

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Group Project Final Report



**QReactive - A QR code web-app**

*Members*

BI12-218	Nguyễn Đức Minh Khôi
BI12-215	Đàm Hữu Khoa
BI12-229	Võ Minh Kiệt
BI12-200	Lương Quang Huy
BI12-205	Nguyễn Ngọc Khanh
BI12-206	Lê Quang Khánh

*Supervisor*

Mr. Phạm Hồng Sơn - AKB Software

January, 2024

# Table of Contents

<b>I. Introduction</b>	<b>4</b>
1. Overview	4
2. Motivation	4
3. Objectives	5
4. Proposed System	5
<b>II. Requirement analysis</b>	<b>6</b>
1. Overall system requirements	6
2. Use cases	7
2.1. Use cases overview	7
2.2. Use cases for registered users	7
2.3. Use case for guest users	7
3. Use cases specifications	8
3.1. Authentication	8
3.2. Create QR code	9
3.3. Create QR code	9
3.4. Scan QR code	9
3.5. Search Vcard	9
<b>III. Design &amp; Implementation</b>	<b>9</b>
1. System Architecture	9
1.1. Generate QR code sequence	10
1.2. Edit QR code sequence	11
1.3. Delete QR code sequence	12
1.4. Search QR code sequence	12
2. QR Code generation development	13
2.1. Library Selection:	13
2.2. Integration with Node.js:	13
2.3. Error Handling and Validation:	13
2.4. Storage and History:	13
2.5. Testing and Validation:	13
3. Tools and Techniques	14
3.1. Programming Languages	14
3.2. IDE (Integrated Development Environment)	14
3.3. Framework	14
3.4. Database System	14
3.5. Version Control and Collaboration	14
3.6. Security	14
3.7. API Testing Tool	14
4. Designs overview	15
4.1. Database design	15
4.2. Web design	17
<b>IV. Overall Evaluation</b>	<b>20</b>
1. Out come	20
2. Areas for improvement	20
<b>V. Project Management</b>	<b>21</b>
1. Initiation	21

2. Planning	21
3. Implementation	21
4. Monitoring and control	21
5. Project close	21
<b>VI. Reference</b>	<b>22</b>

## List of Acronyms

<b>Acronyms</b>	<b>Definitions</b>
QR code	Quick-Response code
Vcards/VCF	Virtual Contact File
UI	User Interface
DB	Database

## List of Figures and Tables

<i>Figure 1. Quick Response code example</i>	4
<i>Figure 2. System diagram</i>	6
<i>Figure 3. Use case diagram</i>	8
<i>Figure 4. Generate QR code sequence diagram</i>	10
<i>Figure 5. Edit QR code sequence diagram</i>	11
<i>Figure 6. Delete QR code sequence diagram</i>	12
<i>Figure 7. Search QR code sequence diagram</i>	12
<i>Figure 8. Home page</i>	18
<i>Figure 9. Text QR code generate</i>	19
<i>Figure 10. QR code management</i>	19
<i>Figure 11. Personal info management</i>	20

# I. Introduction

## 1. Overview

The purpose of the report is to document the design, development, and implementation of QReactive, a web-app designed to create and manage QR codes suitable for a wide range of applications.

The report provides a detailed description of the requirements of the design and implementation of the web-app. It also includes a summary of the key achievements of the project and recommendations for future work. The report concludes with a list of references for further information.

## 2. Motivation

When almost every information is displayed under digital forms, we need faster and more reliable ways to deliver them to users, and that's how QR codes are popular.

A QR code (short for Quick Response code) is an array of black and white squares or pixels set in a grid that stores data for a machine to read. Any devices capable of reading and processing images can “scan” QR codes and almost immediately bring results, not to mention the QR code itself can be saved and displayed under various methods.



*Figure 1. Quick Response code example*

Some popular websites that help you create and customize your own QR codes are Me-QR, GenQRcode and QR-code-generator.com. These apps do a great job at creating and customizing your QR code, however the direction we are proposing for our web-app is more social media oriented by implementing a search function for public Vcard QR codes.

### 3. Objectives

We aim to create a web-app that can create QR codes, the types we aim to create are as follow:

- **Hyperlinks:** Converting hyperlinks/Urls of other websites to QR code which the user can scan and be redirected to the link destination.
- **Text:** Converting basic text to QR code, when scanned it will display text in a window.
- **Wifi:** Converting your wifi name and password into QR codes allowing others to scan the QR code and be connected automatically without having to reveal your password.
- **Email:** Converting emails into QR codes, when scanned by others will redirect you to their email page.
- **Vcard:** also known as VCF (Virtual Contact File), is a file format standard for electronic business cards. You can choose between a dynamic Vcard or a static one that when scanned the info will be autofilled in your contact.
- **Custom:** Users can input custom information depending on their needs.

Secondly, we want to create a way for users to manage the QR codes they have created. This includes the ability to download, edit, delete and search for the QR codes you have created.

Thirdly, we want to create the basis for a social media type system where users can find public Vcard QR codes, this will allow users to quickly find and save contact info of others.

