**E-money**

From: Ramya Radhakrishnakumar, Sushant Sharma and Vinokkumar Uthayakumar  
Discipline: Computer Engineering Technology  
Date:

# Declaration of Joint Authorship

We acknowledge that this report to be assessed is a group work by Ramya Radhakrishnakumar, Sushant Sharma and Vinokkumar Uthayakumar. The entire report and the project itself is prepared with the group’s consent and has almost our own ideas and words. All the other references and ideas invoked in this report are confirmed and are provided in the bibliography session. Ramya Radhakrishnakumar worked on the Hardware and the Technical aspect, Sushant Sharma handled the Database connectivity and Vinokkumar Uthayakumar on the other hand focused on developing the mobile application and the establishing connectivity between the sensors.

# Approved Proposal

## Executive Summary

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a database as well as to a mobile device application. The internet connected hardware will include a custom PCB with the following sensors and actuators «SensorsEffectors\_choices». The database will store «The\_database\_will\_store». The mobile device functionality will include «The\_mobile\_device\_functionality\_will\_inc» and will be further detailed in the mobile application proposal. I will be collaborating with the following company/department «I\_will\_be\_collaborating\_with\_the\_followi». In the winter semester I plan to form a group with the following students, who are also building similar hardware this term and working on the mobile application with me «My\_group\_in\_the\_winter\_semester\_will\_inc». The hardware will be completed in CENG 317 Hardware Production Techniques independently and the application will be completed in CENG 319 Software Project. These will be integrated together in the subsequent term in CENG 355 Computer Systems Project as a member of a 2 or 3 student group.

## Background

The problem solved by this project is «M\_50\_word\_problem\_statement». A bit of background about this topic is «M\_100\_words\_of\_background».

Existing products on the market include [1]. I have searched for prior art via Humber’s IEEE subscription selecting “My Subscribed Content” [2] and have found and read [3] which provides insight into similar efforts.

In the Computer Engineering Technology program, we have learned about the following topics from the respective relevant courses:

* Java Docs from CENG 212 Programming Techniques In Java,
* Construction of circuits from CENG 215 Digital And Interfacing Systems,
* Rapid application development and Gantt charts from CENG 216 Intro to Software Engineering,
* Micro computing from CENG 252 Embedded Systems,
* SQL from CENG 254 Database With Java,
* Web access of databases from CENG 256 Internet Scripting; and,
* Wireless protocols such as 802.11 from TECH152 Telecom Networks.

This knowledge and skill set will enable me to build the subsystems and integrate them together as my capstone project.

## Methodology

This proposal is assigned in the first week of class and is due at the beginning of class in the second week of the fall semester. My coursework will focus on the first two of the 3 phases of this project:  
 Phase 1 Hardware build.  
 Phase 2 System integration.  
 Phase 3 Demonstration to future employers.

*Phase 1 Hardware build*

The hardware build will be completed in the fall term. It will fit within the CENG Project maximum dimensions of 12 13/16" x 6" x 2 7/8" (32.5cm x 15.25cm x 7.25cm) which represents the space below the tray in the parts kit. The highest AC voltage that will be used is 16Vrms from a wall adaptor from which +/- 15V or as high as 45 VDC can be obtained. Maximum power consumption will be 20 Watts.

*Phase 2 System integration*

The system integration will be completed in the fall term.

*Phase 3 Demonstration to future employers*

This project will showcase the knowledge and skills that I have learned to potential employers.

The brief description below provides rough effort and non-labour estimates respectively for each phase. A Gantt chart will be added by week 3 to provide more project schedule details and a more complete budget will be added by week 4. It is important to start tasks as soon as possible to be able to meet deadlines.

«Brief\_description\_of\_planned\_purchases»

## Concluding remarks

This proposal presents a plan for providing an IoT solution for «Solution\_description». This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects such as the initiative described by [3]. I request approval of this project.

# 

# Abstract

The peculiarity of the mobile application is that one can transfer, pay and also manage their bank account all in one. It is something like a banking application but has more features to it such as one does not have the need to go to a bank to transfer money to someone far away also the user can pay merchandised bills using this application. E-money application needs sufficient amount of memory and access to the cellular data/Wi-Fi. The ideology of this project is to save time, carry transactions and merchandize bill payments with ease. Also this application can prevent investing money for purchasing credit cards. This application is available for android mobile OS only for time being.

