

ASHESI UNIVERSITY COLLEGE

**INTERBANK MONEY TRANSFER
SYSTEM FOR
PC AND MOBILE**

WILHELM ARMAH KWAKU HESSE

2010

APPLIED PROJECT

Declaration

I hereby declare that this dissertation is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:.....

Candidate's Name:.....

Date:.....

I hereby declare that the preparation and presentation of the thesPs were supervised in accordance with the guidelines on supervision of thesis laid down by Ashesi University College.

Supervisor's Signature:.....

Supervisor's Name:.....

Date:.....

ACKNOWLEDGEMENT

I am forever grateful to my parents, Mr. and Mrs. Ludwig Hesse my grandmother, Mrs. Gladys Hesse and my entire family for their love, support and encouragement to persevere.

My gratitude goes to Ashesi University College for giving me the privilege to study and meet such resourceful and intelligent individuals.

I am thankful to my supervisor Mr. Kwadwo Gyamfi Osafo-Mafo for taking time to read and critique this project as well as Dr. Nathan Amanquah for his advice and support in completing this project.

I would like to show my appreciation to the lecturers that took their time and used their effort to help me understand the concepts of Computer Science and Liberal Arts Core.

Finally, to all who in one way or the other, helped in the completion of this project I say a big thank you.

Abstract

“Money makes the world go round”. The importance of money has made it a very powerful tool man cannot live without. Thus, it is very important for people to have money when they need it.

The mobile phone has developed drastically from a device just for making and receiving calls to one that can be used to take pictures, send emails, connect to the internet and do many other things that at one point in time were dedicated to specific devices such as computers.

Mobile phone users in Ghana have increased over the years and are still increasing with some people having more than one phone. The introduction of mobile phones that have the capability of accessing the internet has brought rise to applications developed solely for the mobile platform and also websites that are built to fit the mobile specifications such as screen size, page size, size of images, and processing power.

Contents

1.	Introduction.....	1
2.	Previous Work.....	4
2.1.	The Old System of Going to the Bank	5
2.1.1.	The problem	6
2.1.2.	The Solution	6
2.2.	MTN MobileMoney	7
2.2.1.	The Problem	8
2.2.2.	The Solution	9
2.3.	M-Pesa Money Transfer.....	9
2.3.1.	The Gap	11
2.4.	PayPal	11
2.4.1.	The Problem	12
2.4.2.	The Solution	13
3.	Approach.....	13
3.1.	Differences.....	13
3.2.	Ideal Situations and their Reasons	15
4.	Methodology	17
4.1.	Parts That Have Been Implemented	17
4.2.	Steps Taken during development	19
4.2.1.	Developing Requirements	19
4.2.2.	Design	24
4.2.3.	Development	33
5.	Testing and Results.....	35
6.	Discussion	40

6.1. Approach	40
7. Limitations & Conclusion.....	42
Works Cited	47
Appendix	48
Screen shots	48
Mobile Screenshots.....	51
Figure 1 Aesthetics.....	25
Figure 2 Error Reporting	26
Figure 3 System status.....	26
Figure 4 Notifications.....	27
Figure 5 Main menu	27
Figure 6 Page Flow Diagram	29
Figure 7 Bank Database schema	30
Figure 8 Payinn Database schema.....	31
Figure 9 Class Diagram.....	32
Figure 10 CAPTCHA image	37
Figure 11 Login Page.....	48
Figure 12 Registration Form	48
Figure 13 User Home Page.....	49
Figure 14 Send Money.....	49
Figure 15 View Transactions.....	50
Figure 16 Change PIN.....	50
Figure 17 Add a Bank	51
Figure 18 Home Page J2ME	51
Figure 19 User Registration J2ME.....	52
Figure 20 User Account J2ME.....	52
Figure 21 Send Money J2ME.....	53

1. Introduction

Money is a commodity that is very important in the daily lives of individuals. Money is used in various aspects of life as a tool for payment. It is normal to be in a situation where one does not have enough money and will need to go to the bank or the nearest Automated Teller Machine (ATM) to withdraw some money. It therefore becomes a big problem if the bank is nowhere close and the only ATM around does not belong to one's bank.

The system is named Pay-Inn. It is an Interbank Money Transfer System with a web server component as well as a client component. The client component has two parts, a standalone mobile client application developed with J2ME and a web application developed to run on both mobile and PC.

