

QR CODES FOR GOOD

By

COREY JUSTIN MINER

A Thesis Submitted to The Honors College

In Partial Fulfillment of the Bachelors degree
With Honors in

Electrical & Computer Engineering

THE UNIVERSITY OF ARIZONA

M A Y 2 0 1 9

Approved by:

Gary Redford
Department of Electrical & Computer Engineering

Abstract

This project is a payment transaction system based on QR codes. The purpose of the project is to create a system where you can donate to homeless people electronically. 70% of people today no longer carry cash, so homeless people are no longer getting the donations that they once got. Using QR codes, the homeless person can go to the public library and print a QR code, and then any person who wants to donate only needs to scan the QR code to donate. By having the money electronically, by partnering with homeless shelters or EBT program, the recipient of the donation can be restricted in what they can purchase. Not allowing them to buy alcohol or drugs can make people donating feel better about their donations as well. Overall, people are donating less and cash can be spent anywhere, but with this project, donations can be sent quickly and painlessly through a QR code.

Engineering Team

Muneeb Ahmed: Software Engineer

Muneeb worked with Corey Miner and Benjamin Wodhams on the main mobile application as well as the backend JavaScript code hosted on the Azure server

Amanda Chesin: Software Engineer

Amanda was in charge of making the initial mockups of what the application would look like, some application software design, and was chief android engineer.

Corey Miner: Software Engineer

Corey worked with Muneeb Ahmed and Benjamin Wodhams on the main mobile application as well as the backend JavaScript code hosted on the Azure server.

Sanarya Salah: Systems Engineer

Sanarya was in charge of all scheduling and systems engineering work.

Benjamin Wodhams: Software Engineer

Ben worked with Muneeb Ahmed and Corey Miner on the backend JavaScript code hosted on the Azure server and the website.

Table of Contents

	Page number
1.0 Scope	2
2.0 System Block Diagram	2
3.0 Technical Data Package	3
4.0 Acceptance Test Procedure	5
5.0 Models / Analyses	7
6.0 Acceptance Test Results	24
7.0 Final Budget	25
8.0 Lessons Learned	25

1.0 Scope

The purpose of this project is to create a payment transaction system that generates QR codes on a mobile application, or a website, with a specific amount of money decided by the receiver and can be adjusted by the payer. This project is mainly directed to two different groups of users. The first group is people without a steady income source who live based on donations and might not have a phone to access other methods of payment. In this case, they can print out the QR code through public resources and present the QR code to potential payers to scan the code. The second group is people with jobs where a big portion of what they make is based on tips, such as waiters, doormen, or valet attendants. This group of people may not be allowed to have their phones on them during work, thus they can pull up a QR code card and present it to the customer. In general, people are less likely to donate money if they don't have change, or don't have the time to stop and give money. With this service, an individual can easily donate a preset amount of money without having to go to a bank to withdraw cash, or break larger bills into smaller ones. Instead, the users of this service can simply just scan a code and donate money to the other party.

2.0 System Block Diagram

The diagram below shows the project interfaces. The user (whether it's the payer or the receiver) interacts directly with the user interface (website and mobile apps) in order to create an account, sign in (if an account already exists), generate a QR code, Pay a third party, print a QR code, customize payment methods as well as other settings properties. Once the the data is uploaded to the app, then it will connect to the server that branches to the database (Azure) where all the data will be stored and the payment processor where the transaction will be approved and sent.