Table of Contents

[Declaration of Joint Authorship iii](#_Toc512108225)

[Approved Proposal iv](#_Toc512108226)

[Executive Summary iv](#_Toc512108227)

[Background v](#_Toc512108228)

[Methodology v](#_Toc512108229)

[Concluding remarks vii](#_Toc512108230)

[Abstract ix](#_Toc512108231)

[1. Introduction 13](#_Toc512108232)

[2. Project Description 14](#_Toc512108233)

[2.1 Problem 14](#_Toc512108234)

[2.2 Rationale Behind Project 14](#_Toc512108235)

[2.3 Project scope 14](#_Toc512108236)

[2.4 Software Requirement Specifications 15](#_Toc512108237)

[2.4.1 Purpose 15](#_Toc512108238)

[2.4.2 Product Perspective 16](#_Toc512108239)

[2.4.3 Product Functions 17](#_Toc512108240)

[2.4.4 User Classes and Characteristics 17](#_Toc512108241)

[2.4.5 Operating Environment 18](#_Toc512108242)

[2.4.6 Design and Implementation Constraints 18](#_Toc512108243)

[3 External Interface Requirements 19](#_Toc512108244)

[3.1 Hardware Interfaces 19](#_Toc512108245)

[3.2 Software Interfaces 19](#_Toc512108246)

[4. Mobile Application Features 19](#_Toc512108247)

[4.1 Case Scenarios 19](#_Toc512108248)

[4.2 Database 20](#_Toc512108249)

[Database 1: 21](#_Toc512108250)

[Database 2: (yet to be modified) 22](#_Toc512108251)

[4.3 Mobile Application 22](#_Toc512108252)

[5. Project Overview 23](#_Toc512108253)

[5.1 Fingerprint Sensor 23](#_Toc512108254)

[5.1.1 Introduction 23](#_Toc512108255)

[5.1.2 Invoice/Bill: 24](#_Toc512108256)

[5.1.3 Budget 25](#_Toc512108257)

[5.1.4 Time Commitment 25](#_Toc512108258)

[5.1.5 Mechanical Assembly of my Fingerprint sensor 26](#_Toc512108259)

[5.1.6 Power up 27](#_Toc512108260)

[5.1.7 Installation of the Raspberry Pi Fingerprint Library 27](#_Toc512108261)

[5.1.8 Unit Testing 28](#_Toc512108262)

[5.1.9 Production testing 28](#_Toc512108263)

[5.2 QR Scanner 29](#_Toc512108264)

[5.2.1 Introduction 29](#_Toc512108265)

[5.2.2 Bill of Materials Budget 29](#_Toc512108266)

[5.2.3 Time Commitment 30](#_Toc512108267)

[5.2.4 Mechanical Assembly 30](#_Toc512108268)

[5.2.5 Unit Testing 31](#_Toc512108269)

[5.2.6 Production Testing 31](#_Toc512108270)

[6. Problems Encountered 31](#_Toc512108271)

[6.1 Data Communication 31](#_Toc512108272)

[6.2 LCD Display 31](#_Toc512108273)

[6.3 Mobile Application 32](#_Toc512108274)

[6.4 QR code email 32](#_Toc512108275)

[7. Progress Reports 32](#_Toc512108276)

[7.1 Report 1 32](#_Toc512108277)

[7.2 Report 2 35](#_Toc512108278)

[7.3 Report 3 37](#_Toc512108279)

[8. Conclusions 39](#_Toc512108280)

[9. Recommendations 39](#_Toc512108281)

[10. Technical References/Bibliography 39](#_Toc512108282)

[11. Appendices 40](#_Toc512108283)

# Introduction

A significant problem being faced by customers these days is that, every customer has to go to the bank, wait in line to be helped and then consult our needs with the representative to perform his/her banking needs such as monitoring the bank account, transfer money, pay bills, security and credit card fraud. The e-money application can overcome these problems.

To make this application work, it requires certain information such as Acc.no, name and password to be stored in the database which will then be accessed via both a hardware and Smartphone application, providing on-the-go banking experience. This application is convenient and efficient keeping user friendliness as a key to work with the application.

Since our mobile application needs a Wi-Fi or cellular data to send an email to the recipient, the whole set-up should be placed in an environment that supports either Wi-Fi or cellular data.

# 

# Project Description

### Problem

The problem encountered is, the user has to go to a bank to meet is daily banking needs such as transfer money, managing bank account and also pay merchandiser bills. The most current existing mobile applications do not have all the above features and it seems as though the user need to download more than one mobile application. This intern leads to large memory consumption and security issues. There are few banking mobile applications but these have few limitations. One of the major problem is that if the user enter the wrong account number followed by the wrong routing number it will cause the money to go into another account. Also the user needs to double check the amount the user intends to send.

## Rationale Behind Project

In order to solve the above mentioned problems, the team came up with the idea of E-money and worked towards this project. It is simply an inexpensive, user friendly, very specific, sophisticated and realistic mobile application being named the E-money. From the very name the mobile application carries all its functions using the internet. This mobile application will be the key to security for your banking information and also it is free of charge. It helps in transferring, paying and also managing accounts from one place.