## 4. Proposed System

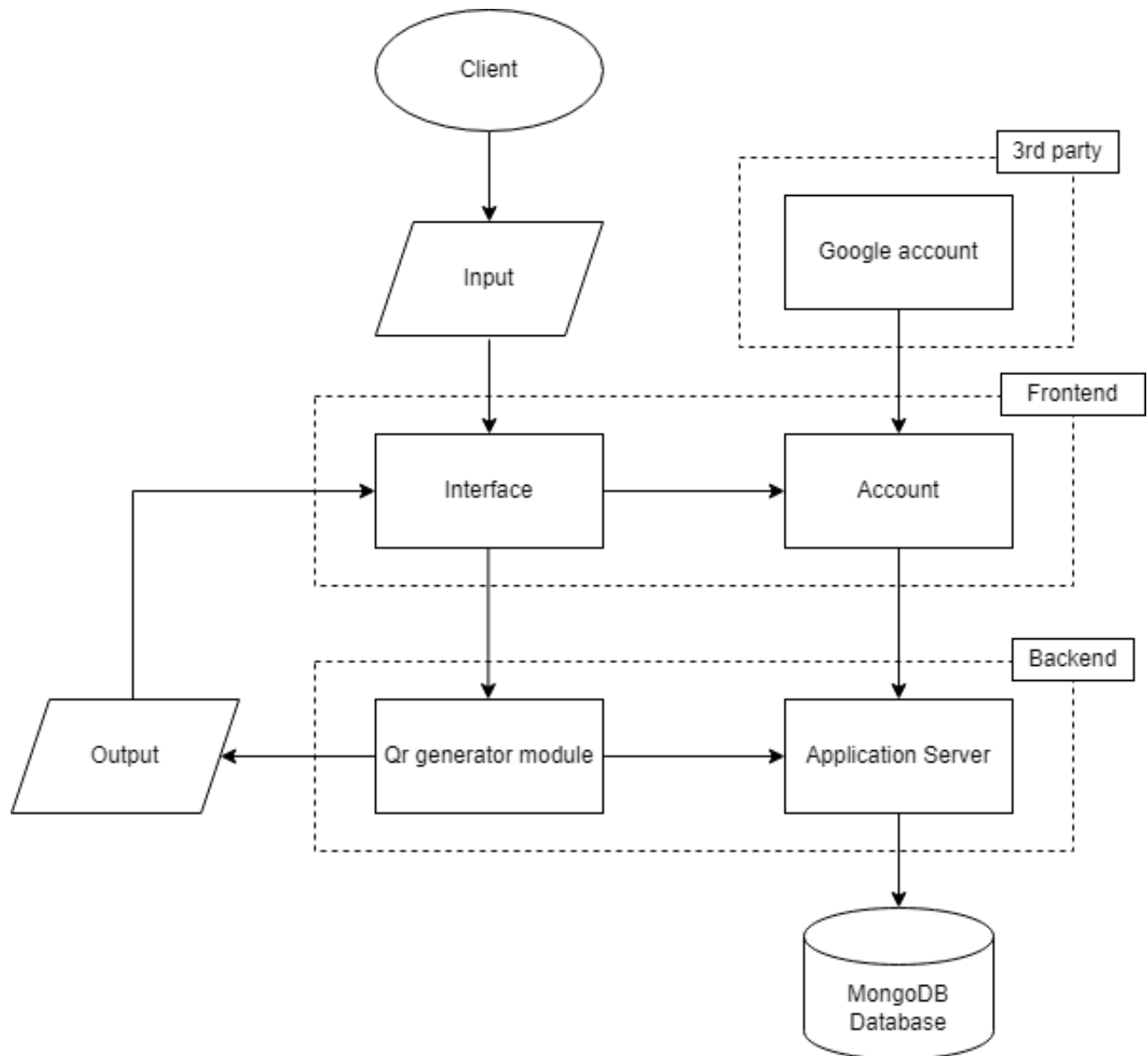


Figure 2. System diagram

## II. Requirement analysis

### 1. Overall system requirements

To make QReactive easily accessible for everyone, we decided that the application should be in website format since a website can run on any device. Therefore, we utilize Next.js, based on the React framework – one of the most popular frameworks for website development. In order for customers to manage and review their generated QR codes, databases will also be involved to store information about customers' accounts and the history of generated QR codes. Consequently, our application will run smoothly on all devices.

## **2. Use cases**

### **2.1. Use cases overview**

There are two main types of users, the registered user and the guest user. Both users can create QR codes with or without an account. However registered users have access to more features.

### **2.2. Use cases for registered users**

Registered users are users with either an account created through the website or an account connected to google. Having an account allows them to see and manage what QR codes they have created. They can delete, edit, download and search for the QR codes.

### **2.3. Use case for guest users**

Guest users are users without an account and are only looking to create a quick QR code. These users can create all types of QR codes except for the dynamic Vcards QR code which needs an account to function. Guest users also can't manage the QR codes that they have created. However they can still use the search function to search for and download other public Vcard but cannot save it to an account.



Figure 3. Use case diagram

### 3. Use cases specifications

#### 3.1. Authentication

As mentioned before a registered user is a user which has an account that was created through the website or by using google. A login user has the option to manage their account by changing their password or general info.