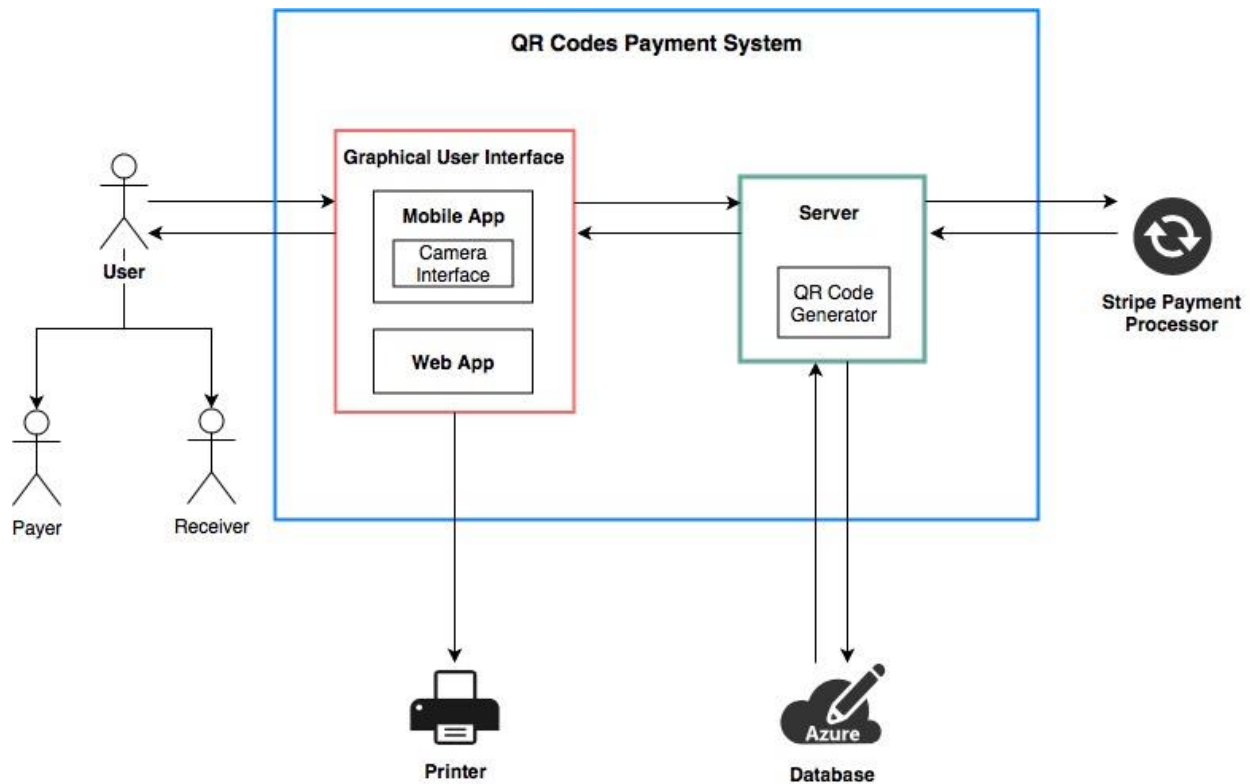


Figure 1: System Block Diagram

3.0 Technical Data Package

3.1 Front End Technical Data Package: this part includes graphical all the graphical interfaces for the the Mobile application in both platforms (iOS & Android) as well as the website interface. All the front end pages were created in React Native, a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. The website shall have the same functionality as the mobile application except few dimensional differences. For that reason, we will focus on the mobile application front end GUIs. Once the mobile app interfaces are completed, we will integrate our code for the website with some minor dimensional changes.

3.2 Back End Technical Data Package: the backend portion of this project is the main part where the functionality of the app is made. Before, in the front end, the code we wrote in React

Native was just to make the page look the way they do, here we coded the the actual payment transactions, generating unique QR codes, Database, verifying accounts, etc. For payment processing,

3.2.1 Payment processing: Stripe is the payment service that will handle all payment transactions between users. It will interact directly with the Azure virtual machine. It has its own database so that we will not have to save or store users payment information. This is extremely important for security and liability measures. All interactions between stripe and the Azure virtual machine will be encrypted as well. Users will be able to send or receive payments through stripe transactions.

3.2.2 Database: Azure database is where all user information will be stored. It must communicate with the stripe payment processor as well as any client that attempts to log on or use our app. The database will be a Azure Table Storage database that will provide the azure virtual machine with the required information to log users in and store information when users make changes and use the service. Azure Table Storage was chosen, as it is cheaper and more lightweight compared to Azure's other storage options, and also supports JSON. The information stored includes the user's account id, password hash, and financial (Stripe) tokens, but not credit card verification values. Moreover, The Azure virtual machine will run the server code connecting Stripe (the payment service), the Azure Table Storage database, and the client all together. This virtual machine will encrypt all data sent and received to/from any user's device via HTTPS. If any payment transactions must be made, the database will send information to the stripe service in order to complete the appropriate transaction. It will also send information to the Azure database to be stored. The server will be written in javascript, utilizing the Node.js runtime environment. Security is extremely important for our project. The Azure virtual machine will encrypt any requests sent or received from any user, via HTTP over TLS (HTTPS). All data stored in the Azure database will also be secured through encryption. All of the payment information will be stored in Stripe and nowhere else. Stripe has its own encryption

and security for all payment information of users. Every level of our application will be secure through encryption.

3.2.3 QR Code Generator: QR Code generator code will execute on the server when it receives an appropriate GET request from the user's device through either the website or mobile application. The server will encode the given data into a QR code and return the generated image to the user's device

4.0 Acceptance Test Procedure

This procedure outlines the acceptance tests that will be performed to verify that the app functions correctly. All testing for each part of the project will be done through unit tests with the same referenced documents, the same equipment, and the same step-by-step procedure. For that reason all tests have been put together as one series of tests.

4.1 App and Web Interface: these unit tests will test the functionality of the application code. Since the language we are using is cross platform, we are able to write one set of code and deploy it on web, IOS and android.

4.2 Camera Interface: These tests will be performed to verify that the camera interface can successfully recognize a QR code and then get the correct information from that QR code.

4.3 Payment Processor: The payment processor tests will verify that a series of different payment methods function correctly.

4.4 Server: These unit tests will test that the server can send and receive information from the database, payment processor, and user's side.

4.5 Printer: These unit tests will prove that a code can be printed from either phone or computer and then be scanned successfully.

4.6 Valid Email: these unit tests will test that user's emails are created correctly in account creation.

4.7 Load Time: these unit tests will be time out tests to test the load time with different webspeeds.

4.8 System Security: these unit tests will test that code is encrypted throughout the entire system.

The steps taken towards performing the tests is basically opening functioning application code, and unit tests then run each unit test. Below is the table of test and their decided requirements:

Test #	Test	Requirement
4.1.1	log in/create account	All tests pass
4.1.2	Submit server request	All tests pass
4.1.3	Receive server information	All tests pass
4.2.1	QR scan tests	All tests pass
4.3.1	Credit card tests	All tests pass
4.3.2	Bank account tests	All tests pass
4.4.1	Server to Stripe tests	All tests pass
4.4.2	Server to Database tests	All tests pass
4.4.3	Server to Mobile tests	All tests pass
4.4.4	Server to Web tests	All tests pass
4.5.1	Web to printer tests	All tests pass
4.6.1	Email reception tests	All tests pass
4.7.1	Load Time timeout tests	All tests pass
4.8.1	encryption tests	All tests pass

5.0 Models/Analyses

5.1 Mobile App

5.1.1 Sign-in page: This page will pop up when an unauthenticated user attempts to access privileged information. This screen allows the user to insert their email and password to sign in, if the user have already created an account, then they can press the button “Sign In” in order to move to the next page. If the user hasn’t created an account yet, they can press the

“Sign-up” button in order to go to the next page and create a new account. This page also features a help button on the left bottom corner of the screen where the user can find the most frequent questions about the app.