## Project scope

E-Money is so beneficial since the users need to have to waste time going into a physical bank and wait for a teller to help them out to transfer money. This can also be out stretched and used to pay the application user’s mortgage or car loan. This is much quicker as a transfer processed by the bank takes a minimum of 2 business days to transfer any amount to any account. It is something like a direct deposit, here the user need not be physically present in the bank to perform the deposit. The security is taken over by the fingerprint reader. This simply accesses the bank database and matches the finger print to the email ID along with the card number and then pulls out the banking information. By doing so, there wont be any miscellaneous handle of banking data’s and only the authorised user can only handle the information.

Since the intention is to produce an user-friendly and inexpensive mobile application, some of the high level security features and also banking information has been voided. Like, the user can register by using the email ID and the intended finger print. But the user need not provide the card number of the card that is issued by the bank. Also the user need not prove other personal details such as photo ID, home address, contact number, etc. . Likewise any user can register into this application regardless of having an account in that particular bank, since that requires a well maintained banking database which keeps updating as soon as bank card is issued or expired. It also provides access to the QR code without the need to enter a passcode to open the file sent via email to transfer money to the recipient.

## 2.4 Software Requirement Specifications

## 2.4.1 Purpose

The hardware products which are going to be included in our project are fingerprint sensor and QR code scanner. The software products are firebase and android studios. The whole system will be built upon the basis of these products. This system can be further divided in two subsystems such as hardware and software aspects. The purpose for this project is to decrease the use of cash or credit/debit cards and use the money generated via QR code. The reason for using QR code is that we don’t have to worry about losing cash or even if we forget our wallet at home we can use our cell phone to pay with QR code.

## 2.4.2 Product Perspective

This product is basically a replacement of the existing banking applications and apple pay/android pay. This is built keeping security as the primary key. The application is designed in a way that the user can perform or manage transactions as well as pay their bills all in one. The following schematic gives a clear understanding:

**Case 1 & 2:**

**E-money application**

* Transfer money
* Pay using QR scanner

DB

Generate the code and let the merchandiser scan it

Generate QR code and send the code through an email to the recipient

Payment successful ☺

QR scanner scans the code and deposits the money

**Case 3:**

Account Balance: $3250.75

Credit: $200

Enter Acc.no: 1234567890

Scan

Finger Print sensor

## Product Functions

\* The Code scanner is used to scan the generated code.

\* Mobile application is used to generate the QR code.

\* The Fingerprint sensor is used for security purpose.

## 2.4.4 User Classes and Characteristics

Almost 90% of world’s population has their day starting with an android smart phone. This project is intended for people having access to their bank account and smartphone. In this era everything is possible via a smartphone for instance, one can order food, taxi, etc. using various android apps. According to e-money the use may vary from an employer to a merchandiser. This application is user friendly for both and is designed in such a way that both has access to all facilities provided by the app itself. People use online banking since one can manage monthly statements and perform transactions with ease. Whereas the merchandiser has to create a default account so that the payment made to them by the customer gets deposited in it.

## 2.4.5 Operating Environment

The environment such as weather, location etc., is not an asset for the e-money. The whole hardware is not legitimately big and hence it is easy to carry around in the parts kit itself. The application is intended for android operating system and hence it is not available for other mobile operating systems. The application collides with the fingerprint sensor to give permissions to access the user’s banking account. The barcode scanner plays a part along with the software application to perform transactions. This is how it is been organized to make the hardware and software work together.

## 2.4.6 Design and Implementation Constraints

The database incorporated in our product contains bank account info such as the ten-digit account number, account types like chequing account and savings account. The database will also include information about user login which will have the username as the account/card number, the password field as well as fingerprint for enhanced security. If the user does not have an account then they can register and make an account in order to login. For registering the user will be asked to enter their first name, last name, email id, cell phone number, account number/username, fingerprint and password. All this information will the stored in database and will be needed in order to login.

# 3 External Interface Requirements

## 3.1 Hardware Interfaces

The hardware interfaces needed for connecting our components differ from sensor to sensor. QR code scanner is going to be connected using a USB interface with the raspberry pi. The fingerprint scanner is connected through the TTL to USB convertor with the pi.

## 3.2 Software Interfaces

The database incorporated is firebase and the operating system used is Android studios (lollipop 22). The mobile application was conveniently developed with the help of Android studios which uses tools and libraries from the internet. The services needed are an internet connection, access to email protocol and converting QR code to image file and finger print sensor. Also some python code was incorporated to showcase a presentable GUI which will ease user interface.

# 4. Mobile Application Features

## 4.1 Case Scenarios

**Case 1: e-mail Transfer:**

In the user interface there is an option called send. Once send is selected it prompt the user to enter the amount to which we want to transfer. This then generated the QR code with the Acc.no and amount. This is packaged into a unique number. This is used only for transferring.

Now the recipient who receives the email with the unique number will be able to deposit the money if he takes the barcode to the bank. The bank who has the barcode scanner scans the code and deposits the money to the recipient’s account.

**Case 2: Payment:**

In this case the user will have access to generate the QR code for the amount he owes the merchandiser. This is packaged into the unique number (varies from the 1st case). This code is scanned by the merchandiser. Now the amount gets deposited directly to the merchandiser’s default account.