The aim of this system is to minimize the distance between a person and his money. The client application puts a user's bank account in their pockets; more like carrying a debit card and the ATM wherever they go. There are situations when one cannot pay for a service or product directly but may need to do it over the internet or over another virtual medium. In a situation like this, the application can be used as a means of payment.

The application is tied directly to the bank accounts of its users and can be used to;

- Transfer money from one user to another.
- Request money from other users.
- View old transactions.
- Change passwords and PIN.

Since the application is tied to the bank accounts of each registered user, the monies transferred are removed and put directly in the respective accounts.

There are various applications for a system like this in Ghana such as;

- Online stores that use it as a means of receiving payment from their clients.
- Organizations that use it as means to pay their employees.
- People can use it to pay for services that they receive in place of cheque.
- Bill payments.

The main challenge with this application is the ability to connect to the different banking interface of the different banks. This is because all the banks do not use the same interface for connecting to third party applications. Thus, for each bank the application connects to, there has to be a different interface developed to fit that which the bank is using.

Another challenge that comes up with such a system is the security of the application. An application that manipulates people's money directly in their bank accounts needs to be secure from hackers, viruses and other security threats. The application should run on systems that are trusted and make sure that there is no misappropriation of people's money. It is important to take issues such as connectivity and other network problems into consideration when developing an application like this one. A person cannot afford to have someone lose money due to a failure in the connection between the application and the bank. It is also important to make sure that

the information about the users is kept in a secure database with the highest security level implemented on it. It should not be easy for hackers to break into a bank's database to get access to confidential data of clients. Transmission of information across networks should also be done in secure environments. Data has to be encrypted with keys known only to the sender and the recipient servers. There should be checksums implemented to make sure that the sent data is in no way corrupted.

In the application, due to the lack of a banking interface to test with, a pseudo banking environment mainly made up of a database of different bank accounts belonging to different users in different banks was created. A user is required to use an email address and a password to login into the system but before a user can do that he will have to register. During registration, the user is required to provide personal details like their name, email and phone numbers. Due to security reasons, users are required to have passwords that they will use when accessing their accounts. The password is required to have some default value in order to make it difficult for other people to figure out. These requirements are:

- A minimum of six characters.
- At least one character must be a number.
- At least one character must be in Uppercase.
- The password cannot be the same as the username.
- The password cannot be the user's name.

A user can only access their profile after their email verification. When a user registers with the system, an email confirmation is sent to their email

address with a link that connects them with the application. The account is verified once the link is clicked and the corresponding page is loaded. Users are given the opportunity to add a bank account to their profile. The bank account is needed for the user to be able to send and receive money. The user has to specify their bank account number and bank name. In order to verify any transaction, the user is required to create a PIN.

This paper is going to look at what has been done in relation to the application. Focus will be on what has been done differently by other people or organizations that have developed similar applications and by that compare this application to theirs. Another thing to be looked at is the approach that was followed in developing the application and also the methodology used for the requirements and different functionalities of the application. Analysis of results in relation to the requirements developed would be next, followed by recommendation of other functionalities that can be added to the application. The paper will then conclude with general impressions of the application and how best it can be improved to serve the public.

2. Previous Work

Similar applications or systems that have been developed are:

- The old system of going to the bank to send and withdraw money.
- MTN Mobile money.
- Safaricom's M-PESA in Kenya.
- Pay-Pal

For each of these various approaches, the general overview, key insights and situations in which they fail or succeed will be analyzed.

2.1. The Old System of Going to the Bank

The term old system is because, with the improvement in technology and the speed of the internet, different banks have introduced internet or SMS banking which more or less saves the user the time of going to the bank. The only problem with internet or SMS banking is that the user is limited to the things that can be done with the application. Internet banking applications allow users to check their balance, pay utility bills and also upload mobile phone credit onto their phones. It does not allow the user to transfer money from one account to the other thus it cannot be used for payment of services and products outside what has been specified for the application. It is therefore obvious that, the old system of having to go to the bank to get money still exists, and may exist for a long time to come. With the introduction of Automated Teller Machines (ATM), the need to go to the bank decreased since one can withdraw money from them. The problem with this is that an individual has to go physically to the bank to deposit money otherwise there will be no money to take from the ATM.