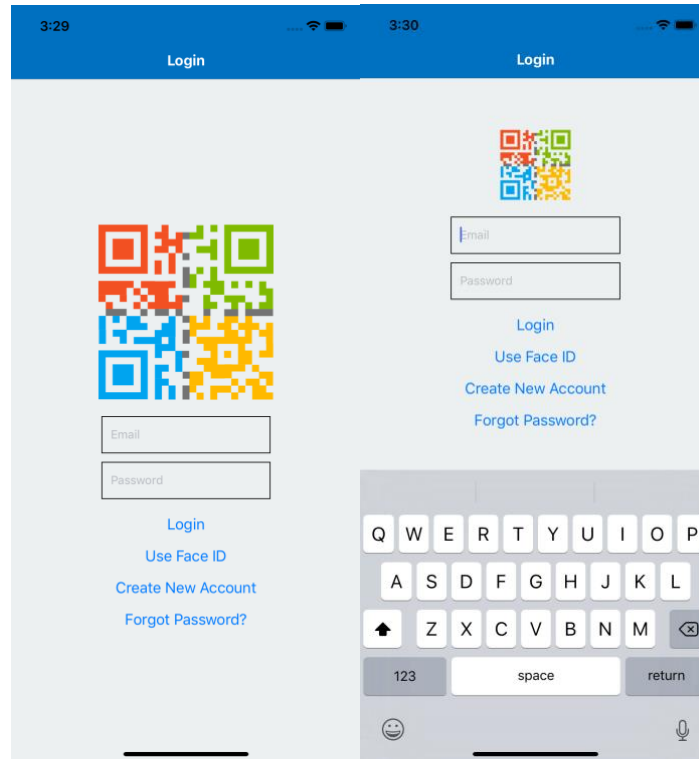


Figure 2: Mobile App Sign-in Page

5.1.2 Account Creation Page: this page will be generated when the user presses on the “Create account” button from the previous page. The user will be asked to input their first and last name, a valid E-mail address (which will be detected in the back end), and a password, which they will have to enter twice to confirm (both passwords must be identical in order to move to the next step). Then the user can press the “Sign Up” button. If all conditions are met and no fields are left blank, then an email will be sent to the user’s email with a verification code where they have to insert it in the next page in order to activate the account and be ready to use the app.

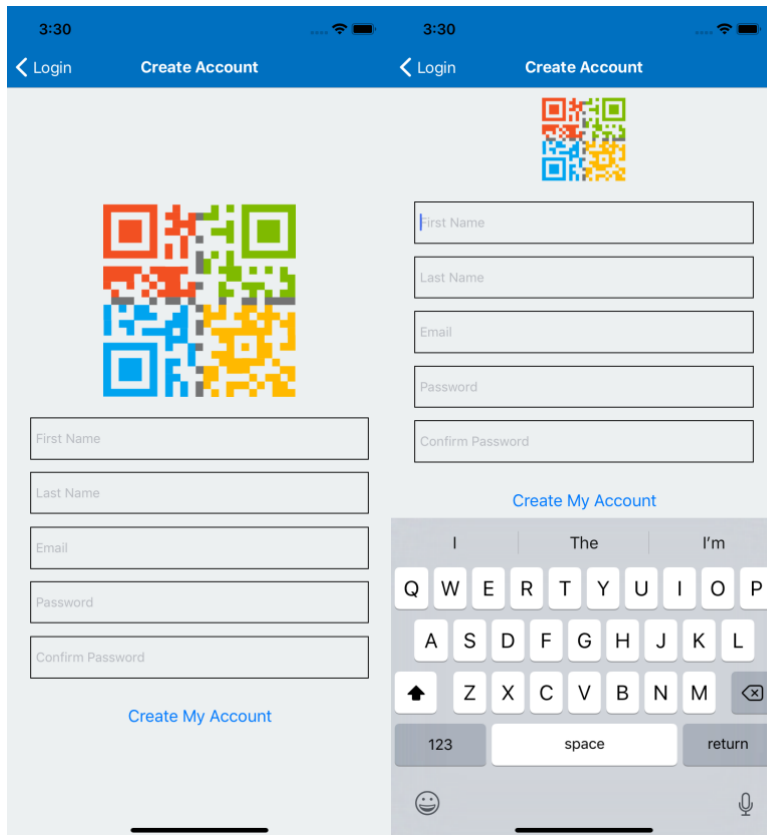


Figure 3: Mobile App Sign-Up Page

5.1.3 Homepage Main Picture: the camera will be the first thing an existing user sees once they sign in to quickly scan QR codes. Putting the QR code in focus and tapping the screen will scan the QR code and bring the user to the next screen where they can modify the payment. On the top left corner there is a hamburger icon which pull up the sidebar where the other pages of the app are easily accessible. In the top center is a link to the sign in page, which will only show if the user has not logged in yet.

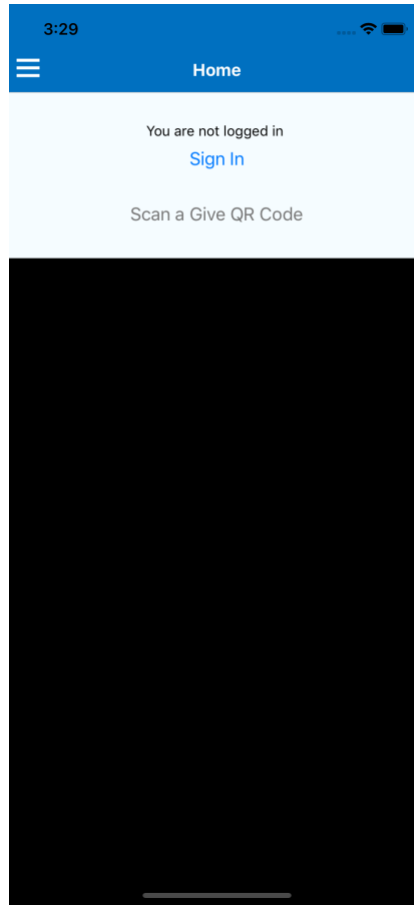


Figure 4: Mobile App HomePage

5.1.4 Homepage Sidebar: this is the Menu, which is accessible from every single page once the user logs in. The top displays your name and your email, so you are constantly reminded what email account you signed up with. The first option is “Home” which takes the user back to the Homepage Main Picture. The second option is “Create a QR Code”, which takes the user to the page where they can generate a new QR code. The third option is “Transaction History” which takes the user to the page where they can see all their former transactions, whether it’s paid or received. The fourth option is “Wallet”, which takes a user to the page where they can add or edit their payment methods. The next option is “Settings”, which directs a user to the page where they can edit the settings. The “Help” option takes a user to the frequently asked questions page. The last option is “Info”, which takes a user to the page where

all legal information. On the bottom of the menu one can see the “Log Out” option, which gives the user the option of signing out at any time

