employee's detail.

2.2 The app must allow the plumber to enter the Geyser Details.

2.3 The website must capture the callers details.

2.4 The website must capture the case details.

2.5 The app must allow the plumber to upload installation related images.

2.5.1 The app must allow the plumber to upload Hot water system images.

Report subsystem

3 The website shall generate a report based on different criteria.

3.1 The website should generate a report based on the details captured for the administrator.

3.2 The website should generate a report based on the number of cases handled per month.

3.2 The website should generate a report on potential risk items.

5 Traceability Matrix

	Authentication	Data Capture	Report
Functional			
1.1	X	X	
1.2	X		
2.1		X	
2.2		X	
2.3		X	
2.3.1		X	
2.3.2		X	
2.3.3		X	
2.4		X	
2.5		X	
3.1			X
3.2			X
3.3			X
Quality			
QR1		X	X
QR2		X	
QR3	X		
QR4	X	X	X
QR5		X	X
QR6	X	X	
QR7			X
QR8		X	

6 Quality Requirements

QR1: Performance
(a) - The app shall check how many geysers are installed a day. This allows the app to update the heat map with additions of new geysers.
(b) -The app should upload the information to the database within 20 seconds of being connected to the internet.
QR2: Reliability
(a) -The information will be stored locally on the phone until it is uploaded to the database on the server where it will have the ability to be accessed.
(b) -The database shall keep all information of every geyser(CRUD), in case of a server failure, all the information should also be duplicated to another storage area for redundancy.
QR3:Security
(a) -Since information stored on the database needs to be confidential and comply with the popi act the database needs to be secured by password access to ensure security.

(b) -Incorrect credentials entered need to alert the admin(by sending notification) that someone is trying to access the application without permission.
(c) -Access permissions for particular system information shall only be changed by the system administrator.

QR4: Scalability

(a) -The app server must be able to scale the amount of computing power it uses depending on the amount of devices currently connected, so it must be able to make more processing power available if the number of connected users rise above a certain.

QR5: Availability

(a) -The app shall allow users the enter data from any location and upload whenever they have internet signal
(b) -The app shall show users the locations of geysers with the corresponding filter options on a heatmap to show certain geysers of various attributes for example manufacturer of the geyser, model,size etc.
(c) -The system shall be available for 99 percent of the time.

QR6: Maintainability

(a)- The database will allow an admin user to maintain all information on the database incase changes need to made.
(b)- The app will be created in a modular fashion to allow changes to be made to modules without affecting the whole app.

QR7: Monitorability

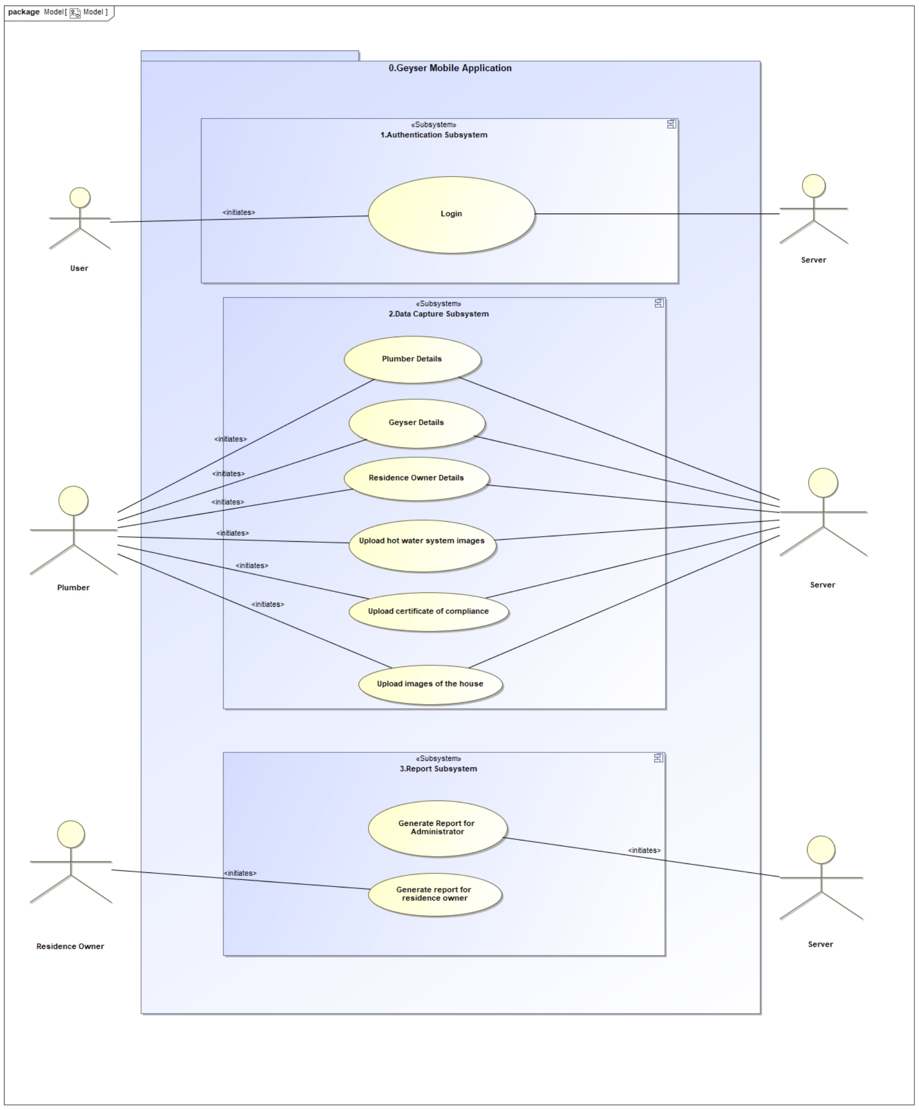
(a) -The app will make use of multiple heat map views to display different details of geysers in the area.

