CEN 4010 - Principles of Software Engineering

Term: Summer 2023

Milestone 4: Beta Launch and Final Project Reviews

Application Title: Bizz QR

Group:15

**Team members:**

| Name | **Role** | **Email** |
| --- | --- | --- |
| Jacob Kahn | **Team Leader, Product Owner, Front End Developer** | **jacobkahn1996@gmail.com** |
| Keanu Francis | **Back End Developer , Scrum Master, Github Master** | **Kfrancis2018@fau.edu** |
| Sunny Chen | **Back End Developer** | **schen2014@fau.edu** |
| Temel Durak | **Front End Developer** | **tdurak2022@fau.edu** |

Date:7/19/2023

Revision History

Github Repo:

<https://github.com/Kfrancis2018/Group-15-Milestones.git>

**Table of Contents**

Product Summary-................................................................................................................................... #3

Usabilitity Test Plan -.............................................................................................................................. #4

QA Test Plan-…………………………….……………………………………………………………..#6

Code Review-.……………………………...…………………………………………...…………..…. #7

Self Check on Best Practice Security-.………….……………...…..………………….…………..…. #13

List of Non-Functional Specifications-.…………………………………………….…………………#14

**Product Summary**

**BIZZ QR - https://kfrancis2018.github.io/Group-15-Project-BIZZ-QR/**

1. vCard – Users can create a vCard that will be saved to their dashboard. The vCard will be encoded into a QR code that, once scanned, will add the saved information to the scanner's phone as a contact.
   1. Saving: The first name, last name, organization, title, phone number, address, email, website address, and optional block of text "About You" will be saved in the database.
2. Location – Users can create a QR code that, when scanned, will use Google Maps to locate the information encoded in the QR code.
   1. Saving: The street, city, state, and postcode will be saved to the database.
3. Call – Users can create a QR code with a phone number encoded that, when scanned, will immediately call the number.
   1. Saving: The phone number will be saved to the database.
4. Message - Users can create a QR code that, when scanned, opens the scanner's messaging app, and creates an SMS with a prefilled message.
   1. Saving: The phone number and message to be saved are stored in the database.
5. Text – Users can create a QR code that contains a block of text that, when scanned, can be viewed by the scanner.
   1. Saving: The block of text will be saved to the database.
6. Website - Users can create a QR code that, when scanned, navigates the default browser to the specified URL.
   1. Saving: The URL of the website will be saved in the database.
7. Edit: Then data saved in the database can be edited , changing all or some of the information saved in that object.
8. Viewing: Users can view the information saved encoded into a qr code. This qr code can then be scanned or downloaded to be shared if the user desires. If the option is a vCard then the user has the option to download the .vcf file.
9. Sign up and Login: users can login and signup to create and save the QR codes of their choice.

All saved QR codes can be viewed in the users dashboard options such as edit delete and view for each qr code will be available from the dashboard. Unlike most of these services BIZZ Qr will offer a limitless number of qr code to be saved to the database. Since the Qr code are generated locally the space needed to store the information is minimal.

**Usability test plan**

**1. Test Objectives**

The objective is to evaluate the ease of use, efficiency, and intuitiveness of the QR Code Generation feature in Bizz QR. The aim is to understand how easily users can input their details and generate a unique QR code, and whether the process is user-friendly. We want to ensure that all of the different QR types are functioning properly after the generation step and that it is compatible with all smartphones that have QR scanning capabilities. We also aim to identify any potential areas of confusion or difficulty that could be improved to enhance the user experience.