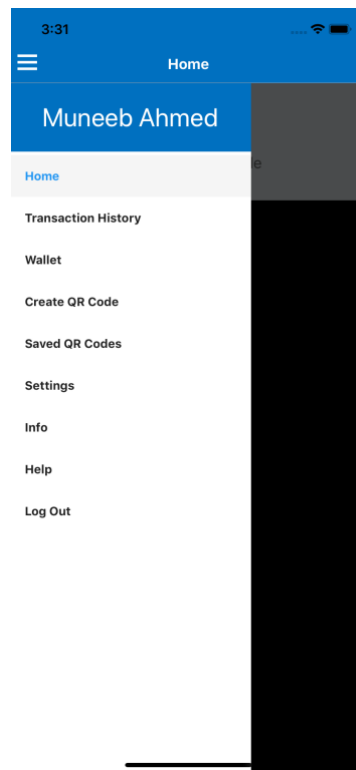


Figure 5: Mobile App HomePage Sidebar

5.1.5 Wallet Page: This is the page where a user can see all of the cards and bank accounts they have associated with their account. Each card displays the type (Visa, Mastercard, Amex, etc.) with the card holder’s name and the card number. Every checking account displays the account holder and the routing number. When the user selects the checkmark on one of their payment methods as the “Primary Payment Method”, it will be the default for every transaction. It can be changed on the payment page if necessary. Moreover, by clicking the “Add New Payment Method” button at the bottom of the page, the user can add a new payment method.

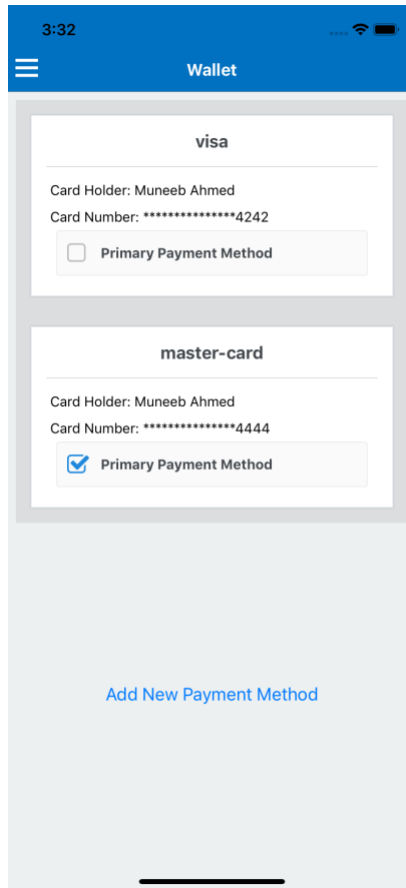


Figure 6: Mobile App Wallet Page

5.1.6 Transactions Page: this page shows all of the payments a user has received. It shows every transaction where the user received money with the amount paid, the date it was paid, and who it was from. A user can also receive anonymous payments and not know who they are from, if the sender opted to pay anonymously.

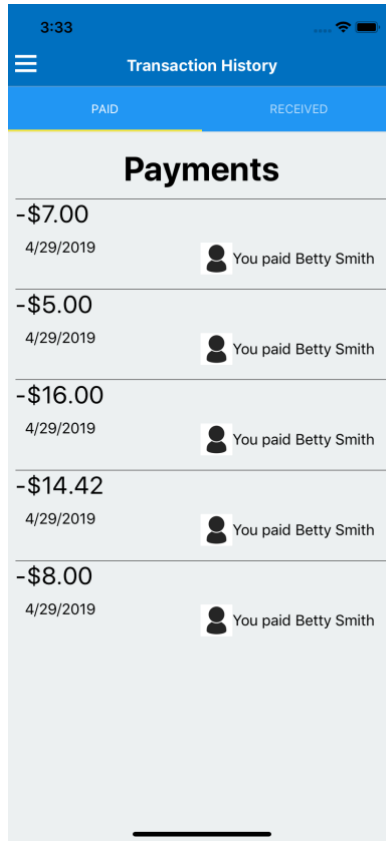


Figure 7: Mobile App Transaction Page

5.1.7 Saved QR Codes Page: this is the Saved QR Codes Page, accessible from the homepage sidebar. The user can see their QR codes that they have previously generated, as well as other important QR codes that the user can save for frequent use. By clicking on their own QR code, they can save the image to their Photos. By clicking someone else's QR code, they will be taken to the payment page above.