QR8: Usability

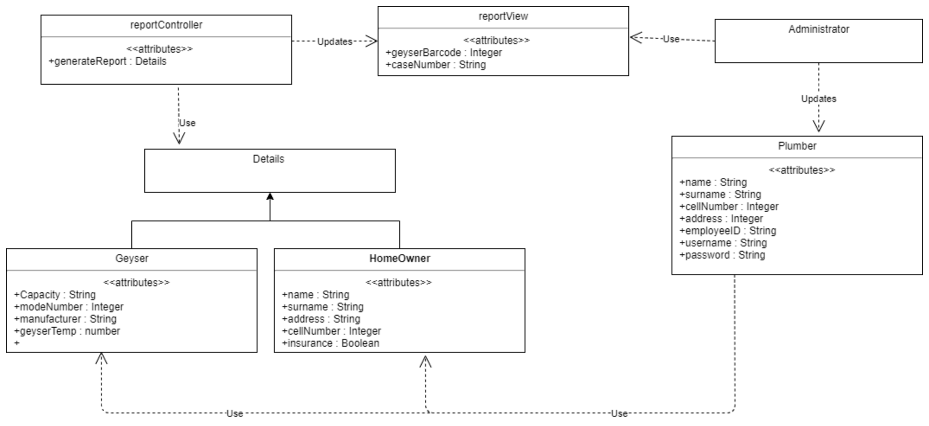
(a) -App users who want to enter information or edit information shall be able quickly and easily to to accomplish any task. -the app shall process actions in less than 30 seconds