**2. Test Plan**

* System Setup: A standard PC with an internet connection, and the Bizz QR application installed.
* Starting Point: The user is logged into their Bizz QR account and is on the home page.
* Task to be Accomplished: The user needs to input their details and generate a unique QR code.
* Intended User: The intended user is a small business owner or student who wants to use Bizz QR for networking purposes.
* Completion Criteria: Test is considered complete when the user has successfully generated a QR code and is able to view or download it.
* URL of the System to be Tested:https://kfrancis2018.github.io/Group-15-Project-BIZZ-QR/pickQRtype.html

**3. Questionnaire Form**

| * **Questionnaire Form** | Strongly Disagree | Disagree | Neither agree or disagree | Agree | Strongly Agree |
| --- | --- | --- | --- | --- | --- |
| * The process of inputting my details for QR code generation was straightforward and easy to understand. |  |  |  |  |  |
| * I was able to generate a QR code quickly and without any difficulties. |  |  |  |  |  |
| * I found the QR code generation feature to be intuitive and user-friendly. |  |  |  |  |  |
| * I was able to find and use a QR code that I had previously generated. |  |  |  |  |  |
| * I was able to share my business info easily using the Vcard. |  |  |  |  |  |
| * My phone was able to automatically create a contact using the Vcard QR. |  |  |  |  |  |
| * My phone was able to automatically input the address into Google Maps using the Location QR. |  |  |  |  |  |
| * My phone was able to automatically call the number stored in the Call QR. |  |  |  |  |  |
| * I was able to visit the website using the Website QR with no issues. |  |  |  |  |  |
| * This App works the same on on my different devices |  |  |  |  |  |
| * The Navigate the Dashboard efficiently |  |  |  |  |  |

This usability test plan is designed to be easy to read and use by usability testers, and the questionnaire is formatted for easy use by reviewers.

**QA test plan**

1. Test Objectives

* The main objective of this QA test plan is to ensure that Bizz QR, a QR code software designed for networking, provides a seamless user experience for creating QR codes.
* The test will focus on the software's ability to generate unique QR codes representing user details, the ease of sharing these details, and the compatibility of the QR codes with different browsers and hardware.

2. Hardware and Software Setup

* The tests will be conducted on a standard PC/Phones with an internet connection.
* The software setup includes Bizz QR application, two different types of web browsers for testing browser compatibility (Google Chrome and Microsoft Edge/Safari).
* a QR code reader installed on a smartphone Iphone and Android for testing the QR code functionality.

3.Tested Feature

* Features to be tested are the QR code generation and sharing functionality. This includes the ability to input user details, generate a unique QR code representing these details, and the ease of sharing and scanning these QR codes.

Test Case 1: QR Code Generation For Each Mode

* Test : Verify that the application generates and saves QR code when user details are inputted.
* Input: User details (Name, Contact Information, Depending on mode)
* Output: A unique QR code representing the user details.
* Test Results: PASS

Test Case 2: QR Code Sharing

* Test: Verify that the generated QR code can be easily shared.
* Input: Generated QR code
* Output: A shareable link or downloadable image of the QR code.
* Test Results: PASS

Test Case 3: QR Code Scanning

* Test: Verify that the QR code can be scanned using a QR code reader and the correct user details are displayed.
* Input: Shared QR code
* Output: User details displayed on the QR code reader.
* Results: PASS

**Code Review- vCard**

**Coding style:**

This project uses a combination of javascript to get and save the qr codes into the database. Since it is required that a user can potentially save large amounts of qr codes in their database it is imperative that this data is as small as possible. To achieve the functional requirements we instead generate the qr code locally and save the contents to be encoded. This is done through receiving this data in a HTML form and then saving this information in the database. This information is then saved as a string then using the google charts as the URL for the image encoded into a qr code to be viewed or downloaded . All qr codes follow this process of saving raw data as a string and encoding the data into a QR code such that it is recognised by the user's phone.

**Code Review Save and edit:**

To edit the value of a saved vCard it is imperative to see what is actually saved so the input fields for the form are populated with the data saved inside the database. This makes it easier to know what to edit.

function savevcard(){

var fname=document.getElementById("fname").value;

var lname=document.getElementById("lname").value;

var organization=document.getElementById("organization").value;

var title=document.getElementById("title").value;

var website=document.getElementById("website").value;

var email=document.getElementById("email").value;

var qrname=document.getElementById("qrname").value;

var about=document.getElementById("about").value;

var address=document.getElementById("address").value;

var phone=document.getElementById("phone").value;

var UserID= auth.currentUser.uid;