### 3.2. Create QR code:

Overall, the user can create 6 main types of QR codes: Link/URL, Custom, Wifi, Text, Email, Vcard by selecting the respective button.

### 3.3. Manage QR code

Here the user can manage the QR codes they have created. Users can download, edit, delete and search for the QR codes they have created. This will only work with registered users.

### 3.4. Scan QR code

This use-case is used to describe what would happen if users were to scan the QR codes.

### 3.5. Search Vcard

Users can choose to make their Vcard QR code public or private; this will allow other users to search for them by name, email, company. This can be expanded on later to create a type of social media network allowing users to quickly search for and download/add the user contact to your phone. You can also choose to save QR codes to your account to check back later.

## III. Design & Implementation

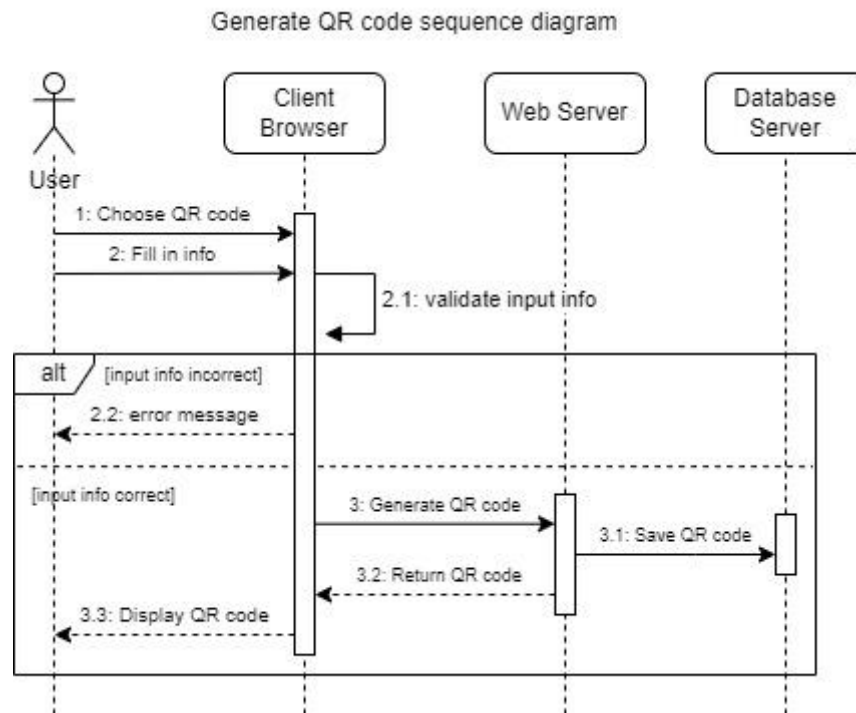
### 1. System Architecture

QReactive consists primarily of three components:

- **Client Browser:** This represents the FrontEnd. It is responsible for receiving user input, sending HTTP requests and calling APIs (with or without payload from user input), and displaying the response.
- **Web Server:** The initial element of the BackEnd responsible for creating the QR codes. As well as managing and transferring requests to the backend database.
- **Database Server:** The second component of the BackEnd responsible for storing data in a secure manner while maintaining accessibility and integrity. The database will store all information from the user's account to the created QR codes.

We've chosen 4 sequences that you will use when managing your QR codes.