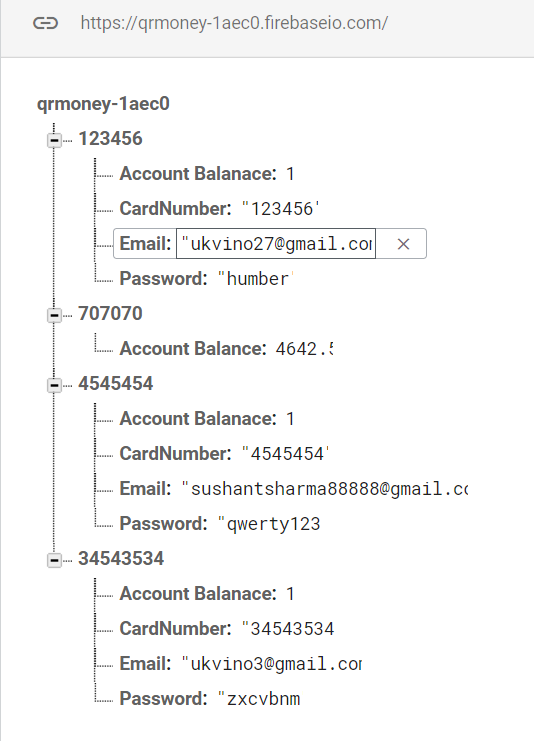
**Case 3: Account Detail:**

Now if the user wants to check the Account information he has to enter his Acc.no and scan the fingerprint associated with the bank account. The fingerprint is basically used for security purposes.

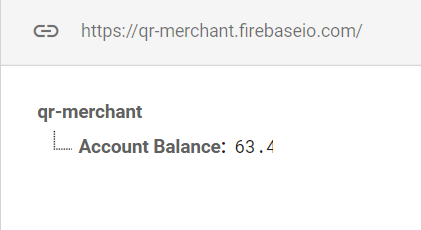
### 4.2 Database

The parent database is controlled by the bank which has details about the users such as name, card number, account balance, transaction history along with the fingerprint enrolled with the application. Also we have incorporated two separate child database for the need of the mobile application. One database is for the mobile application user where the login credentials such as Acc. Balance, card number, email, password and fingerprint of that particular user is used to enroll or register with the application. Only if the information matches it allows to access the account. If it does not match it won’t allow permission to access the account. The second database contains the merchant (Tim-Hortons, Subway. etc) database. This is like a merchant’s account where information such as Acc. Balance of the merchant is available. Along with the account balance there is some additional information such as payment details of the customer and also the time stamp of when and which location the payment was performed. The time stamp is implicated in both the child process because it can be easy to check any miscellaneous payment or security issues

### Database 1:

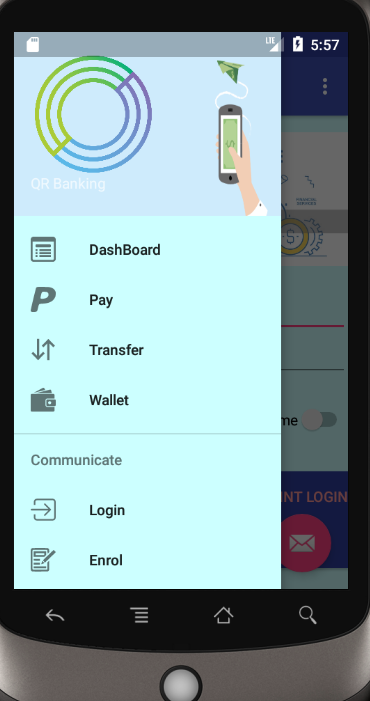


### Database 2: (yet to be modified)



### 4.3 Mobile Application

In the application we added a navigation bar that uses fragments to switch between each layout between screens. Some of the methods/functions added are overview, pay, transfer, login and enroll. Every option has a short description of what it leads to by selecting it.We also added layouts for each of those methods that are mentioned above. We added customized draw able images and icons so that it looks more interactive. Some libraries are used for the firebase connectivity with the application which retrieves information from the database.



# 

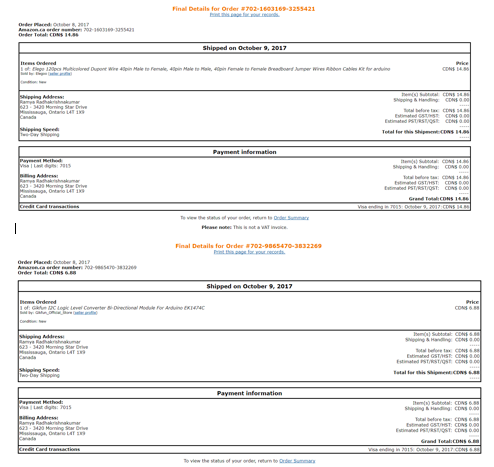
# 5. Project Overview

## 5.1 Fingerprint Sensor

## 5.1.1 Introduction

E-money transfer is a convenient way to send, receive or pay bills in store using a simple QR code generator. We are using a finger print sensor in collaboration with the software app. The sensor is used as a safety measure to prevent loss of information. In order to access your account, you need to login using your login credentials including the finger print pattern that is registered along with your bank account in the E-money application. Basically the finger print lets you enroll, delete, search or generate a picture of your finger. To do this we used a raspberry pi along with a usb to ttl convertor and jumper wires.