7 Diagrams

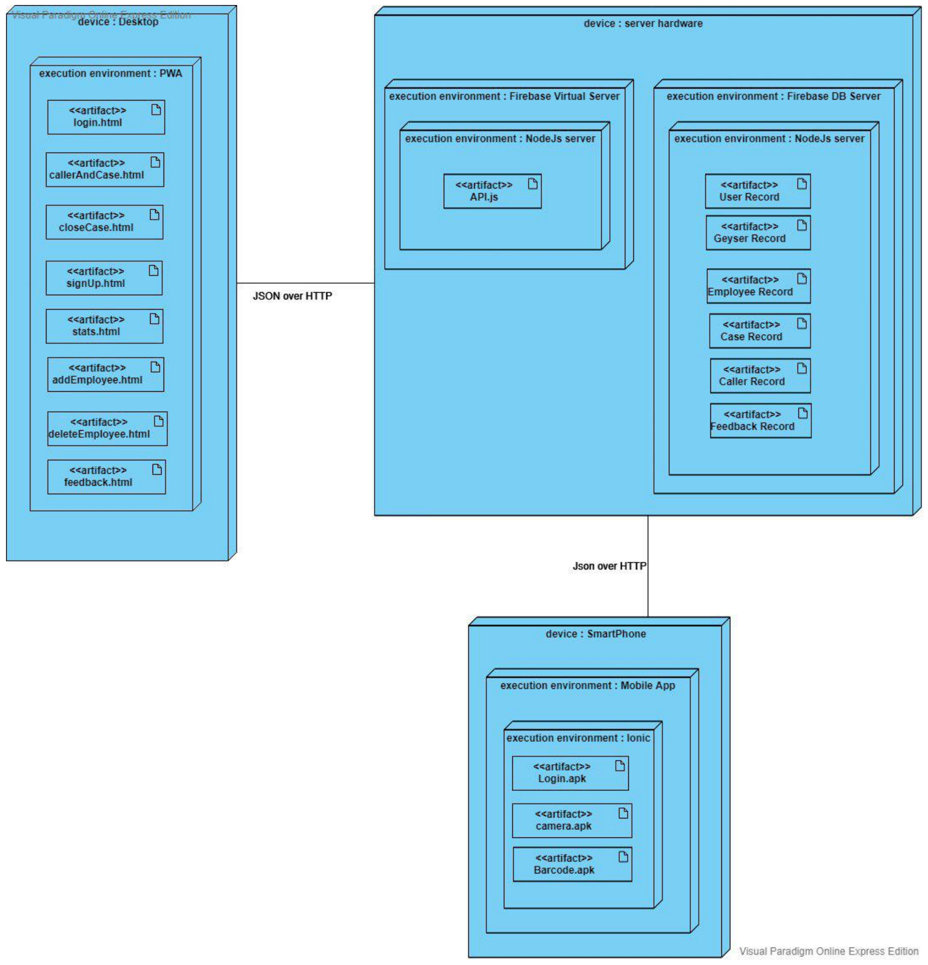
7.1 UML use case diagrams



7.2 Domain Model



7.3 Deployment Diagram


Visual Paradigm Online Express Edition
device : Desktop
execution environment : PWA
<<artifact>> login.html
<<artifact>> callerAndCase.html
<<artifact>> closeCase.html
<<artifact>> signUp.html
<<artifact>> stats.html
<<artifact>> addEmployee.html
<<artifact>> deleteEmployee.html
<<artifact>> feedback.html
JSON over HTTP
device : server hardware
execution environment : Firebase Virtual Server
execution environment : NodeJs server
<<artifact>> API.js
execution environment : Firebase DB Server
execution environment : NodeJs server
<<artifact>> User Record
<<artifact>> Geyser Record
<<artifact>> Employee Record
<<artifact>> Case Record
<<artifact>> Caller Record
<<artifact>> Feedback Record
Json over HTTP
device : SmartPhone
execution environment : Mobile App
execution environment : Ionic
<<artifact>> Login.apk
<<artifact>> camera.apk
<<artifact>> Barcode.apk
Visual Paradigm Online Express Edition