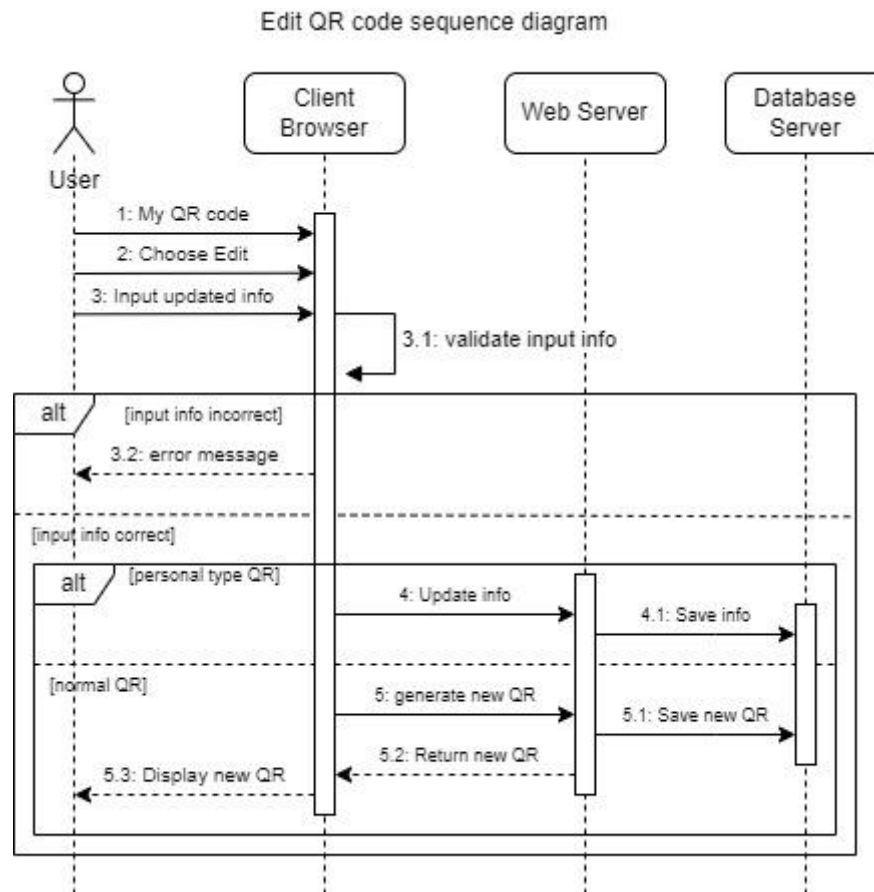
## 1.1. Generate QR code sequence



*Figure 4. Generate QR code sequence diagram*

Any users no matter if they have an account or not will be able to choose between the most options of QR codes (except for dynamic/personal Vcard) and generate a QR code for themselves. The user will choose the QR code and fill in the info they want using the UI, a QR code will be generated through the system and return the user while saving a copy to the Database ( this is for convenience when developing other parts of the app). Do note that unregistered users have no way of checking what QR code they have created.

## 1.2. Edit QR code sequence



*Figure 5. Edit QR code sequence diagram*

Registered users with an account will have the option to edit their old QR codes. From the sequence diagram you can see when choosing to edit a pre existing QR codes for personal/dynamic type Vcard QR code the info is updated and save into the database without returning a new QR code since it will be using the pre existing QR code, while using other QR code options the system will generate a new QR code instead and return that to the user.

### 1.3. Delete QR code sequence

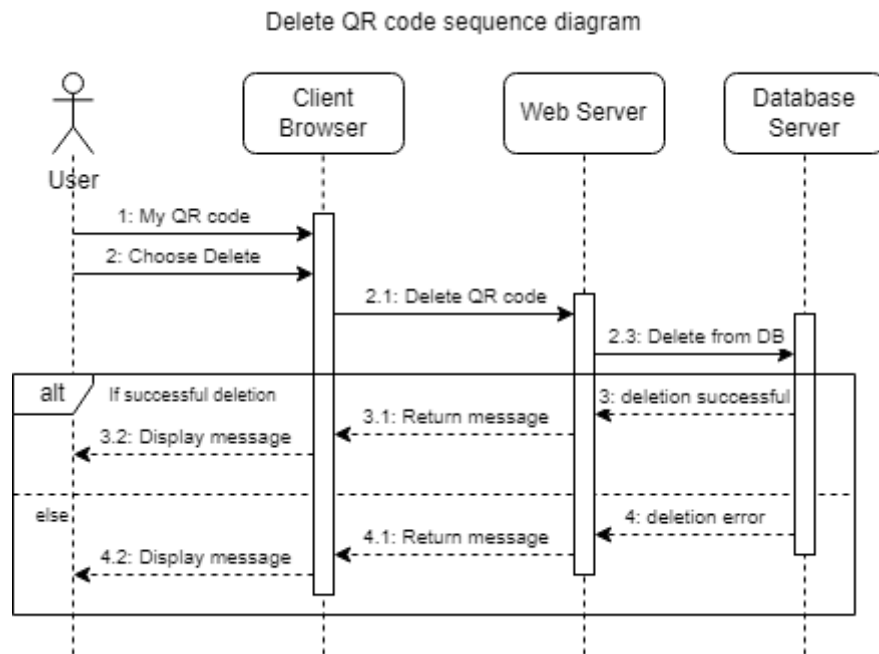


Figure 6. Delete QR code sequence diagram

When deleting a QR code the system simply tells the database to remove the QR code from the database. Then the system will receive a message whether the deletion was successful and return that message.

### 1.4. Search QR code sequence

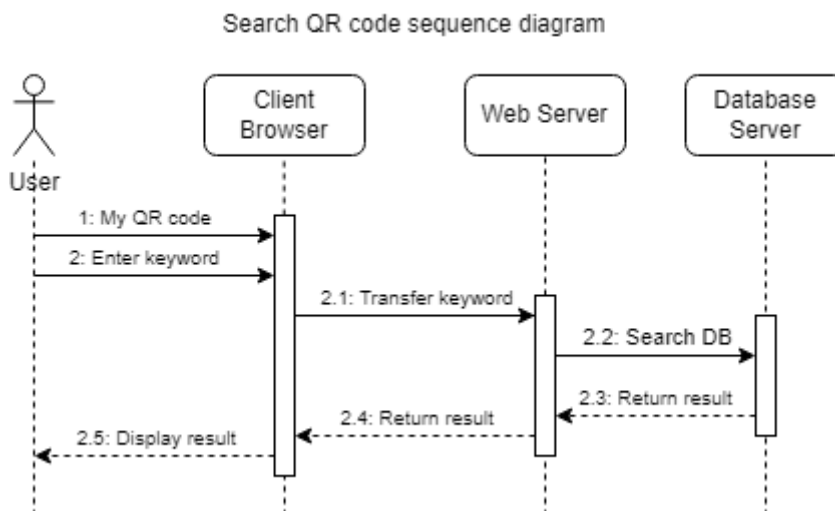


Figure 6. Search QR code sequence diagram

When searching a QR code the system will search the database for keywords regarding the QR code i.e. when searching for link QR code it will only list out link QR codes. Then the system will receive the result from the database and display the result.