Going to the bank is by far the most guaranteed way of getting money even though it may at some point be time consuming. At the bank, things can go wrong, such as network problems and system failures but the employees there will find a way to give money to clients and then update their accounts when the system comes back online. Most of the time it is easier to get

money from the branch where clients' accounts are located than other branches because the accounts can be accessed locally thus no network access will be required.

2.1.1. The problem

- Time consuming: Going to the bank is very time consuming. It becomes more time consuming when persons have to wait in queues for their turn. The time that is used standing in queues at the bank can be used to provide services to a customer or sell a product somewhere.
- Use of resources: By moving to and fro from a bank, resources are being used up; energy, money and time.
- Security risks: Carrying money physically to the bank is highly risky. One faces the risk of losing the money along the way to the bank. Cash cheques that are lost are easily usable by the next person that finds them.

2.1.2. The Solution

Pay-Inn can run on both PC and on mobile phones that have the means of connecting to the internet. By using this application one saves the time and the resources that may be used to go to the bank. Since the application can be accessed on the phone, one can send money to anyone who is registered with the application. If the merchant you are working with has an account with Pay-Inn then there is no need to write him a cheque and he will not

have reason to go to the bank. You can just login with your email address and password, take the merchants email and then send the money after your PIN has been verified. The money is transferred from your account to the merchants and the merchant will receive his money once he accepts the transaction. The cost of the transaction will be the fee for sending data across the internet. The application reduces the risk of having to carry huge sums of money to the bank. Loss of funds is minimized because the money transfer is done automatically.

2.2. MTN MobileMoney

“MTN MobileMoney is a convenient secure and affordable way of sending money, buying airtime, and making basic utility payments from your MTN mobile phone. This service is offered by MTN across its operations in partnership with local banks” [1].

This system is convenient for people who do not have bank accounts because the account is in no way tied to the user’s bank account. It is also available to users who do not have MTN mobile phones. All they have to is visit an authorized dealer and the transaction will be performed.

To be able to use this application, one would have to go to an authorized MTN MobileMoney Agent with an ID. The old SIM card is changed and for another one that has application installed on it. To make any transactions, the user would have to go to the authorized agent, deposit some money into his MobileMoney account and then start doing transactions. For a user to retrieve their money, they would have to go to the authorized dealer prove

their identity and then take their money. This system is very similar to the old system. The only difference is that with enough money in your MobileMoney account you can send money to anyone anywhere. It faces the same problems that the old system faces, especially if an authorized dealer is not located close to you.

In terms of security, this system is safe because the monies that are deposited are kept in a bank account. The application also utilizes the PIN which users will need in order to transfer money or withdraw money.

This application works fine so far as there are enough vendors around that users can access within a short distance. This then limits the time a user might spend waiting to get his money.

As mentioned earlier this application is very convenient for users who do not have bank accounts meaning these people do most of their transactions by cash. For these people they do not have to spend the extra time at the bank withdrawing and depositing money. They can just walk into the agent's office and then deposit their money into their MobileMoney account which will serve as a bank account for them.

2.2.1. The Problem

- Unavailability of authorized vendors: MTN MobileMoney vendors are very difficult to find making it difficult for users to access the service. Users cannot readily deposit money into their accounts or withdraw money if the nearest vendor is miles away from them. This system

depends solely on the availability of the vendors to collect and give money to users.

2.2.2. The Solution

Pay-Inn solves this problem by having the application tied directly to your bank account. Like MobileMoney a user still needs money in their bank account to be able to transfer some money and often than not the user may have to go to the bank. Otherwise, the user can use the application to request for money from another Pay-Inn registered user. The advantage of having the application tied to your bank account is that it eliminates the need for vendors. Users can go to any ATM to withdraw their money or even go to the bank to get their money.

2.3. M-Pesa Money Transfer

M-Pesa is similar to MTN MobileMoney but with a higher success rate in Kenya and in the UK. M-Pesa was introduced in Kenya in 2007 and has since “revolutionized money transfer services in the country by providing Safaricom subscribers with a fast, safe and affordable way to transfer money by phone” [2]. M-Pesa is owned by Safaricom and Vodaphone both telecommunications companies. It was developed mainly for the unbanked populace but is also known to be very popular in the banked as well. Aside transferring money the application can be used to pay utility bills and also buy airtime.

Getting registered with M-Pesa is similar to that of MTN MobileMoney. The user's SIM is exchanged with a new one that has the application installed on it. Users have to deposit money into their M-Pesa account to be able to do any transactions. Users will also have to go to the vendors in order to retrieve their monies.