// access the document containing the vcard information and saves the new data

db.collection(UserID).doc(qrname).set({

fname :fname,

lname:lname,

organization:organization,

title:title,

email:email,

about:about,

website:website,

address:address,

phone:phone,

type:"vcard",

}).then(() => {

window.location.href ="dashboard.html";

console.log("Vcard Created");

})

.catch((error) => {

console.error("Error Writing Document", error);

});

}

function editvcard() {

var UserID = auth.currentUser.uid;

db.collection(UserID)

.doc(localStorage.getItem("qrname"))

.get()

.then((doc) => {

if (doc.exists) {

// auto populated the content of the form with data

console.log("Document data:", doc.data());

document.getElementById("fname").value = doc.data().fname;

document.getElementById("lname").value = doc.data().lname;

document.getElementById("organization").value = doc.data().organization;

document.getElementById("title").value = doc.data().title;

document.getElementById("website").value = doc.data().website;

document.getElementById("email").value = doc.data().email;

document.getElementById("qrname").value = localStorage.getItem("qrname");

document.getElementById("about").value = doc.data().about;

document.getElementById("address").value = doc.data().address;

document.getElementById("phone").value = doc.data().phone;

} else {

console.log("No such document");

}

})

.catch((error) => {

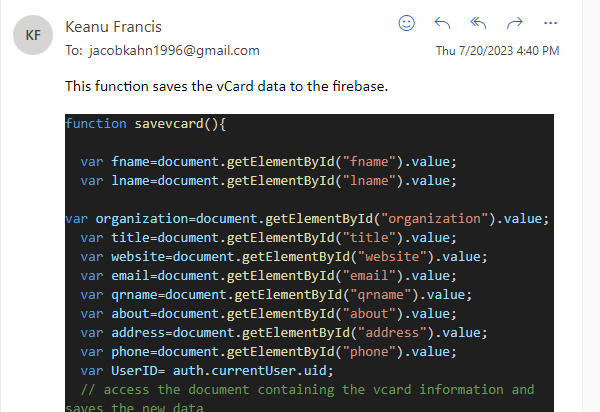
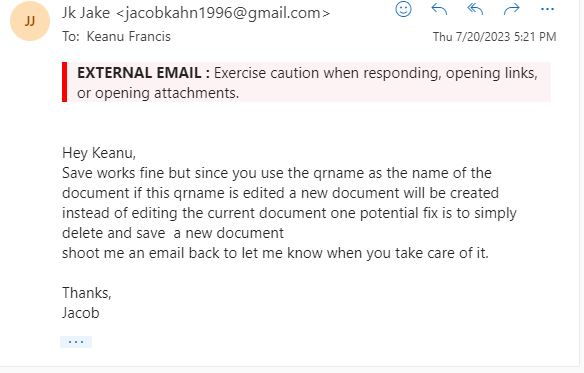
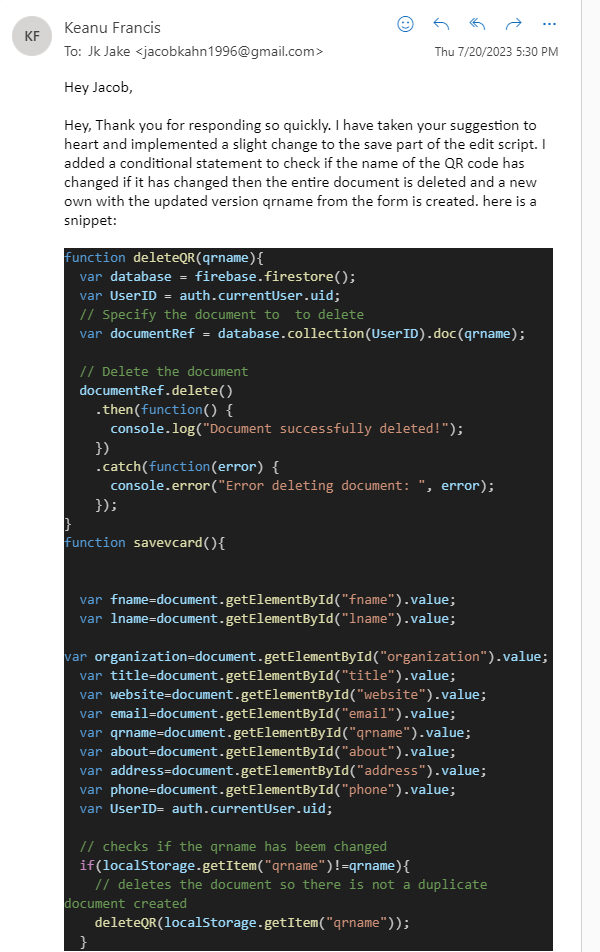
console.log("Error retrieving document:", error);

