CEN 4010 - Principles of Software Engineering

Term: Summer 2023

Milestone 5: Final Project Delivery and Demonstration

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** |
| Milot Jeune | **Front End Developer** | **Mlouisjeune2022@fau.edu** |

Date:7/27/2023

Revision History: N/A

Project URL:

[**https://kfrancis2018.github.io/Group-15-Project-BIZZ-QR/**](https://kfrancis2018.github.io/Group-15-Project-BIZZ-QR/)

**Project 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.

**3.3 Milestone documents**

**Executive Summary**

In today's fast-paced business environment, establishing and expanding professional networks is essential for small businesses and students alike. However, traditional methods such as paper resumes and physical business cards can often be inefficient and easily overlooked. Many hiring processes now rely on automated systems that prioritize keyword matching, making it even more challenging for resumes to stand out. Additionally, the inconvenience of carrying and exchanging physical business cards can lead to missed opportunities.

Introducing Bizz QR, a QR code software designed to address these challenges and revolutionize the way professionals connect. Bizz QR is specifically tailored to serve small businesses and students, providing a modern and efficient solution for networking. By transforming contact information into a QR code, Bizz QR enables users to easily share their details and make lasting connections. Bizz QR was designed at FAU and will be made available to FAU students for free wishing to enhance their networking capabilities.

With Bizz QR, the hassle of handling physical resumes is eliminated. Instead, users can input their contact information into the program, which generates a unique QR code representing their details. When networking, all it takes is a simple scan of the QR code using a smartphone or QR code reader to instantly access and save the contact information. Saving that contact information to a personal phone will provide a strong point of contact. Whether you’re a student looking for an employer or a business looking for customers, Bizz QR gets you one step further in prospect.

By leveraging QR codes, Bizz QR ensures that crucial contact information is readily available and easily transferable, enhancing networking opportunities. Whether attending career fairs, conferences, or casual meetups, users can effortlessly share their professional details with others. No longer will individuals miss out on valuable connections due to the absence or loss of physical business cards.

# **Competitive Analysis**

In this competitive analysis we wish to see how our proposed idea for BIZZ QO stacks up against the competition. These will be done in specific sections

1. **QR code creation:** How easy is it to understand the process of creating QR code.
2. **VCard Creation:** how easy is it to create a Vcard.
3. **QR Code Creation Speed:** How fast the QR code can be created and deployed for use.
4. **QR Code editing:** How easily can a user edit a already existing qr code on their account
5. **QR Code Customization:** How much personalization can be done on the QR Code e.g. colors themes backgrounds images.
6. **QR Code diversity**. How many kinds of qr codes can be created .
7. **QR Code quantity**. How many QR codes can be stored.

These will be quantified from a value rom **1-5**

**1**=Poor

**2**=Subpar

**3**=Fair

**4**=Acceptable

**5**=Great

| **Features** | Bizz QR | **.qrcode-tiger.com** | **.flowcode.com** | **myqrcode.com** | **qrcodechimp.com** |
| --- | --- | --- | --- | --- | --- |
| **QR code Creation** | 5 | 3 | 2 | 5 | 5 |
| **VCard Creation** | 5 | 3 | 1 | 4 | 4 |
| **QR Code Creation Speed** | 5 | 2 | 3 | 3 | 5 |
| **QRcode Editing** | 5 | 4 | 3 | 4 | 4 |
| **QR Customization** | 3 | 5 | 4 | 3 | 4 |
| **QR Code diversity** | 3 | 5 | 5 | 4 | 3 |
| **QR code quantity.** | 5 | 2 | 1 | 1 | 2 |

**BIZZ QR:**

Bizz Qr shall have an easy to understand and effective design that offers users an easy method of creating, saving and editing QR codes. The services provided by BIZZ QR will be completely free and with no limitations on the amount of QR codes that a person can save to their account. Since the design and interface of the website will be simple the time it takes to learn the functions of the website will be minimal.

**Qr Tyger:**

Qr Tyger features a cluttered design, offering a wide range of customization options for QR codes, including the ability to encode images into dynamic QR codes. Users can save their QR codes as templates for future reference. However, it is important to note that this service is not free, and user accounts are valid for only one year. To view a QR code, users need to navigate through their account, access the dashboard, and then view or edit the desired QR code.