## 2. QR Code generation development

In the development of QReactive, the QR code generation functionality was a pivotal component, and it was achieved by utilizing the "qrcode" library available through npm. The QR code generation process involves several key aspects:

### 2.1. Library Selection:

The team opted to use the "qrcode" library from npm due to its reliability, ease of use, and comprehensive features for QR code generation. This library can save QR code as image, auto generate optimized segments for best data compression and smallest QR Code size, providing a simple and efficient way to encode data into QR codes, aligning with the project's goal of versatility.

### 2.2. Integration with Node.js:

The "qrcode" library was integrated into the Node.js framework, allowing for the efficient generation of QR codes within the web application. Integration was done on the backend to ensure that QR codes could be dynamically generated based on user input and preferences.

### 2.3. Error Handling and Validation:

The QR code generation process incorporated robust error handling and validation mechanisms. Users were notified of any issues with the input data, ensuring that only valid and accurate information was encoded into QR codes.

### 2.4. Storage and History:

Generated QR codes were securely stored in the database, creating a history for each user. This feature allowed users to manage and retrieve QR codes they had previously generated, enhancing the practicality of the application.

### 2.5. Testing and Validation:

Extensive testing was conducted to validate the accuracy and functionality of the QR code generation process. Unit tests were performed to ensure that the "qrcode" library integration worked effectively within the application context.

### **3. Tools and Techniques**

#### **3.1. Programming Languages**

We choose to use the normal standards when it comes to programming languages HTML, CSS, JavaScript, TypeScript.

#### **3.2. IDE (Integrated Development Environment)**

We choose Visual Studio Code as the primary development environment because of its versatility.

#### **3.3. Framework**

Next.js is a React-based framework specifically designed for building web applications; it offers automatic optimization, advanced routing, and support for external libraries.

Express (Node.js Framework) is utilized for building the backend server in conjunction with Node.js since Express is a minimal and flexible Node.js framework, streamlining the development of robust and scalable web applications which enables efficient routing, middleware integration, and handling of HTTP requests, crucial for QReactive's backend functionality.

#### **3.4. Database System**

MongoDB is our database system of choice for its flexible and scalable data storage, it is a NoSQL system which makes it accommodating to the dynamic and diverse nature of the data generated by QReactive.

#### **3.5. Version Control and Collaboration**

GitHub is used to provide version control and collaboration between team members.

#### **3.6. Security**

JWT (JSON Web Tokens) and "bcrypt" is used to improve data security for user authentications. Usage: Implemented for secure user authentication. Although basics it gets the job done.

#### **3.7. API Testing Tool**

Postman is used for thorough testing of backend functionalities, including API endpoints because it provides a user-friendly interface for testing API requests.

## 4. Designs overview

### 4.1. Database design

The database design of QReactive is structured to efficiently store user account information and QR code generation history. MongoDB, a NoSQL database system, was chosen for its flexibility and scalability, accommodating the dynamic nature of the data generated by QReactive. The database schema is outlined below:

#### Account:

```
{
  "_id" : "ObjectId('AAA')",
  "email" : "JohnDoe@gmail.com",
  "password" : "abcxyz",
}
```

#### myProfile:

```
{
  "_id" : "ObjectId('AAB')",
  "account" : "ObjectId('AAA')",
  "firstName" : "John",
  "lastName" : "Doe",
  "email" : "JohnDoe@gmail.com",
  "phone" : "01234567",
  "dob" : "2004-04-01",
}
```

#### VcardQR:

```
{
  "_id" : "ObjectId('AAC')",
  "name" : "John",
  "email" : "JohnDoe@gmail.com",
  "phone" : "01234567",
  "address" : "18 Hoang Quoc Viet, Cau Giay, Hanoi",
  "website" : "https://usth.edu.vn/en",
  "company" : "USTH",
  "position" : "Manager",
  "account" : "ObjectId('AAA') ",
  "QRcode" : "iVBORw0KGgoAAAANSUhEUgAAAgwFwX6E7n",
}
```

```

"Link" : "http://localhost:5000/personalQR/scan/65ecb53e1aea",
"generateAt" : ISODate("2014-03-28T09:42:41.382Z"),
"type" : "personal",
"status" : ["public", "private"],
"DoS" : ["static", "dynamic"],
"tag" : "qr name",
}

```

#### **customQR:**

```

{
  "_id" : "ObjectId('AAD')",
  "properties" : {
    "key": "Name"
    "value": "John",
  }
  "account" : "ObjectId('AAA')",
  "QRcode" : "iVBORw0KGgoAAAANSUhEUgAAAgwFwX6E7n"
  "generateAt" : ISODate("2014-03-28T09:42:41.382Z"),
  "type" : "personalData",
  "tag": "qr name",
}