## 5.1.2 Invoice/Bill:



Ordered the sensor from Adafruit website. Also ordered raspberry pi, USB to TTL and jumper wires from Amazon. It roughly took a week to reach.

### 5.1.3 Budget

* The Fingerprint sensor (751) costed $64.00.
* The Raspberry pi costed $99.99.
* The USB to TTL convertor costed $6.99.
* The Jumper wires costed $5.99

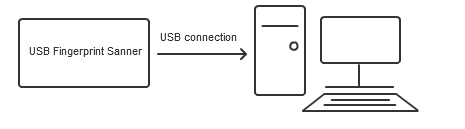
## 5.1.4 Time Commitment

The time commitment was a cake walk since I created a schedule during the start of my semester.

It really was one reason that we organized the work properly and was able to achieve the milestones as intended.

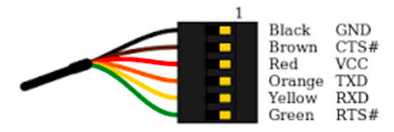
Setting up the hardware must take around 20 minutes provided if you have the equipment’s and have previous knowledge of the hardware. We took roughly 1 week to set it up. We used python code to make the sensor work. The testing should take around 10 minutes, but actually took time to read the code and understand what was actually in the code. It took 3 days to work on understanding the code and then successfully executed.

## 5.1.5 Mechanical Assembly of my Fingerprint sensor

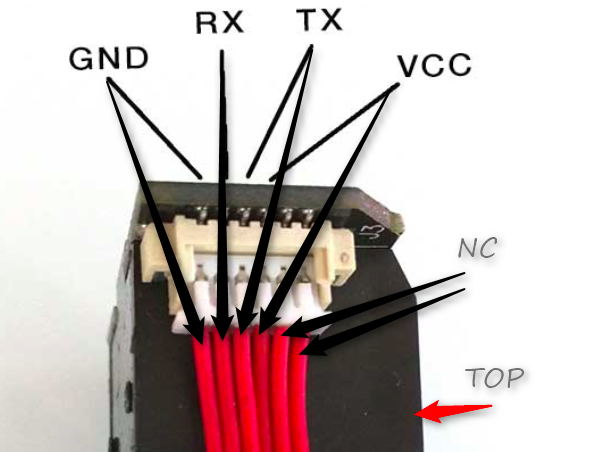


1. Connect the fingerprint to the USB to TTL convertor as follows:

2. This is the pinout of the USB to TTL.



3. The pinout of the fingerprint sensor.



4. Connections using the jumper wires:

\*Connect the GN of the sensor to the GN of the TTL converter.

\*Connect the RX of the sensor to the TX of the TTL converter.

\*Connect the TX of the sensor to the RX of the TTL converter.

\*Connect the VCC of the sensor to the VCC of the TTL converter.

5. Then connect the sensor through the USB port of the Raspberry Pi.

## 5.1.6 Power up

So once assembled the hardware part, then connected my mouse and keyboard along with the charger cable to my pi and then turned it ON. Also we need not connect the Ethernet cable since we already configured the pi to connect it to the Wi-Fi. We were able to see the sensor light up. You will be able to achieve this if your hardware connection is perfect.

## 5.1.7 Installation of the Raspberry Pi Fingerprint Library

We need to go to the root for some commands of the installation. Then we started a terminal session and typed the following:

sudo bash

Now add the necessary package sources:

wget -O - http://apt.pm-codeworks.de/pm-codeworks.de.gpg | apt-key add -

wget http://apt.pm-codeworks.de/pm-codeworks.list -P /etc/apt/sources.list.d/

Then had to update the available packages and install the Python library as follows:

apt-get update

apt-get install python-fingerprint –yes

Now the packages are installed and ready for testing.

## 5.1.8 Unit Testing

1. As mentioned above the package has files for storing a new fingerprint, reading out and deleting stored fingerprints. Now we tested the enrolling option by issuing the following:

python2 /usr/share/doc/python-fingerprint/examples/example\_enroll.py

2. The terminal asks you to place the finger you want to enroll on the sensor, wait for the message "remove finger" and then again "place the same finger".

Now the finger is given an ID and it is enrolled.