});

}

After finishing adding the editing portion of the vCard requisition this to be peer reviewed by jakob.

**Peer Review:**

1. 
2. 
3. 

Qcode

Encoding and Generation Peer review:

function generateVCard(qrname) {

return new Promise((resolve, reject) => {

var UserID = auth.currentUser.uid;

db.collection(UserID)

.doc(qrname)

.get()

.then((doc) => {

if (doc.exists) {

console.log("document data", doc.data());

var firstName = doc.data().fname;

var lastName = doc.data().lname;

var email = doc.data().email;

var phone = doc.data().phone;

var organization = doc.data().organization;

var jobTitle = doc.data().title;

var address = doc.data().address;

var about = doc.data().about;

var website = doc.data().website;

const vcard = `BEGIN:VCARD\r\nVERSION:3.0\r\nN:${lastName.trim()};${firstName.trim()};;;\r\nFN:${firstName.trim()} ${lastName.trim()}\r\nORG:${organization.trim()}\r\nTEL;TYPE=CELL:${phone.trim()}\r\nEMAIL:${email.trim()}\r\nTITLE:${jobTitle.trim()}\r\nADR:${address.trim()}\r\nNOTE:${about.trim()}\r\nURL:${website.trim()}\r\nEND:VCARD`;

resolve(vcard);

} else {

console.log("no such document");

reject(new Error("No such document"));

}

})

.catch((error) => {

console.error("Error retrieving vCard data:", error);

reject(error);

});

});

}

async function viewQR() {

try {

const vcardData = await generateVCard(qrname);

// Create a QR code using Google Charts API

const qrCodeUrl = `https://chart.googleapis.com/chart?cht=qr&chs=300x300&chl=${encodeURIComponent(vcardData.trim())}`;

// Display the QR code

const qrCodeContainer = document.getElementById("qrcode");

qrCodeContainer.innerHTML = `<img src="${qrCodeUrl}" alt="QR Code">`;