```

#### **linkQR:**

```

{
  "_id" : "ObjectId('AAE')",
  "content" : "https://www.facebook.com"
  "QRcode" : "iVBORw0KGgoAAAANSUhEUgAAAgwFwX6E7n"
  "generateAt" : ISODate("2014-03-28T09:42:41.382Z"),
  "type" : "link",
  "tag": "qr name",
}

```

#### **textQR:**

```

{
  "_id" : "ObjectId('AAF')",
  "content" : "Hello World",
  "account" : "ObjectId('AAA')",
  "QRcode" : "iVBORw0KGgoAAAANSUhEUgAAAgwFwX6E7n"
}

```



```
"generateAt" : ISODate("2014-03-28T09:42:41.382Z")
"type" : "text"
>tag": "qr name"
}
```

#### **emailQR:**

```
{
  "_id" : "ObjectId('AAG')",
  "email" : "xyz@gmail.com",
  "subject" : "Greeting",
  "content" : "Hello",
  "account" : "ObjectId('AAA')",
  "QRcode" : "iVBORw0KGgoAAAANSUhEUgAAAgwFwX6E7n"
  "generateAt" : ISODate("2014-03-28T09:42:41.382Z"),
  "type" : "email",
  "tag" : "qr name",
}
```

#### **wifiQR:**

```
{
  "_id" : "ObjectId('AAH')",
  "name" : "USTH classroom",
  "encryption" : "WPA",
  "password" : "abcxyz",
  "account" : "ObjectId('AAA')",
  "QRcode" : "iVBORw0KGgoAAAANSUhEUgAAAgwFwX6E7n"
  "generateAt" : ISODate("2014-03-28T09:42:41.382Z"),
  "type" : "wifi"
  "tag": "qr name",
}
```

## **4.2. Web design**

We wanted the UI to be “friendly” and “simple” to the user. By running a minimized version of the website this makes it easier for users to navigate and use the website, we also try to pick light colors to improve visibility and give a sense of “friendliness” to the website. The simple design also helps a lot with debugging and maintaining the website. Below are some example images.