**Flowcode:**

Flowcode provides impressive customization options, particularly in the realm of background art. Users can choose from various themes, such as Father's Day or Pride Day. The platform offers an intuitive interface, but it appears to lack the option for vCard creation. Additionally, there is a limitation on the number of vCards it can store, allowing only two unless users opt for a paid plan.

**Myqrcode.com:**

Myqrcode.com presents a user-friendly website with straightforward navigation. It offers a moderate level of customization for QR codes, allowing users to choose colors, add or remove text, and incorporate borders. However, this service is not free, and users are limited to a single QR code creation during the 14-day free trial period. The website demonstrates fast response times, enabling convenient and instant QR code editing.

**QR code chimp:**

QR code chimp stands out with its professional design and user-friendly interface. It facilitates quick QR code creation and even allows users to scan images. However, when scanned, the image redirects to another website containing the vCard data, which can then be saved to the user's contacts. It is worth noting that QR code chimp imposes a monthly scan limit of 1000, and the process of adding information to contacts involves an additional step. Furthermore, the website does not offer the option to log in with Google credentials.

**Planned Advantages:**

The major focus of BIZZ QR is ease of use. Many QR code sites over expose the user with large amounts of options and at times it is not clear if the information is required for creation. While a seasoned user of these websites might find it easier to deal with, new users are likely to be overwhelmed by the amount of options. Most are unlikely to need such a diverse amount of QR Codes. Another aspect is the limitations placed on non paying users such as limited scanning and a cap on the amount of qr codes that can be stored on the users account. These problems are something that BIZZ QR shall aim to fix. It shall have a very simple to understand interface and no caps on QR storage. These features will be more beneficial to users who prefer to have a simple and fast method of QR code creation without dozens of unnecessary features that clutter the interface. Most of which the user is unlikely to use.

# **Data Definition**

| Name | Type | Usage | Meaning |
| --- | --- | --- | --- |
| BIZZ QR | Actor | Domain Name | Name given that represents all pages on the website. |
| BIZZ QR System | Hardware and Services | Use Case Scenario | Name given to all functional system in the Bizz QR Website including both front and back end |
| User | Actor | Use Case Scenario | The person that is using the QR code System, The creator of the QR code. The person scanning the qr code. |
| Vcard | DATA | Use Case Scenario | Contact information Encoded into the Vcard format |
| QR codes | DATA | Use Case Scenario | Vcard or other information such as websites encoded into a scannable image. |
| social media profiles | DATA | Use Case Scenario | Links to the users Social media Accounts such as facebook, twitter ect. |
| Contact Information | DATA | Use Case Scenario | Contains data including , Name , Phone number , **Social media Profiles** |
| Log in with Google | Service | Use Case Scenarios | Serves as an alternate means for login provided the user has a google account |
| Edit | Service | Site User Service | Allows to user to change contact information |
| Share | Service | Site User Service | Allows users to send Vcard data for QRcodes via email or messages. |
| Log in | Service | Site User Service | Allows users to login and save contact information for Vcard QR Creation |
| Save | Service | Site User Service | Allows users to save the The generated Vcard and QR code data |
| Download | Service | Site User Service | Allows users to download Vcard for QRcode Data directly to the device |
| Contact Sharing | Service | Site User Service | Term used to describe sharing contact information |
| FireBase | Database server | Site User Service | The services used to store user information for later use |
| Application | Actor | Use Case Scenarios | The mobile compact version on BIZZ QR |
| Home Page | User Interface | User Interface | The first page a user will see |
| Dashboard | User Interface | User Interface | Can only be seen once a user has logged in contains saved QR codes and Vcard data |
| Networking | User Interface | Use Case Scenarios | The act of sharing contact information with other people. |
| Website | User Interface | User Interface | The front end for user interaction |
| Github Pages | Live server | Use case | The server that hosts the BIZZ QR live website. Also stores non user data |

# **Use Cases**

This use case outlines the various functions and capabilities offered by Bizz QR to enhance networking and contact sharing for users.

**Use Case: Create QR Code**

A user wants to create a QR code so that they are able to share their personal/business information with users.

**1. Description:**

This use case describes the process of how a user can create a QR code using the Bizz QR website.