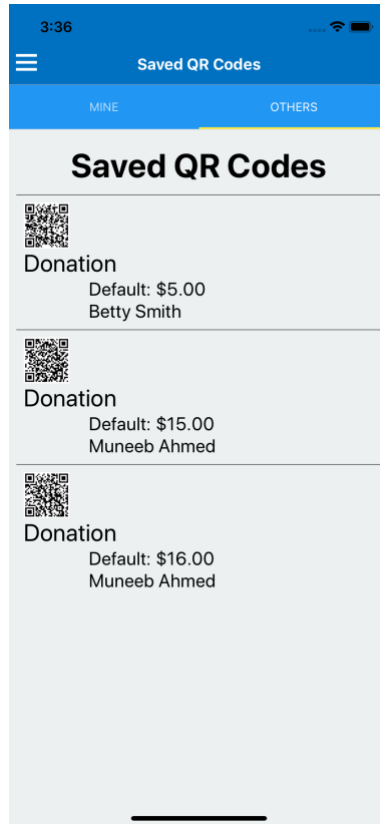


Figure 8: Mobile App Saved QR Code Page

5.1.8 Settings Page: this is the user's Settings Page. For payment, the user can see their bank account and saved credit cards, as well as adding a payment method, similar to what we have in the payment page we discussed previously. The user can see the email they signed up with and change it if need be. The user can select "Change Password..." to bring up a screen where they are prompted to enter their old password and create a new one. For this app it is an iPhone X, which has FaceID so the user can decide to enable that. For lower models of the iPhone that will say "Enable Touch ID".

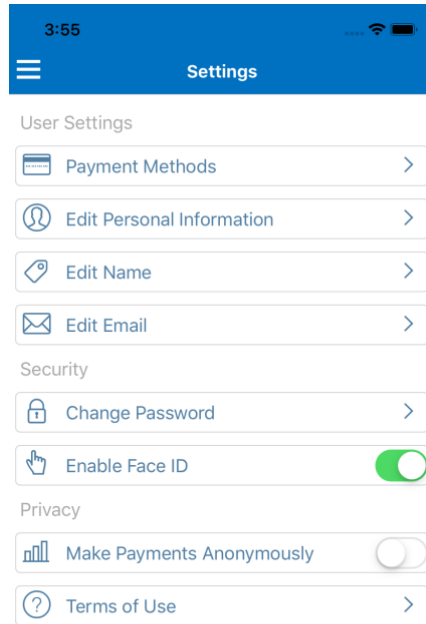


Figure 9: Mobile App Settings Page

5.1.9 Create QR Code Page: the Create QR Code page, accessible also from the homepage sidebar, generates a new QR code for a user to share to receive money. The user can determine a set amount for their QR code, and once they select “Submit”, it will show up in the center of the screen. It can be saved from there to the user’s camera roll. Closing the popup will redirect the user to their saved QR codes.

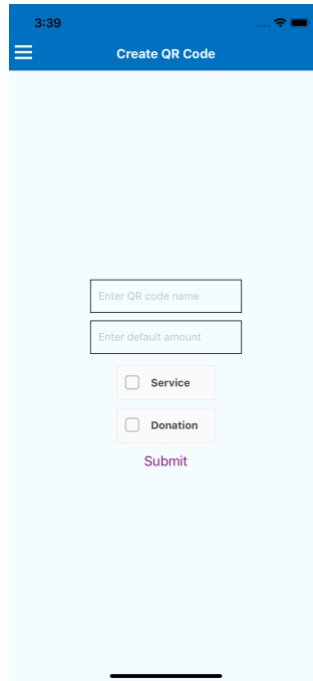


Figure 10: Create QR Code Page

5.2 Web App

The main reason behind why a website was needed for our project is so that a user (homeless person for example) can go to a public library, generate a QR code, and print it there if they don't have a smartphone. This way they can present their printed QR code to the public without having to use a smartphone. The payer, on the other hand, would still need a smartphone in order to scan the QR code.

Functionality and design of the website is very similar to our mobile app, the only main difference is you can't scan the QR codes with the website. Otherwise, you can create an account, add payment methods, generate QR codes, adjust settings, and view the terms and conditions.

5.2.1 Home Page : This page will show up when you access the root directory of the website (<https://www.qrcodes4good.com>). If you are currently signed in when you access this

webpage, the website will look a little bit different than a user that is signed out. If you are not signed in, you have the ability to create a new account or sign in to your existing account.

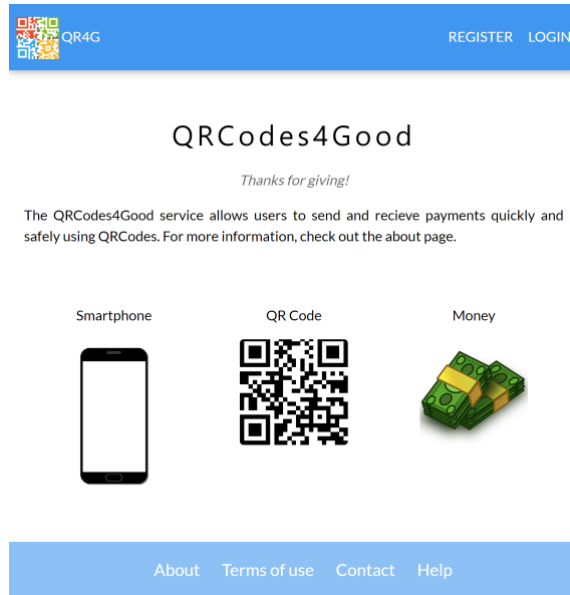


Figure 11: Home page (signed out)

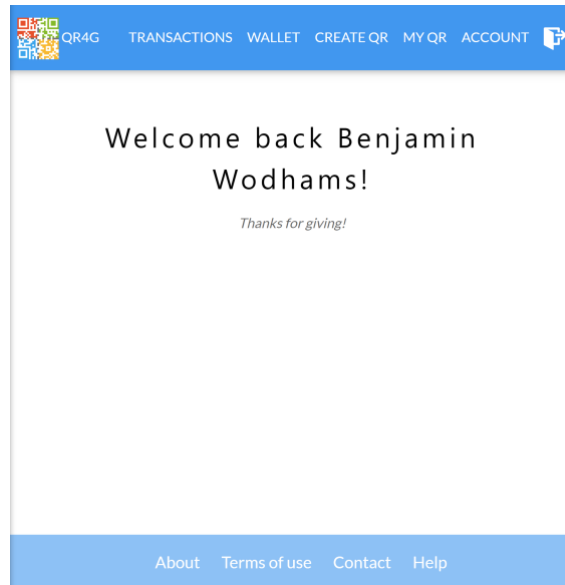


Figure 12: Home page (signed in)

5.2.2 Login Page : This page will show up when you press the “login” button on any page on the website that has the button, and you’ll be redirected to the login page (<https://qrcodes4good.com/login.html>). Once on this page, you may login to your account. If you

have forgotten your password, you may also reset your password on this page. Upon login, a cookie is stored in your browser, which allows your session to be maintained even if you close the tab. If the cookie hasn't expired yet (10 minute expiration for security), you don't have to login again unless you manually logout by pressing the logout button.