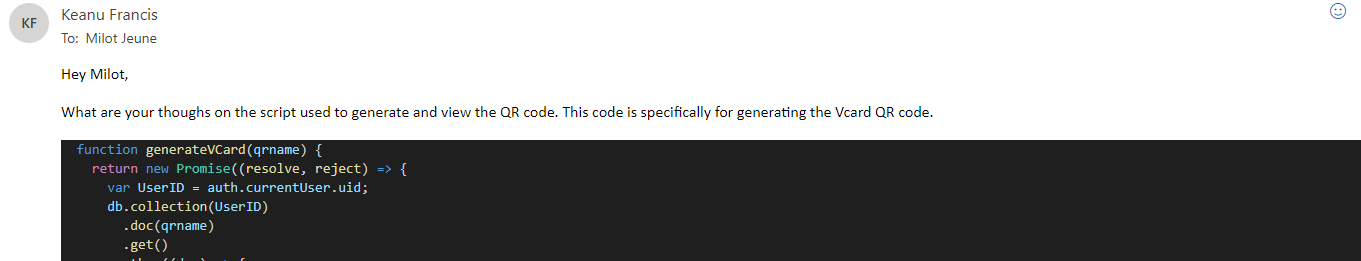
} catch (error) {

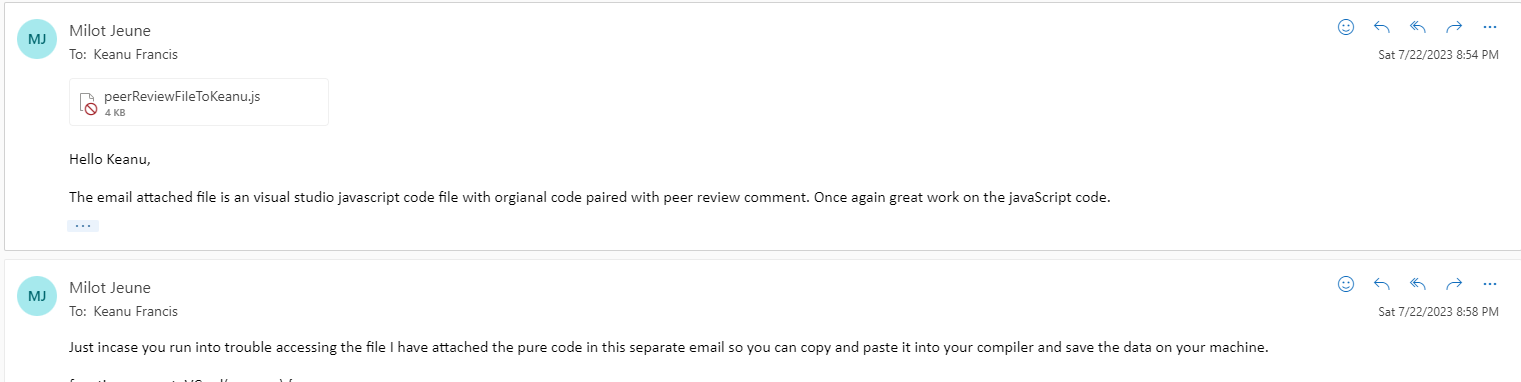
console.error("Error generating QR code:", error);

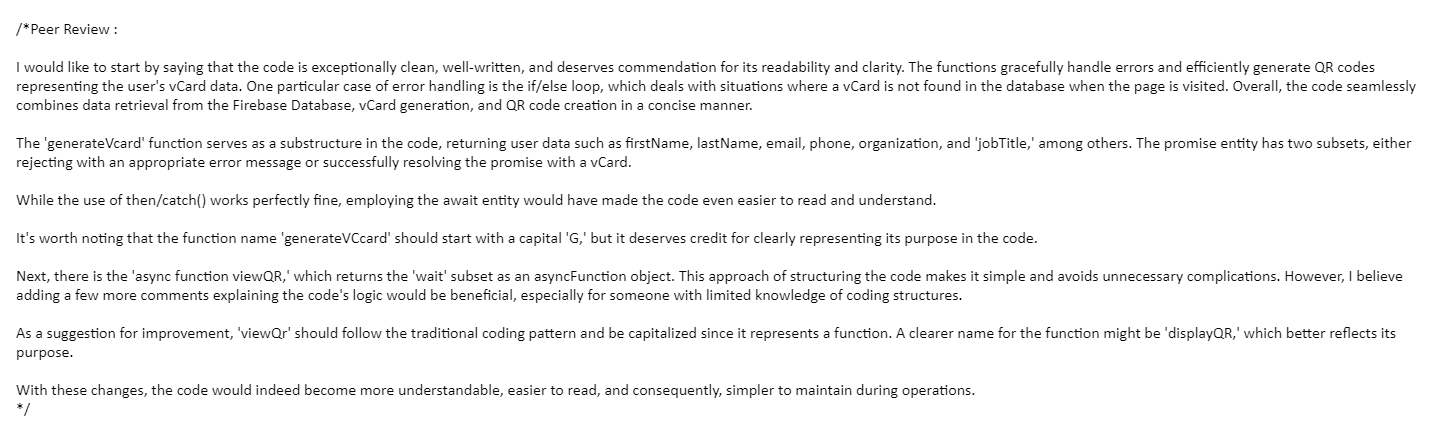
}

}

This was peer reviewed by Milot







**Self-check on best practices for security**

1. Major Assets Being Protected

The major assets that we are protecting in Bizz QR include:

* User Personal Information: This includes users' names, contact information, and other details inputted for QR code generation.
* User Account Information: This includes users' login credentials (username and password).
* Generated QR codes: Generated QR codes are only accessible by the owner, or shared personale.