**2. Actors:**

2.1 User

2.2 Bizz QR System

**3. Preconditions:**

3.1 User has an active internet connection.

3.2 Bizz QR System is available.

3.3 User has an account on the website.

3.4 User has an email account to register with the website.

**4. Primary Flow of Events:**

1. User will access the Bizz QR website or application.

2. User Navigates to the “Login” page.

3. User enters their account information to login or does “Login with Google”.

4. If the user does not have an account, go to Alternative Flow.

3. User navigates to the "Create Bizz QR Code" section.

5. User enters their contact information, such as name, phone number, email, and social media profiles.

6. Bizz QR System generates a unique QR code containing the user's contact information.

7. The QR code is displayed for the user to save or share.

**5. Alternative Flows:**

1. User does not have an account registered with Bizz QR.
2. Users will access the “Create an Account” page.
3. Bizz QR will display a form for the user to register an email and password.
4. User enters information and clicks “Register”.
5. User credentials are saved into the Firebase database.
6. Users are given the option to go back to the “Login” page.
7. End Alternative Flow, return to Primary Flow, step 3.

**4.2 Use Case: Scan Bizz QR Code**

**1. Description:**

This use case describes the process of a user scanning a QR code generated through Bizz QR to view another user's information.

**2. Actors:**

2.1 User 1

2.2 User 2

2.3 Bizz QR System

**3. Preconditions:**

3.1 User 1 and 2 have an active internet connection.

3.2 Bizz QR System is available.

3.3 User 1 has an account on the website.

3.4 Users have a smartphone with QR scanning capabilities.

**4. Primary Flow of Events:**

1. User 1 encounters another individual with a Bizz QR code.

2. User 1 logs into the Bizz QR application and selects the "Bizz QR Code" option.

3. User 2 scans the QR code using their device's camera.

4. The contact information is displayed to the user 2 for review.

5. User 2 can choose to save the contact information to their device's address book or directly connect on social media.

**5. Alternative Flows:**

1. User does not have an account registered with Bizz QR.
2. User will access the “Create an Account” page.
3. Bizz QR will display a form for the user to register an email and password.
4. User enters information and clicks “Register”.
5. User credentials are saved into the Firebase database.
6. User is given the option to go back to the “Login” page.
7. End Alternative Flow, return to Primary Flow, step 2.

**4.3 Use Case: Choosing QR Code**

**1. Description:**

This use case outlines the steps for a user to create a Qr code

**2. Actors:**

2.1 User

2.2 Bizz QR System

**3. Preconditions:**

3.1 User has an active internet connection.

3.2 Bizz QR System is available.

3.3 User has an account registered on Bizz QR.

**4. Primary Flow of Events:**

1. Users access the Bizz QR website or application.

2. User navigates to the Create QR code section.

3. User User is prompted to choose a type of qr code to create

4. Bizz QR System the adds data to the field corresponding to the type of qr code (see section 5: **QR Creation Extended**)

5. User saves the qr code to database

**5. QR Creation Extended:**

**1. Description:**

This use case outlines the steps for a user to create the 6 different Qr code that will be displayed when their QR code is scanned.

**2. Actors:**

2.1 User

2.2 Bizz QR System

**3. Preconditions:**

3.1 User has an active internet connection.

3.2 Bizz QR System is available.

3.3 User has an account registered on Bizz QR.

3.4 User chooses the Create QR Code.

**4. Primary flow of events.**

**4.1 Vcard Creation:**

1. User enter Qr name, First name, Last name, Address, about(personal description), organization, title, phone number
2. User clicks save QR code

**4.2 Location Creation**

1. User enters Qr name, Street, city, state and postcode
2. User clicks save QR code

**4.3 Call creation**

1. User enters Qr name, and phone number
2. User clicks save QR code

**4.4 Text Creation**

1.User enters qr name and adds a block of text

2. User click saves QR code

**4.5 Message Creation**

1. User enters phone number, SMS message
2. User clicks saves qr code

**4.6 website Creation**

1. User enters qr name and url of website
2. User clicks saves qr code

**4.4 Use Case: Share QR Code**

**1. Description:**

This use case outlines the various functions and capabilities offered by Bizz QR to enhance networking and contact sharing for users.