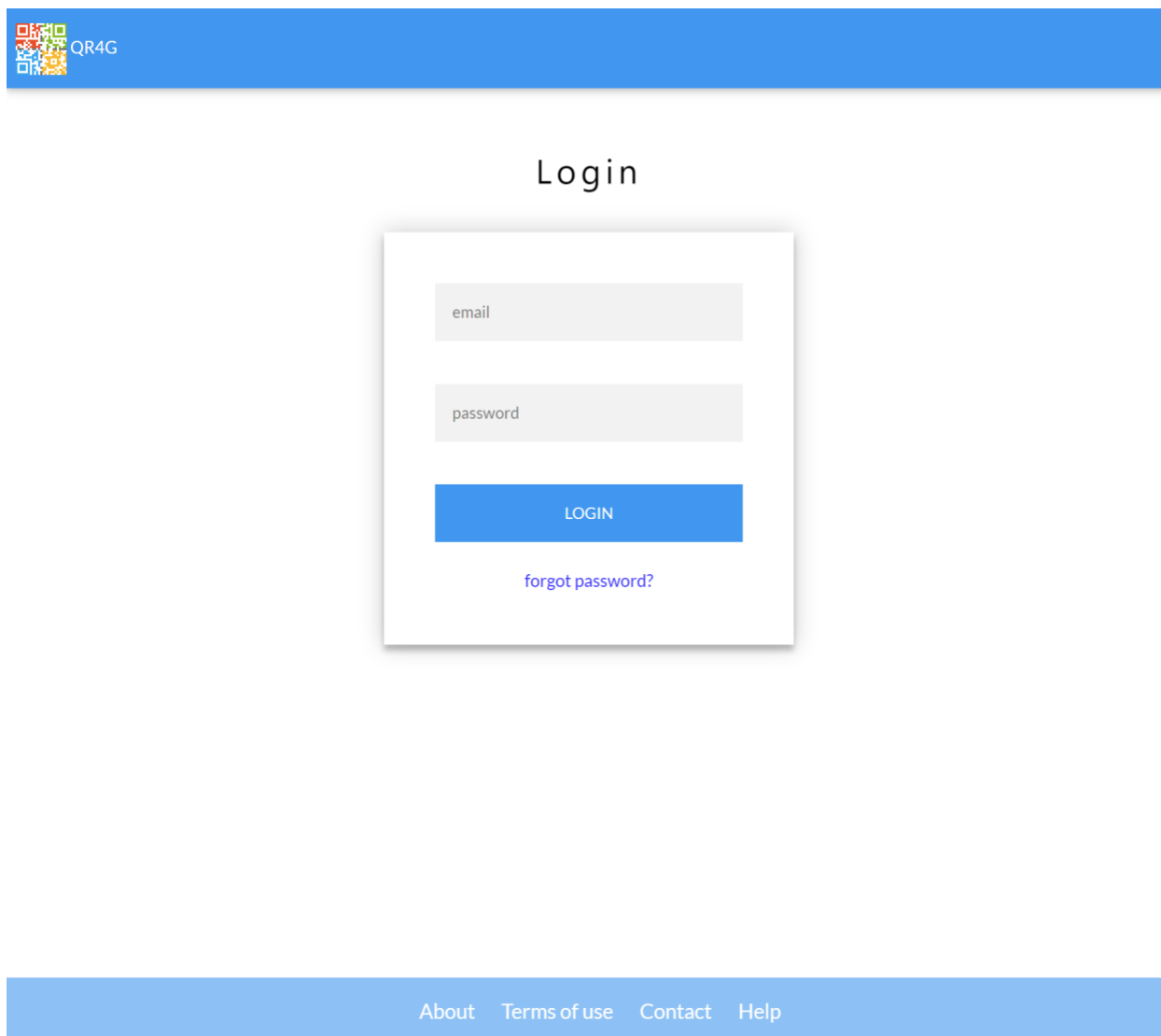


Figure 13: Login page

5.2.3 Account Creation Page : This page will show up when you press the “register” button on any page on the website that has the button, and you’ll be redirected to the account registration page (<https://qrcodes4good.com/register.html>). Once on this page, you may create your account. Once you submit your information, it will be checked for validity of the E-mail

address, as well as for security of the password. If the inputted information is invalid, the form will let you know, and will let you know what needs to be fixed to continue with your account registration.

QR4G

Register

name (first last)

email

password

confirm password

REGISTER

About Terms of use Contact Help

Figure 14: Account creation page

5.2.4 Wallet Page : This page will show up when you press the “wallet” button on any page on the website that has the button, and you’ll be redirected to the wallet page (<https://qrcodes4good.com/user/wallet.html>). Once on this page, you may change your default (primary) payment method, add a new credit card, add a new bank account, and delete credit

cards or bank accounts from your account. The card that is highlighted light purple is your primary payment method. Clicking on any other card or bank will change your primary payment method to the selected method.