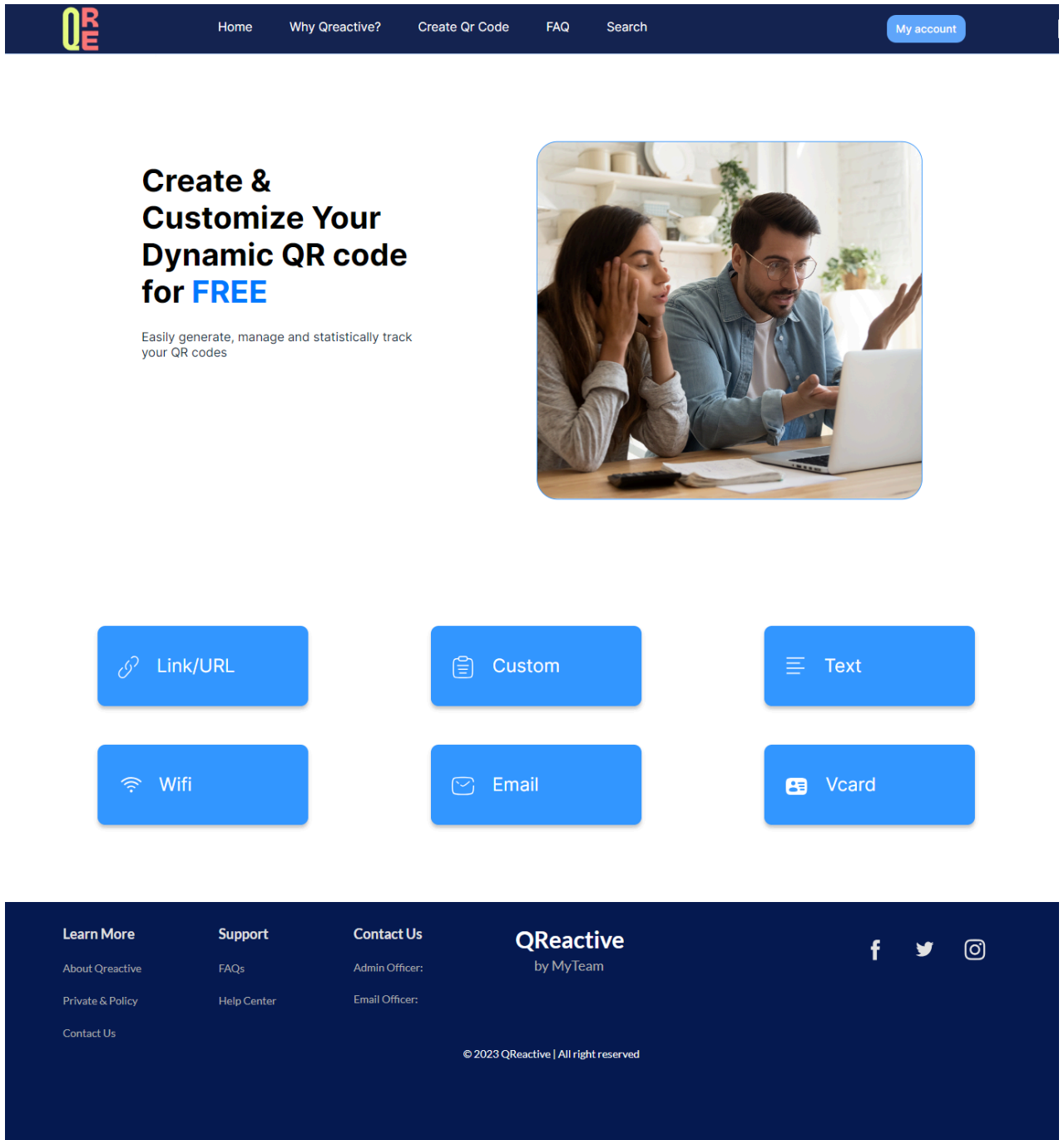


Figure 8. Home page

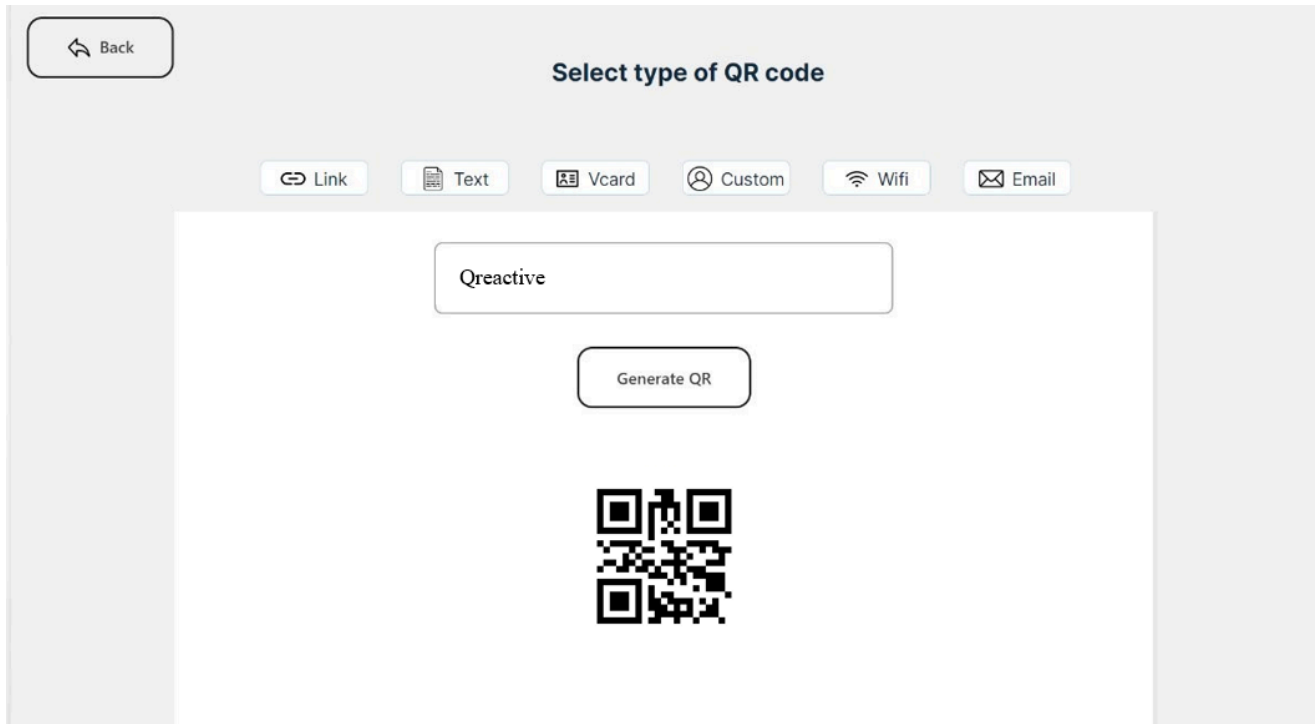


Figure 9. Text QR code generate

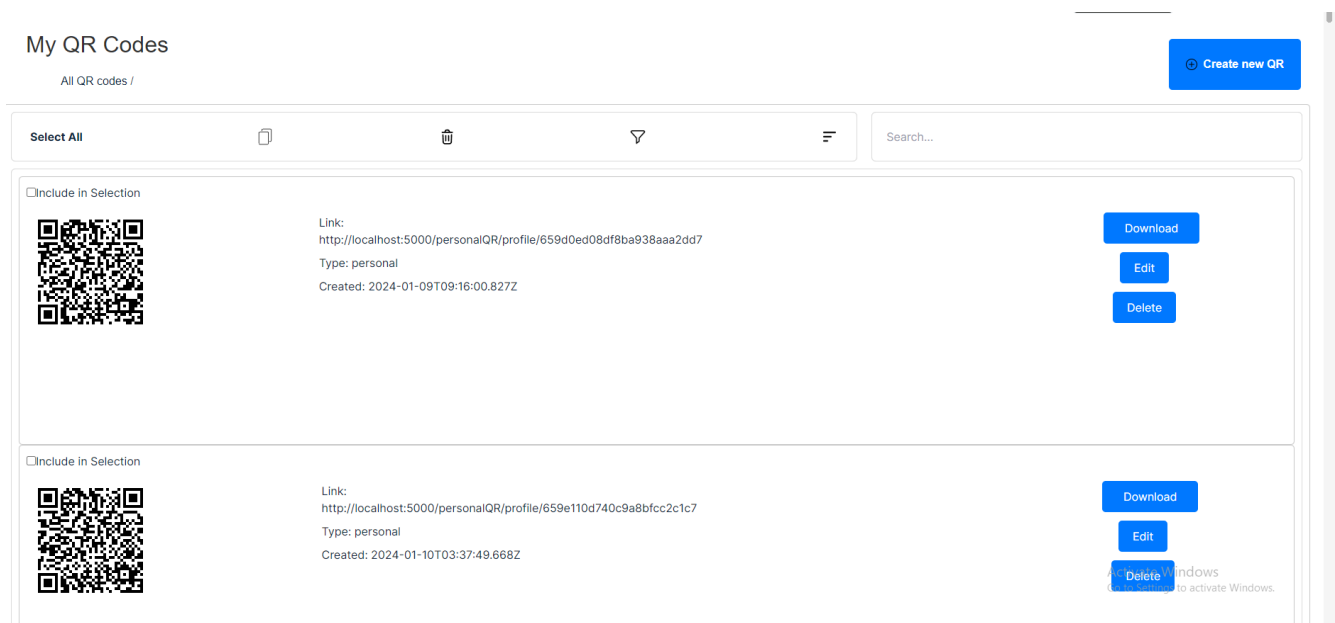


Figure 10. QR code management

**General information**

FIRST NAME | Kiết

LAST NAME | Võ Minh

EMAIL | kietvm.bi12-229@st.ussth.edu.vn

PHONE NUMBER | 0646815146341

08/04/2003

Save

**Change Password**

OLD PASSWORD | \*\*\*\*\*

NEW PASSWORD | \*\*\*\*\*

Save

*Figure 11. Personal info management*

## IV. Overall Evaluation

### 1. Outcome:

Overall we have created a user-friendly web-app that provides users with the ability to create and manage their own QR codes while using industry standard tech such as Next.js, HTML, CSS, JavaScript and TypeScript. The backend system including the database are capable of storing an incredible amount of data ensuring future expendability. We have also created the basis to expand our website to become a type of social media allowing users to find others contact info. Basic security measures, including the use of “bcrypt” hash for password storage have also been implemented.

### 2. Areas for improvement:

There is still a lot to be done before the app can move on to production. A lot of bug fixes needed to be done and a few different more types of QR codes should be added, the security is still quite barebone and for a web-app that handles so much info this is undesirable. The search Vcard function needs to be expanded and improved before it can function as a proper social media.

## **V. Project Management**

### **1. Initiation**

During the initiation phase, the team defined the project scope, objectives, and the technologies to be used. Understanding the prospect of expanding QRcode to social media allowing users to easily search for and add contact info to their devices.

### **2. Planning**

Developed a detailed project plan that included timelines, milestones, and deliverables. Created a list of requirements for the overall system and use cases. Decided on the technologies and tools to be used for website development, including Next.js, HTML, CSS, JavaScript, TypeScript, Visual Studio Code, and GitHub. Outlined the database design, material design, and web design, ensuring a cohesive and user-friendly interface. Planned the use case implementation for main and user pages.