**2. Actors:**

2.1 User

2.2 Bizz QR System

**3. Preconditions:**

3.1 User has an active internet connection.

3.2 Bizz QR System is available.

**4. Primary Flow of Events:**

1. Users want to share their QR information conveniently.

2. User opens the Bizz QR application and selects the "View QR Code" option.

3. Bizz QR System generates a shareable link that downloads the image of the qr code

4. The user can share the QR code through various communication channels, such as email, messaging apps, or social media.

**5. Alternate Flows:**

1. Users want to share their contact information conveniently.

2. User opens the Bizz QR application and selects the "View QR Code" option.

3. User selects Download Vcard Bizz QR System generates a shareable link that downloads the vcf of the qr code.

4. The user can share the vcf through various communication channels, such as email, messaging apps, or social media.

**High Level Functional Requirements**

**Non-Member expectation**

1. **Creating Account (1- Must Have)**

**○ 1.1** The system shall allow the user to create an account by storing

UserID, Password, Date of Birth, First Name, Last name, Location, and

answer to security question/phone number.

The system shall not allow the User to Create an account if the UserID

choose by the User already exist in the System’s Database Also the

system shall prevent the user from creating an account if the User’s

chosen password does not match the re-enter password field. System

shall prevent the creation of the user’s account following fields is not filled.

Fields that have to be filled are First Name, Last Name, Location, UserID,

Password, Re-enter Password, Security Answer Security Question or

Phone number, and Date of Birth.

**○ 1.2 Stimulus/Response Sequence**

* + i. User enters a UserID (same as email)
  + ii. User enters a Password
  + iii. User re-enters Password for confirmation
  + iv. User shall enter their First and Last Name
  + v. User shall enter their Location
  + vi. User shall enter their date of birth
  + vii. User shall provide to an answer to given Security Question Or
  + provide their phone number
  + viii. System shall check if UserID is available
  + ix. System shall validate Password
  + x. System shall store user Name, date of birth, and answer to
  + selected security question/phone number
  + xi. System shall confirm that the account was created to the User
  + xii. System will have a button to redirect the user back to the home
  + page at will

**○ 1.3 Function requirement label**

i. REQ 1.1 Creating Account

1. **Information (1- Must Have)**
   * Users will be able to view Info about Bizz QR on the homepage about the functionality.

**Members expectations**

1. **2. Edit Profile(2- Desired)**

**○ 2.1** Users shall be able to edit their profile by providing a name, business name,address, phone number, email, any social media , and date of Birth. The System shall store that information when the user clicks the save button.This information will be used to generate and manage the QR code. The System shall prevent any changes to the User’s profile if any of the fields are left blank. The user has to type the information in a valid format in order for the system to store the information.

1. **QR Code Generation: (1- Must Have)**
   * The system should be capable of generating QR codes from user-provided contact information.
   * It should support various formats, such as vCard, to encode contact details efficiently.
   * The generated QR codes should be visually appealing and scannable by popular QR code scanning apps.

4.1 Vcard - User will be able to create a vcard then encode the information into a scannable qrcard open the users contacts act then populate the data fields with the data stored on in the vcard

4.2 Call - User will be able to create a Scannable Qr code that will immediately call the phone number that is encoded into the qr code

4.3 Message -User will be able to save a message and a phone number in the database that will then be encoded into a qr code that can be scanned. Once scanned it will open the messaging app and populate the massage field with the saved message.

4.4Text - User will be able to encode The user will be able to encode a string of text into a qr code this data can then be viewed by the scanner when scanned.

4.5 Location -User will be able to create a location qr code that is encoded with street, city state and postcode. Once scanned it will open the google maps application and immediately find the address saved. If the google maps application is not installed it will simply open a browser to that location.

4.6 Website - Users will be able to save the url of a website. Once encoded into the qr code once scanned it will open the browser and navigate to that web page.

1. **QR Code Sharing(1- Must Have):**

* Users should be able to download the QR code image or generate a shareable link.

**5.1(1- Must Have)** Users will be presented with a button that will allow them to immediately download the qrcode. In the case of vcard the user will be able to download both the qr code and the .vcf file itself.