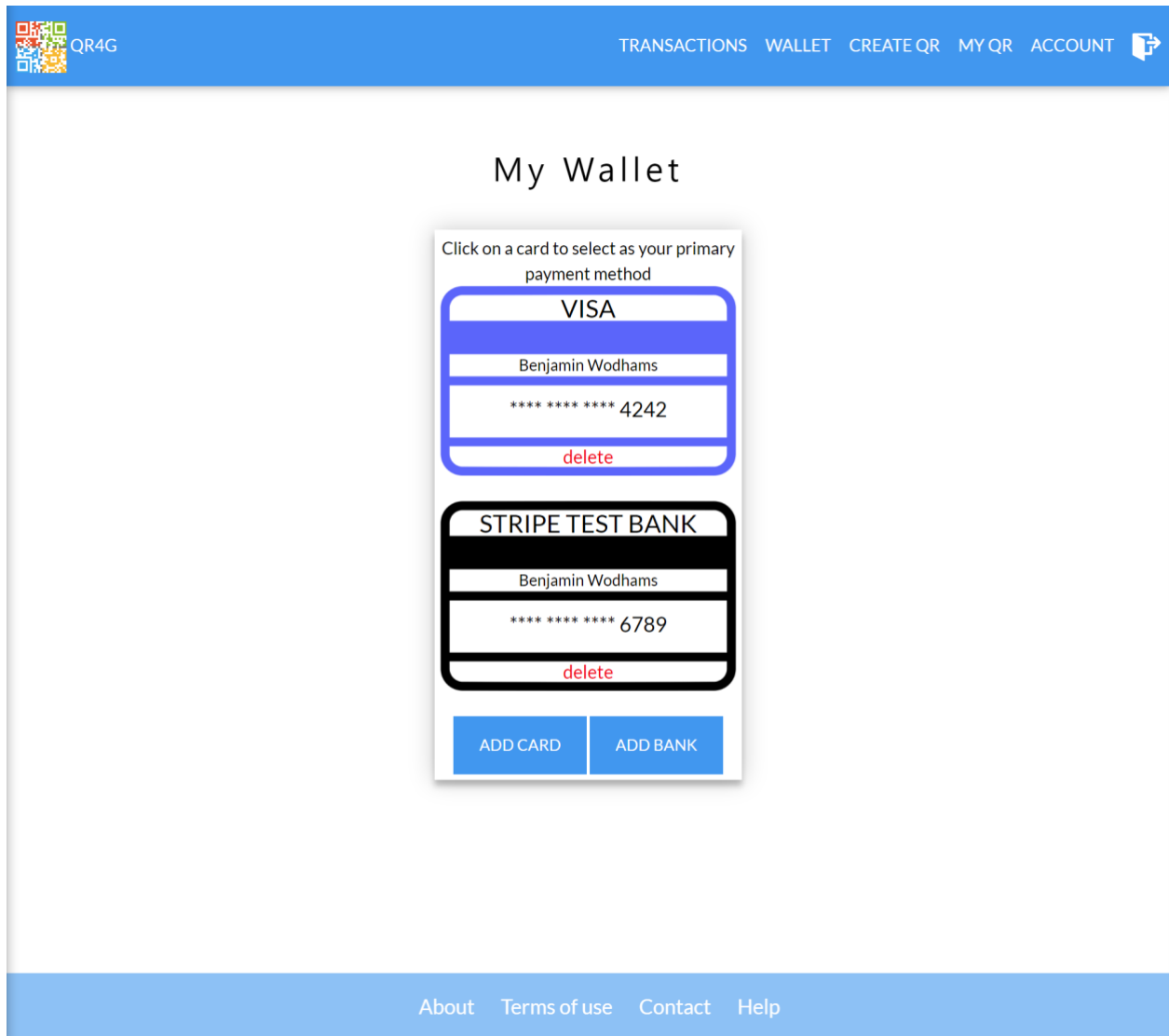


Figure 15: Wallet page

5.2.5 Transactions Page : This page will show up when you press the “transactions” button on any page on the website that has the button, and you’ll be redirected to the transactions history page (<https://qrcodes4good.com/user/transaction/history.html>). Once on this page, you will see every payment you’ve made as well as all payments that you’ve received. Each payment will include the date it was made, the amount, and the person who sent the money to you, or who you sent the money to. If any of these payments are made anonymously, the name will display as anonymous.