3. Now we test if the finger is recognized. So we issued the following script:

python2 /usr/share/doc/python-fingerprint/examples/example\_search.py

4. Then put the finger on it again. If the fingerprint on the Raspberry Pi is detected, a message like this appears:

Currently stored templates: 2

Waiting for finger...

Found template at position #1

The accuracy score is: 90

SHA-2 hash of template: 3aa1b01149abf0a7ad0d7803eaba65c22ba084009700c3c7f5f4ecc38f020851

## 5.1.9 Production testing

The testing was successful. We were was to make the sensor work by enrolling the finger, delete the finger by giving the appropriate ID, search for a fingerprint that was enrolled and retrieve the fingerprint image. This is a very reliable for security purposes.

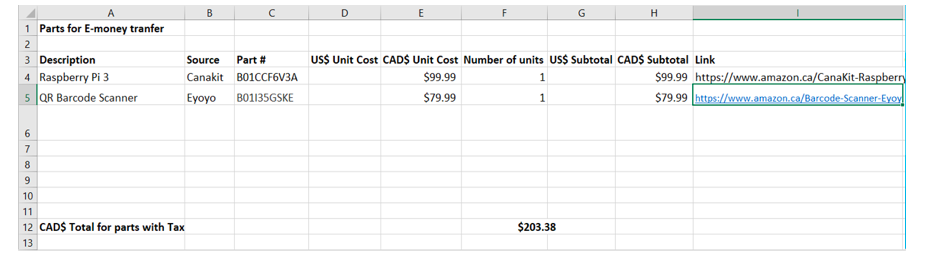
## 5.2 QR Scanner

## 5.2.1 Introduction

This project is about e-money transfer by using Qr code sensor. The components that have been used for this project are raspberry pi 3 and Qr code sensor. The raspberry pi comes with a case, power adapter and micro sd card with raspbian OS. A monitor, mouse and a keyboard is required to operate raspberry pi, unless you have laptop to connect it with. To connect a laptop with the pi, you have to configure specific settings accordingly which is not covered in the following instructions. The qr code sensor is connected through a usb with raspberry pi. Recommended libraries have been used to power up the sensor. I plan to demonstrate transactions of an amount of money using the qr code sensor. It can be used in stores to buy stuff and to transfer money. Building this project should require a couple of hours if you follow these instructions.

## 5.2.2 Bill of Materials Budget

We ordered raspberry pi from amazon and it arrived within three days as we used amazon prime shipping. For the most part, we used QR code scanner and raspberry pi. QR code sensor can be used to scan both barcodes and QR codes which is one of the reasons they are more expensive compared to barcode scanners. They can also scan both 1 dimensional and 2 dimensional codes and are equipped with laser.

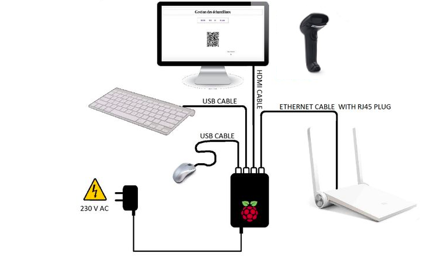


## 5.2.3 Time Commitment

Once we received the equipment, it took approximately 2 weeks to complete the project. This period includes the time that we spent on researching and building the sensors in order for them to operate in the way we wanted them to. The QR code scanner is connected to the pi through a USB interface which took about 2 minutes to connect and test. However, the part that took the most amount of the time was creating the python code.

## 5.2.4 Mechanical Assembly

We did not need any extra electrical equipment such as wires, motors, resistors, capacitors etc. The assembly of this project requires raspberry pi, power adaptor and QR code scanner with a USB interface. Just used a mouse, keyboard and monitor to connect the pi and to test my sensor.



## 5.2.5 Unit Testing

When we first received the sensor, we were testing it by connecting it to my powered up pi. Initially, was just scanning randomly generated QR codes and barcodes from the internet. And every time we scanned, the screen we were on, happened to scroll down automatically. So, thought that the scanned item was being stored somewhere on the SD card of the pi. However, the attempt was unsuccessful. Then thought that maybe it would be stored in a text editor like notepad or MS word. So we scanned while opening a text editor and the code was stored in the file.

## 5.2.6 Production Testing

Then we developed the Python code for the intended function to be carried by the sensor. The developed code works in a way that it is connected to the firebase database account. Once the QR code generated by the application is scanned it will recognize the amount that was stored in the QR code and it will add that amount to the linked bank account in the firebase database.

# 6. Problems and Approaches

### 6.1 Data Communication

The mobile application has many fragments. One of the fragment is used to get the user information. This asks the user to enter the card number, email ID and fingerprint. The issue was the application being connected to the Firebase database was not able to fetch the account information despite giving the correct credentials.

Also issue with the python code is something about the version. It is an in built function of the firebase which may not compile the python code if the version of the python code and firebase is outdated. The python code will not be issued or compiled resulting in not displaying the GUI. So it is always wiser to have the same working version for both the firebase and the python code to avoid such minor errors. This intern leads to not fetching the data stored in the firebase database which is a serious issue. Once the version was changed, the python code was able to access the firebase database and present the GUI.