* Users should be able to view QR code on device that will be scanned by another phone device with supporting camera

1. **Help and Support: (2- desired)**
   * The system should include comprehensive documentation or tutorials to assist users in using the platform effectively.
2. **User Management: (2 - Desired.)**
   * User authentication and password reset functionalities should be implemented to ensure data security and privacy

# 

# 

# 

# 

# **List of Non-Functional Specifications**

**Performance Requirements**.

1. **Responsiveness:** 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.
2. **User/event response time**: 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
3. **Screen refresh time:** System takes no more than 100 milliseconds to load the home page of the website provided there is adequate bandwidth on the user end.
4. **Reliability:** The system endeavors to be available at all times Regardless if the database is experiencing failure or slow downs.
5. **Execution speed:** 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:
   * 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:** 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.
2. 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.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.
2. **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.
3. **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

**Storage:** 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.

# 

# 

# **High-Level System Architecture**

1. **Discord:**Discord is used for vocal communication for each team member all communication both voice and text will be done through discord.
2. **Google Analytics:**Google Analytics is used to get analytics regarding the website to better improve the products.
3. **Google Docs:**Google Docs was used to coordinate the creation of the requirement documentation allowing for developers to make changes and add to the documentation in real time.
4. **Vcard.js:** A library used as an extension of javascript that allows for the easy creation of Vcards using java script. **License** (https://github.com/enesser/vCards-js/blob/master/README.md). Manually creating a text string then encoding the format for the vcard is faster than using Vcard.js so it was decided that it is not necessary.
5. **QRcode.js**: QRcode.js is also an extension of javascript that can be used to generate QRcode images using proper test input. Well be paired with Vcard.js for QRcode based vcards. QRcode uses the standard MIT **License** (<https://opensource.org/license/mit/>) QRcode.js no longer necessary since google charts provides the same functionality.
6. **QR API Google charts** : The QRcode functionality of the google charts api will also extend QR code creation to more than just vcards and offer more customization if needed.
7. **Trello:** Trello workspace will be used to keep track of current sprints and objectives using a to do list.
8. **Development environment**: Visual Studio Code is the code editor that will be used to develop the code for the website. To do this we will utilize the Live Server extension provided by Ritwick Dey. This is also covered by the MIT License(https://github.com/ritwickdey/vscode-live-server/blob/HEAD/LICENSE)
9. **Firebase:** Firebase realtime database will be used to store userdata
10. **Github pages:** Github pages will be used to host the website and its functionality.
11. **Github Repositories**: Using a central repository for the project developers will be able to work
12. **Languages:** The programming languages that will be used will be.
    1. **HTML:** Will provide the basic structure of the website and allow the browsers to view the website.
    2. **CSS :**CSS will allow the website to create a pleasing visual appearance including colors and other visual functionality.
    3. **Javascript :**Javascript will enable dynamic interaction of the elements in the html and we will also use javascript to communicate with the firebase database.
13. **Browsers:** BIZZ QR will be compatible with most major browsers that support javascript such as Google Chrome Safari FireFox and Microsoft Edge.
14. **Udemy :** was used to learn new technologies for front-end web development and stay updated on emerging trends in the industry that could be applied to our applications. For example, React was utilized for the front-end development. As this is the first time Front-end Developer Milot is creating the ICP application, we will closely monitor the progress to ensure it aligns with the project sprint schedules.
15. **Figma** :Figma was utilized to create a custom one-of-a-kind favicon and various infographics. Although Milot has not yet rolled out the infographics, they are scheduled to be launched in the upcoming weeks. Currently, Milot is exploring alternative platforms that offer improved and more efficient customization features in conjunction with Figma
16. **Blender**: Although Blender has not been utilized for the project, it is currently being explored and practiced to evaluate its suitability for 3D rendering. Milot will provide an update to the team regarding the feasibility of using Blender and whether he intends to incorporate designs created with the software into the project.
17. **SketchUp:** SketchUp is a user-friendly 3D modeling software that Milot is currently testing for product design purposes. Its simple and intuitive interface makes it accessible to newcomers in the field. Additionally, SketchUp provides a vast library of pre-made 3D models that can be utilized in website design. Milot will update the team regarding his plans for incorporating SketchUp into the project once a decision is made.on several aspects of the website in parallel by forking from the main repo. These forks can then be merged with the main file once it has undergone extensive testing. It also allows developers to view the changes made to the code made by other programmers.
18. **Google Maps Api:** Since we decided to add the ability to save locations to a qr code. It is necessary to find a way to query locations that a user saves. This is done through the <https://www.google.com/maps/search/?api=1&parameters> functionality that encodes data to the url format and finds the location using google maps. We have also found that if google maps is, it will immediately open and use the app instead of a browser.