### **3. Implementation**

Utilized Visual Studio Code as the primary Integrated Development Environment (IDE) for coding tasks. Adopted Next.js, HTML, CSS, JavaScript, and TypeScript for website development. Leveraged GitHub for version control and collaboration among team members. Designed backend code to handle QR code generation, user accounts, and database interactions. Tested backend functionalities using Postman to ensure seamless API interactions and data handling. Integrated the backend functionalities with the frontend code for a cohesive user experience.

### **4. Monitoring and control**

Conducted thorough testing of the integrated system to ensure the proper functioning of both frontend and backend components. Monitored code quality, conducting code reviews to maintain consistency and best practices. Ensured adherence to the project timeline and adjusted plans as needed to accommodate changes or challenges.

### **5. Project close**

Conducted thorough testing to identify and address any bugs or issues in the application. Ensured all features were implemented according to the requirements. Prepared documentation for end-users and future developers. Celebrated the successful completion of the QReactive project.

## VI. Reference

qrcode - npm. (n.d.). Retrieved from <https://www.npmjs.com/package/qrcode>

MongoDB - The database for giant ideas | MongoDB. (n.d.). Retrieved from <https://www.mongodb.com/>

Next.js - The React Framework. (n.d.). Retrieved from <https://nextjs.org/>

JSON Web Tokens - jwt.io. (n.d.). Retrieved from <https://jwt.io/>

bcrypt. (n.d.). Retrieved from <https://www.npmjs.com/package/bcrypt>

Lama Dev - <https://www.youtube.com/@LamaDev>

Authentication using JWT in Node.js - Viblo. (n.d.). Retrieved from <https://viblo.asia/p/authentication-trong-nodejs-su-dung-jwt-YWOZraQvKQ0>