8 Architectural Design

<u>n-tier</u>

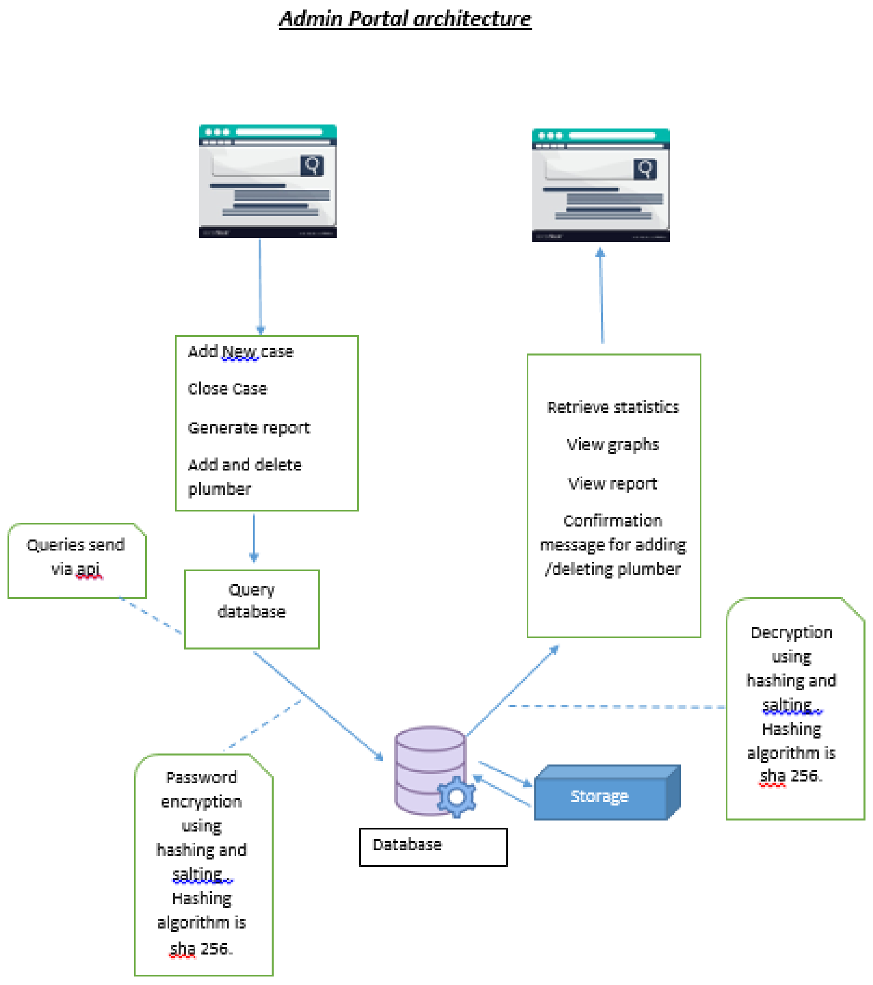
admin portal side

On the portal side we start on the presentation tier with the login form with the option to move to the signup form if you are a new user, followed by retrieving the necessary information in order to generate a new case. From there the logical layer comes into play where we receive the corresponding details from the user or caller and add it to the data tiers part which is the database. Another aspect of the presentation tier is the interface for display reports and statistical graphs as well as the interface for closing cases marked as complete. In order to do this we need to first retrieve the information accordingly and

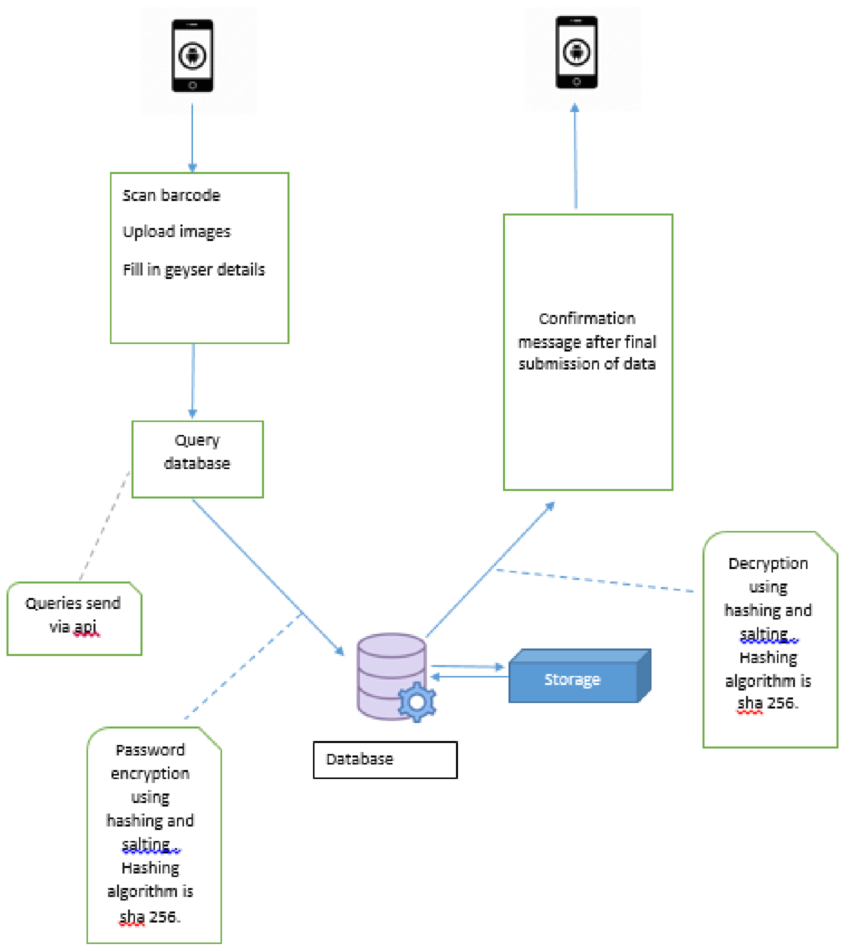
display them in the correct form using logic tier. The final section of the presentation tier is the interface for adding and removing employees which uses the logic tier to navigate to the corresponding tables and either remove or populate the tables as needed according to the data tier. Mobile Application side
Similarly, to the portal side the mobile application also follows an n tier architecture with the presentation tier using a login interface, a barcode scan interface and a interface to take images and store documents which is the main purpose of the mobile application. When we move to the next tier which is the logic tier we deal with getting the credentials for the login, using the plumbers ID to know what job they have to complete, obtaining a value barcode from the scan and obtaining the image to use in the next tier. The next tier being the data tier, this allows us to match the correct credentials with what was used to login in with that on the database and then know how to proceed, allows us to retrieve the case information for the plumber to complete, this is done based on the plumbers ID. For the barcode scan it just stores the value in to correct table to assign later to the specific geyser of the specific case. For the images it allows us to store the actual image taken or a compressed version in the database in order to save space. This is also saved in its relevant folder under the image folder and the path which is the caseID to this folder is then stored in the corresponding table of geyser information to make searching easier.