**Database organization:**

| Collections | Documents | Fields |
| --- | --- | --- |
| USER ID | QR NAME | QR CONTENT |

The databases will contain several different collections. Each collection will be created using the user's USERID. The documents will have a name based on the user's selection, and the content field will depend on the type of QR code that is created. There will be six different QR codes (call, message, Text, location, type, and website) that a user can add to the database in any quantity. Each document will contain a field called 'type' that will aid in organizing, displaying, and editing the QR code. Allowing the user to choose the name of each QR code makes it easy to use the snapshot functionality to create a list of all QR codes in the dashboard using the user ID, the QR name, and the type.

1. Vcard: Used to create business cards in the form of qr codes that can be instantly added to a users contacts

| COLLECTION | DOCUMENTS | FIELDS |
| --- | --- | --- |
| [Current User ID] | “Keanus’s business card” | Fname: *Keanu*  Lname: *Franics*  Organization: *FAU*  Title: *Student*  Website: *keanu.com*  Email: *kfrancis2018@fau.edu*  About :*I am a senior engineering student…….*  Address: *1995 Dade Ave #1973, Boca Raton, FL 33431*  phone : *9545943216*  Type: *vcard* |

1. SMS: Creates a text message and auto-populates the message and the phone number it should be sent to .

| COLLECTION | DOCUMENTS | FIELDS |
| --- | --- | --- |
| [Current User ID] | “Keanus’s SMS” | Message: *Hi my name is……*  phone : *9545943216*  Type: *message* |

1. CAll: Automatically dials the chosen phone number

| UCOLLECTION | DOCUMENTS | FIELDS |
| --- | --- | --- |
| [Current User ID] | “Keanus’s call” | phone : *9545943216*  Type: *call* |

1. Location: If google maps is installed on the phone instantly finds the location encoded in the qr code or simply opens a browser to google maps and then finds the location.

| COLLECTION | DOCUMENTS | FIELDS |
| --- | --- | --- |
| [Current User ID] | “Keanus’s location” | Street: *4995 dade ave #1973*  City: *Boca Raton*  State: *Florida*  Postcode: *33431*  Type: *location* |

1. Text: Encodes a string of text into a qr code that once scanned can then be viewed

| COLLECTION | DOCUMENTS | FIELDS |
| --- | --- | --- |
| [Current User ID] | “Keanus’s Text” | name : *company information*  text: *this a company based in ……..*  Type: *text* |

1. Website:instantly navigates the user to the desired webpage

| COLLECTION | DOCUMENTS | FIELDS |
| --- | --- | --- |
| [Current User ID] | “Keanus’s Website” | URL: *https://keanuwebsite.com*  Type: *website* |

**Filtering and Searching**

All searches and filtering of the database will be done using the User ID of the current user and the QR code name. For example, in the dashboard, a table will be created containing the name and type of each QR code in the database. This is done by using the snapshot functionality to create an array containing all the current document names. With this list of document names (QR names), we can then use the user ID and QR code name to find the type. By iterating through the list in a loop until we reach the end, I can print a list of all QR codes saved in the database and their types. I can also use JavaScript to create buttons that have a unique ID corresponding to each QR code name. Once a button is clicked, I can retrieve the ID of the button, which contains the QR code name, and save it to localStorage. Depending on the button (EDIT, DELETE, or VIEW), I can load the appropriate page to view, edit, or delete the QR code. Since I can simply use the ID of the button, which would be the same as the document name, it simplifies the process.

**Data types**

The only data that will be stored in the server are strings containing the data use to generate the qr codes details on how this data is stored and organized is described above.

# 

# 

# 

# 

# 

# **High-Level UML diagrams**

# 

# 

# **UML Component Diagram**

# 

# 

# 

# UML CASE DIAGRAM

# 

# 

# Key Risks

1) Skills risks (The technology involved is basic web development, which can be grasped by entry-level programmers in a matter of hours. For more specific information, please refer to the system architecture and the programming languages used.),

2)Schedule Risks: The website is on track for deployment, provided that it adheres to the guidelines for future milestones. Additional features will be implemented as time permits.