### 6.2 LCD Display

A Liquid Crystal Display was incorporated in the project. The LCD’s purpose was to present the GUI of the project. But the issue was that, it wasn’t turning on. But the touch of the screen happened to work. The team then came up with a conclusion that it was a hardware issue. After certain encounters and learning phases the problem was with the adapter of the LCD. The LCD required a minimum voltage of 5V but the adapter that was used only supplied 3.3V. Once the adapter was replaced, the LCD was up and running displaying the GUI once the python code was established.

### 6.3 Mobile Application

The mobile application had a difficulty which intern comes under software issues. The buttons and variables are used in the fragments. These buttons and variables couldn’t be passed to other fragments. This was the issue, one solution was to bundle the buttons or variables. The procedure is to bundle these buttons or variables into one and then pass this bundle as indented to other fragment. Another solution used is also similar to the previous one. Here the only difference is we bundle the object. And this bundled object will be passed to another fragment. The issue with the second solution is that it sometimes returns a null. The 1st solution is more reliable than the second.

### 6.4 QR code email

One of the case scenarios is that using the mobile application the user can send money to another user whom he owes. So to perform this transfer it requires the sender to generate the QR code and send that code over email. Initially here the issue was QR code was not transmitted via email to the receiver. It keeps asking the sender to save the QR code to the images folder of the phone. Only after saving the QR code in the image folder the user was able to email it. Then after numerous trouble shooting it was fixed by incorporating some libraries into the Android code. Now the QR code need not be saved into the image folder. It can be created and sent directly by creating the QR code for the indented transfer and entering the receiver’s email ID.

# 7. Progress Reports

## 7.1 Report 1

Ramya Radhakrishnakumar < [ramyarkkumar27@gmail.com](mailto:ramyarkkumar27@gmail.com) >

To: Austin Tian < [Austin.Tian@humber.ca](mailto:Austin.Tian@humber.ca) >

Cc: Sushant Sharma < [sushantsharma88888@gmail.com](mailto:sushantsharma88888@gmail.com) >, Vinokkumar Uthayakumar<[ukvino3@gmail.com](mailto:ukvino3@gmail.com) >

Dear Austin Tian,

We have decided student A to be Ramya Radhakrishnakumar, student B to be Sushant Sharma and student C to be Vinokkumar Uthayakumar. This email is to update the progress shown in e-money since week 3.

Since our very idea of the project has been overlooked, few changes had to be made to the mobile application. So we dedicated more attention towards it. Vino worked on the changes to be made in that application and some of the modifications are; we added a navigation bar that uses fragments to switch between each layout between screens. This actually gives a proper flow for the application and also made it a bit more user friendly now than before. Some of the methods/functions added are contact, enrolment, overview, pay and transfer. We also added layouts for each of those methods that are mentioned above. Moving on to the database part controlled by Sushant, the database has been created successfully and we tried to connect it to the mobile application. It wasn’t connecting initially because of some technical issue. We spent more time on fixing the issue and now we are able to connect the database to the mobile application. We are able to interact between the application and the database now. Ramya made sure if the hardware/sensors incorporated in this project are in proper working conditions. We are yet to work on the hardware and database connection this week. Also we carried out some research on the hardware parts that might be needed in the future.

So far there is no change in the finical status of the project. It remains the same. Further updates on the budget are to be seen down the following weeks.

 As per the project schedule we have completed almost as intended until week 5, yet the major milestone is yet to be completed and that is the connection of the database to the hardware (QR scanner and Fingerprint sensor) which we are planning to work on week 6 and also we might get help from Kelly in the prototype laboratory.

Links to the Media added on GitHub (<https://github.com/RamyaRadhakrishnakumar/ceng355>):

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/PastedGraphic-1.png>

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/PastedGraphic-2.png>

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/Screen%20Shot%202018-03-05%20at%202.04.40%20PM.png>

Some of the websites used as reference for the application and database development are as follows:

<https://www.youtube.com/watch?v=F6UWb9FNnj4>

<https://stackoverflow.com/questions/22882074/download-image-and-its-associated-info-through-qr-code-on-android-app>

<https://stackoverflow.com/questions/45967649/firebase-authentication-error-in-android>

<https://firebase.google.com/docs/auth/android/google-signin>

<https://developer.android.com/studio/write/firebase.html>

<https://firebase.google.com/docs/database/android/start/>

Yours sincerely,

Ramya Radhakrishnakumar, Sushant Sharma & Vinokkumar Uthayakumar

Humber College

School of Applied Tech

## 7.2 Report 2

Sushant Sharma < [sushantsharma88888@gmail.com](mailto:sushantsharma88888@gmail.com) >

To: Austin Tian < [Austin.Tian@humber.ca](mailto:Austin.Tian@humber.ca) >

Cc: Ramya Radhakrishnakumar < [ramyarkkumar27@gmail.com](mailto:ramyarkkumar27@gmail.com) >, Vinokkumar Uthayakumar<[ukvino3@gmail.com](mailto:ukvino3@gmail.com) >

Dear Austin Tian,

This email is an overview of our progress during week 6 updated by student B.