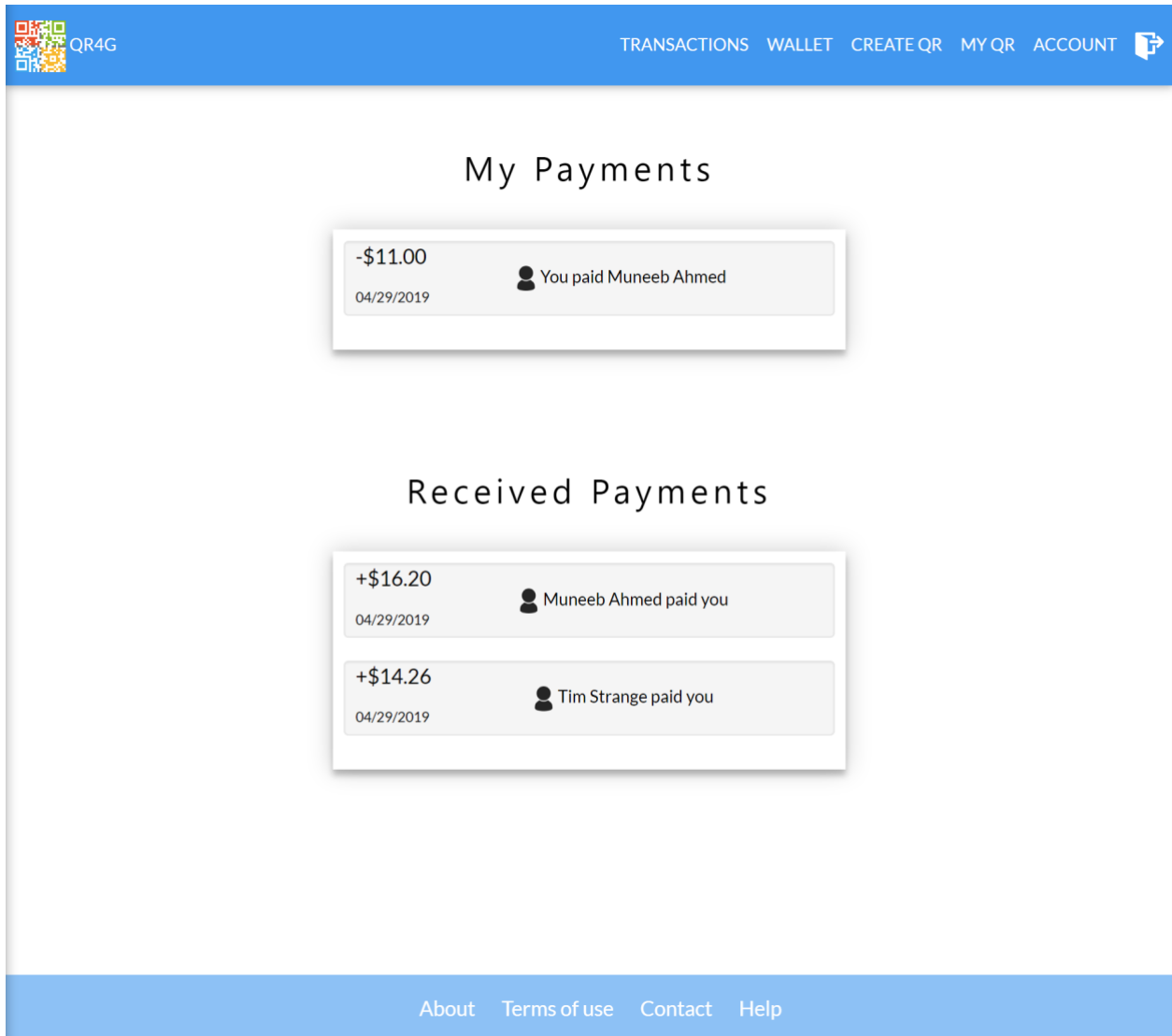


Figure 16: Transactions page

5.2.6 QR Codes Page : This page will show up when you press the “my QR” button on any page on the website that has the button, and you’ll be redirected to the QR codes page (<https://qrcodes4good.com/user/qrcode/saved.html>). Once on this page, you will see all your currently active QR codes, as well as all the QR codes that you have saved. Each QR that you have created and haven’t deleted will display the name you gave it, and you can click on the code to view more details about it, delete it or print it. The same can be done for each of your saved QR codes.



Figure 17: QR codes page

5.2.7 Account Page : This page will show up when you press the “account” button on any page on the website that has the button, and you’ll be redirected to the account page (<https://qrcodes4good.com/user/account.html>). Once on this page, you will see all your current info. You may change your name and/or email by clicking on the “edit” button next to the appropriate field. You may also change your password by clicking on the “change password?” button. Every change to your account requires you to enter in your current password. Submitting the form checks all entered info for validity just as it did when you registered for an account.

QR4G

TRANSACTIONS WALLET CREATE QR MY QR ACCOUNT

My Account

Benjamin Wodhams edit

bpwodhams@gmail.com edit

change password?

current password

SUBMIT

About Terms of use Contact Help

Figure 18: Account page

5.2.8 Create QR Page : This page will show up when you press the “create QR” button on any page on the website that has the button, and you’ll be redirected to the create QR page (<https://qrcodes4good.com/user/qrcode/create.html>). Once on this page, you will be able to create a new QR code. If creating a QR is successful, you’ll be redirected to the “my QR” page, where you will see your newly created QR code. If it’s unsuccessful, the field will show you the error.

QR4G

TRANSACTIONS WALLET CREATE QR MY QR ACCOUNT

Create A QR Code

enter QR code name

enter default amount

Service Donation

SUBMIT

About Terms of use Contact Help

Figure 19: Create QR page

6.0 Acceptance Test Results

Each of the requirements detailed in section 4 were tested either on mobile smartphones, web browsers, or both. Success of all relevant tests counted as one successful test for that device. Each device was tested ten times.

6.1 Mobile Application (iOS) Tests

iPhone 6S:	10/10 passed
iPhone 7:	10/10 passed
iPhone 7+:	10/10 passed
iPhone X:	10/10 passed
iPhone XR:	10/10 passed
iPhone XS:	10/10 passed

6.2 Mobile Application (Android) Tests

Samsung Galaxy J7:	10/10 passed
Samsung Galaxy S9:	10/10 passed
Google Pixel 2:	10/10 passed
Google Pixel 3:	10/10 passed
HTC One:	10/10 passed

6.3 Web Browser Tests

Google Chrome:	10/10 passed
Mozilla Firefox:	10/10 passed
Apple Safari:	10/10 passed
Microsoft Edge:	10/10 passed

7.0 Final Budget

Items	Quantity	Cost Each	Total Cost
Team Shirts	5	\$30	\$150
Presentation Display	1	\$90	\$90
Android Smartphones	6	\$220	\$1320
iPhones	2	\$500	\$1000
Television	1	\$300	\$300
Contingent Cost	1	\$1,000	\$1,000

Table 1: Final Budget Table

8.0 Lessons Learned

Teamwork was a very important lesson that all of our team members learned and demonstrated through the process of working on this graduation project. Communication between the team members was easily established. The git version control system was utilized during software development, which allowed members to work on separate parts of the code base simultaneously, and seamlessly merge their contributions. The team had weekly sprints, assigning each member a task for the week and keeping track of its status using Github's Projects flow. Each member of the team ensured that the others were held accountable for their assignments, and would articulate if any member was falling behind or blocking another's progress.