3) Technical risks (Front End style to make website user friendly and more appealable.This will make users more likely to revisit the app),

4) Teamwork risks (There is a concern with one team member who has not engaged with any team documents or class milestones..);

5) Legal/content risks (Currently, there are no identified legal or content-related risks).

The front-end design of the website presents the highest risk at launch due to its generic style. However, the risk associated with teamwork is minimal and manageable. As team members become more familiar with the system's architecture, the skills risk will be reduced.

**2) summarize instructor’s feedback of Milestone 3 and 4**

Milestone 3 Feedback:

Our Milestone 3 submission was praised and no change was asked.

Milestone 4 Feedback:

No feedback was provided.

**3.4 Screenshots of actual final product as shown in the demo**

Homepage:

|  |
| --- |

DashBoard:

|  |
| --- |

QR type:

|  |
| --- |

Vcard Generation Page:

|  |
| --- |

QR code Display:

|  |
| --- |

**3.5 Google analytics plot for your website (1 page)**

| **Realtime Table** |
| --- |
|  |

| **Script added to each webpage** |
| --- |
|  |

| **Google Analytics Report** | |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**3.6 Team members contribution**

1) List each team members’ contribution to the project. Use “Project Peer Evaluation”

(posted on Canvas under “Resources”), to give each team member’s contribution

points. Note: All team members should reach consensus on those points.

| Members | Contribution Points |
| --- | --- |
| Jacob Kahn | (20/20) |
| Keanu Francis | (20/20) |
| Sunny Chen | (20/20) |
| Temel Durak | (20/20) |
| Milot Jeune | (5/20) |

2) Describe each team members contribution to teamwork (technically and any other

contribution). No more than half a page. Bulletin format is fine

* Keanu Francis
  + QR code generation, Dashboard creation, Github management
* Jacob Kahn
  + Front end css, Liaison
* Sunny Chen
  + Firebase authentication (email and password), front end creation.
* Temel Durak
  + Documentation, Database Management (filtering and search)
* Milot Jeune
  + Front end developer, documentation.

| Name | **Role** |
| --- | --- |
| Jacob Kahn | **Team Leader, Product Owner, Front End Developer** |
| Keanu Francis | **Back End Developer , Scrum Master, Github Master** |
| Sunny Chen | **Back End Developer** |
| Temel Durak | **Front End Developer** |
| Milot Jeune | **Front End Developer** |

3) Number of submissions each team member made to GitHub team development

Branch

Keanu Francis: 34 Commits

Keanu was the designated github master.

**Post-project analysis:**

After wrapping up our recent project, our team took a moment to reflect and found some important insights, learning opportunities, and ways we can do better next time. One of our main challenges was figuring out how to properly format strings using JavaScript for the Google Charts API. We were newcomers to this API and ended up doing a lot of experimenting before we got it right.

Additionally, we bumped into issues with data synchronization. We needed to ensure that our functions, which were dependent on info from our internal storage and the database, worked well with asynchronous data. This brought to light some gaps in our understanding of handling data efficiently.

On top of these, we also had to crack the code on how to format data so it worked well on both Android and iPhone, which was quite a task! But it wasn't all struggles. We did find a silver lining in the form of collections, documents, and fields. They were easy to work with and made filtering and searching for individual items from users a breeze.

From these experiences, we learned a great deal about data management and synchronization. We also learned valuable lessons on how to format data for different platforms, knowledge that will undoubtedly be useful for our future projects. If we were to do this project again, we'd focus more on understanding the Google Charts API and coming up with efficient strategies for handling asynchronous data. We'd also spend more time upfront getting to grips with how to format data for different platforms to avoid hiccups down the road. And also add Google Login for ease of use.

We didn't manage to finish all the features we wanted due to these challenges, particularly with data formatting and synchronization. But we're not discouraged. Instead, we see the unfinished tasks as valuable lessons, and we plan to tackle them head-on in future projects. These experiences, although challenging, helped us grow as a team, enriched our technical skills, and taught us how to solve problems more robustly. They also gave us some great real-life examples to share in job interviews to show our adaptability and resilience when tackling tricky tech issues.

**Demo:**

BizzQR Demo URL here: <https://youtu.be/cCsmgAQMmQw>