This week we worked on the application and database connectivity with the hardware as well as the app. Sushant and Vino worked on validating the login, registering and authentication with the firebase database. Ramya spent some time researching on how to connect the firebase database with our raspberry pi, so that we can store our sensor’s scanned data into it.

We created a separate database for Merchant. We are able to store the value scanned by QR code scanner and add it to the merchandiser database.

Some of the problems/challenges encountered this week was, we tried to hide the navigation bar after the user logs in so that they stay logged in even if they click on different fragment/activity. We are also looking for a way to hide the login fragment when the user is already logged in until they click logout button.  We also imported the libraries for python to connect it to the database. We are up to date according to the schedule.

The financial status is the same as before. Most likely it will not change as we have acquired all the necessary equipment.

Links to the Media added on GitHub (<https://github.com/RamyaRadhakrishnakumar/ceng355)>:

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/week6/transfer.PNG>

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/week6/enrol.PNG>

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/week6/login.PNG>

Some of the websites used as reference for the application and database development are as follows:

<https://pypi.python.org/pypi/python-firebase/1.2>

<https://www.raspberrypi.org/forums/viewtopic.php?t=183455>

<https://www.youtube.com/watch?v=BJfVoaifnzc>

<https://repl.it/repls/WorrisomeArcticBlock>

<http://python-textbok.readthedocs.io/en/1.0/Variables_and_Scope.html>

Yours sincerely,

Ramya Radhakrishnakumar, Sushant Sharma & Vinokkumar Uthayakumar

Humber College

School of Applied Tech

## 7.3 Report 3

Vinokkumar Uthayakumar<[ukvino3@gmail.com](mailto:ukvino3@gmail.com) >

To: Austin Tian < [Austin.Tian@humber.ca](mailto:Austin.Tian@humber.ca) >

Cc: Ramya Radhakrishnakumar < [ramyarkkumar27@gmail.com](mailto:ramyarkkumar27@gmail.com) >, Sushant Sharma <[sushantsharma88888@gmail.com](mailto:sushantsharma88888@gmail.com) >

Dear Austin Tian,

This email is an overview of the progress during week 8 and week 9 updated by student C.

In the last two weeks, work was focused on the Graphical User Interface(GUI) for the sensors. Now the GUI for QR code scanner and fingerprint programs has been created. There is a plan to use LCD display for the sensors. Also, the team is working on subtracting the money from the bank account of the user after they have made a transaction(Pay/Transfer). Also, work was done on python code for fingerprint sensor to store information in the database based on the user’s fingerprint and email address provided.

There have been updates made to the layout of the application.

The problems being faced in the last 2 weeks are as follows, firstly the user information from the fire-base database was not being displayed. Here the user information could be seen from the database when debug is used. Finally, the information is being displayed on the overview such as the account number and account balance of the current user. Secondly, the object class is not working with fragments. The problem is, the buttons could not be transferred/passes from one class to another. The team tried fixing the issue by incorporating instance to make it work but it wasn’t successful. We are not able to call a method from one fragment to another fragment.

We are on track with the project schedule.

There is a recent purchase that adds up to the financial update of the project. An LCD display has been purchased for CAD $55.

Links to the Media added to GitHub (<https://github.com/RamyaRadhakrishnakumar/ceng355)>:

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/qr%20code%20scanner%20gui.PNG>

<https://github.com/RamyaRadhakrishnakumar/ceng355/blob/master/fingerprint%20gui.PNG>

<https://github.com/RamyaRadhakrishnakumar/ceng355/tree/master/Layouts%20of%20the%20application>

# 8. Conclusions

# 9. Recommendations

# 10. Technical References/Bibliography

<https://www.youtube.com/watch?v=F6UWb9FNnj4>

<https://stackoverflow.com/questions/22882074/download-image-and-its-associated-info-through-qr-code-on-android-app>

<https://stackoverflow.com/questions/45967649/firebase-authentication-error-in-android>

<https://firebase.google.com/docs/auth/android/google-signin>

<https://developer.android.com/studio/write/firebase.html>

<https://firebase.google.com/docs/database/android/start/>

<https://pypi.python.org/pypi/python-firebase/1.2>

<https://www.raspberrypi.org/forums/viewtopic.php?t=183455>

<https://www.youtube.com/watch?v=BJfVoaifnzc>

<https://repl.it/repls/WorrisomeArcticBlock>

<http://python-textbok.readthedocs.io/en/1.0/Variables_and_Scope.html>

# 11. Appendices