9 Architectural Design Diagrams

9.1 Architectural Design Diagram for admin portal


Admin Portal architecture
Add New case
Close Case
Generate report
Add and delete plumber
Queries send via api
Query database
Retrieve statistics
View graphs
View report
Confirmation message for adding /deleting plumber
Decryption using hashing and salting. Hashing algorithm is sha 256.
Password encryption using hashing and salting. Hashing algorithm is sha 256.
Storage
Database

9.2 Architectural Design Diagram for mobile application


Scan barcode
Upload images
Fill in geyser details
Query database
Queries send via api
Password encryption using hashing and salting. Hashing algorithm is sha 256.
Database
Storage
Confirmation message after final submission of data
Decryption using hashing and salting. Hashing algorithm is sha 256.

10 Conclusion

The new way of loading geyser installation details need to comply with the current direction in which reporting is done all over the country - into the mobile, portable, personal sphere. With the system specified in this document, the same old logging functionality is shifted into this new sphere - as well as gaining some improvements.
In conclusion the proposed system will change the way plumbers interact with the logging of information. More specifically it will focus on evolving the way plumbers and homeowners submit their information by moving towards a more mobile approach allowing users to upload images and type out legible data. Interfaces will be created on Geyser24(mobile app) and will speed

up the process of uploading of info and allowing a user to simply enter their geyser number and display the generated report with all major details included.
In turn the process times will greatly be reduced as well as increasing security by encrypting important data. Geyser24 will also allow users to log queries about their geyser to receive help faster. Geyser24 will change the plumbing industry as well as the household as it will provide the fastest and easiest way to log issues and keep track of geyser information.

11 References

Adams, D. (1995). *The Hitchhiker's Guide to the Galaxy.*

Groenfeldt, T. (n.d.). *Some Banks Are Heading To The Cloud -- More Are Planning T.* Retrieved from Forbes: https://www.forbes.com/sites/tomgroenfeldt/2014/06/26/some-banks-are-heading-to-the-cloud-more-are-planning-to/#75e6fa7158c9

Schwartz, B. (n.d.). *Do the Math: Can Your Business Benefit From a Cloud Database?* Retrieved from Tech: https://tech.co/news/math-can-business-benefit-cloud-database-2015-08

Zamfiroiu, A. (2012). *Integrability and Interoperability of Mobile.* Retrieved from Zamfiroiu: https://www.researchgate.net/publication/236648829_Integrability_and_Interoperability_of_Mobile_Applications