2. Password Encryption in the Database

* We confirm that we are encrypting passwords in the database. We use a secure hashing algorithm to store passwords. This means that even if the database is compromised, the actual passwords will not be exposed. The stored hash is compared with the hash of the entered password for authentication.

4. Protection Against Competitor DDoS Attacks on Release

We acknowledge the risk of Distributed Denial of Service (DDoS) attacks, especially around the time of release when visibility and traffic are high. To mitigate this risk, we would theoretically implement the following measures:

* Traffic Monitoring and Anomaly Detection: We use tools and services that monitor network traffic and detect unusual spikes in activity or patterns that may indicate a DDoS attack. This allows us to respond quickly if an attack is detected.
* If further issues arise regarding DDoS attacks, we may incorporate a 3rd party solution like Cloudflare to help, identify and solve the issue.

**Self-check: Adherence to original Non-functional specs**

# **List of Non-Functional Specifications**

Performance Requirements.

1. Responsiveness Done: The website will be performant on a wide range of devices. Website elements will dynamically scale to fit both screen size aspect ratio and resolutions.

1. User/event response time Done: The website shall load and update with new information between 100-500 milliseconds and be able to update the database and change website elements within 500 milliseconds

1. Screen refresh time Done: System takes no more than 100 milliseconds to load the home page of the website provided there is adequate bandwidth on the user end.

1. Reliability Done: The system endeavors to be available at all times Regardless if the database is experiencing failure or slow downs.

1. Executions speed Done: The system shall generate and deploy a QR code no less than 3 seconds after execution .

Usability Requirements:

1. The system will be able to be used by anyone with basic understanding of computers; this is done through minimal nested menus, large , clear buttons and prompts that allow users to produce the type of QR code they desire.
2. Mobile Compatibility Done:

○ Bizz QR should be accessible and optimized for mobile devices, including smartphones and tablets.

○ The application should have a responsive design and provide a seamless user experience across different screen sizes and resolutions.

Availability Requirements:

1. Operational Time Done: The website will be available 24 hours a day 365 days a year and so long as there are no interruptions from Github pages there will be no interruptions in the website operation.

1. In the case that there is interruption from the firebase, there will be a clear message stating that there is an interruption. Users can create and change QR code information stored locally on the device and when firebase services are online again users will be prompted if they wish to commit these new changes.

Expected load Done.Security requirements.

1. Username And Password

○ A username and password will be required to enable the save information used to generate the users QR codes. This password will require A capital letter in the beginning and with a number at the end .The password will have a minimum length of 8 and a maximum length of 20 characters.

○ At no point will BIZZ QR disclose or share username and password and if there is any indication of a breach there will be announcements on the homepage alerting users to change passwords if a breach is detected. And until the password is reset no one can access or change the user's information.

1. User Data:BIZZ QR shall not share or use any information added to the users database. This information will remain private and will only be able to be accessed by the user.
2. Encryption: No end to end encryption is not needed since no critical data such as credit card information or social security numbers will be transferred or hosted Database should contain hash encryption for if data is accessed

Storage Done: QR Code Storage and Management:

○ Bizz QR should have a storage system to save and organize generated QR codes.

○ The application should provide options to edit, delete, or update QR codes as needed.

Expected Load: Since the scale of the project is not large we will expect no more than 5 - 20 users using the website and either accessing or making changes to the database at once. The application will be built with functionality to account for this.

Checklist

1)

Title page

Done

2)

Product summary

Done

3)

Usability test plan

Done

4)

QA test plan

Done

5)

Code review

Done

6)

Best practice for security

Done

7)

Non- functional requirements

Done