Even though MTN MobileMoney and M-Pesa are the same kind of service, M-Pesa is doing very well compared to the MobileMoney. This is mainly due to the problem that was outlined with MobileMoney. "Currently, 300 bank branches offer M-Pesa services. With over 7000 agent outlets employing more than 12,000 people directly and indirectly" [2]. A higher number of agents means that it is easier for a user to retrieve his money and also transfer money. Another reason may be that both Safaricom and Vodaphone are doing well to develop the application is something much bigger. "Safaricom...partnered with Equity Bank ... to introduce a service that enables customers withdraw cash from the bank's ATMs without using an ATM card" [2]. With this improvement in the service users do not even need the agents as they can go to the ATM and withdraw their money.

Security is another major area that M-Pesa is strong in. Each user has their PIN which they use in transactions. Users are always encouraged to cross check the phone numbers of the people they send money to in order to make sure that it's the right person. In the case of loss or theft of a user's phone, the user can go to a vendor and retrieve their new SIM and also have their money back.

M-Pesa is doing very well and will still do well if it increases the means through which people can access their money.

2.3.1. The Gap

The problems with M-Pesa are not clear. This is because it seems to be running effectively and is still growing in popularity. Having an application tied to your bank account would not beat an application that has over 12000 outlets. The ability to withdraw money from the ATM without your ATM is also a plus for the system; something that can be added to Pay-Inn.

2.4. PayPal

“PayPal is the safer, easier way to pay and get paid online. The service allows anyone to pay in any way they prefer, including through credit cards, bank accounts, buyer credit or account balances, without sharing financial information” [3].

The success of PayPal in my opinion is due to the fact users have different options by which they can pay or send money and in all these ways critical information is not shared with the other party. Credit Card fraud and other types relating to money are sometimes made easy because users have to enter them directly into the websites or other places that go giving other people the advantage of seeing and using their cards once they are not available.

PayPal is available both on mobile and on PC. A user has different options available to them. They can tie their PayPal accounts to their bank account, debit or credit cards or the other options available. Since the debit is attached to the bank account, any transaction made with it directly reflects in the bank account.

PayPal acts like a virtual bank account into which users can withdraw money from their bank accounts and deposit. When these monies are deposited in the PayPal account, they can be used for any transaction be it payment of service or a product or just transferring of money. The money can also be deposited back into the bank account of the user. Whenever a transaction is done an email notification is sent to the parties involved and the transaction is saved in the application database.

Most online stores use PayPal on their website to sell to the public. Often than not a user has to a PayPal account holder before they can use the application.

The mobile aspect of the application is web based which means it can be accessed using the web browser on a phone. There are special applications built for specific phones like the iPhone, Android and Blackberry. There is also an SMS service available.

2.4.1. The Problem

- PayPal is not in Ghana: Even though PayPal is available in different countries and is available in different currencies, Ghana is not part for reasons best known to PayPal.

2.4.2. The Solution

Pay-Inn is being built with the Ghanaian and African market as the main target. Once Pay-Inn has developed and been accepted as secure application it can either be open to countries outside Africa or maybe integrated with PayPal to provide similar services.

3. Approach

In this part of the document the approach used in developing Pay-Inn will be analyzed. If applicable, comparisons will be made between the approach taken and the previous ones mentioned above.

More research into each of the previous approaches brought to light the fact that PayPal was the closest approach to this system.

The main idea taken from the PayPal approach was the transfer of money between different bank accounts. This was not directly the case in PayPal but that was the bottom line.

3.1. Differences

- Pay-Inn lacks the virtual bank account capability mainly because it's a starter application and its targeted users do not do online shopping at least not in Ghana and even if they did there is close to nothing to purchase online. Thus my main concentration went into moving money from one bank account to the other.
- With this application, the user only has the option to use their bank account unlike PayPal that gives its users different options. Most of the

other options that PayPal provides are not available in Ghana and even though the debit is available, it is not widely used by most of the banking public.

- Pay-Inn is deployed on a web server and also there is a J2ME application that can be downloaded onto phones that support Java. It does not support SMS like PayPal.
- PayPal only asks for a PIN when accessed via mobile. It does not use the PIN when users are sending or receiving money on the PC. Users are only required to specify their logon credentials when logging in and that is used for all the transactions that are made on PC. With Pay-Inn, once a user adds a bank account, they are required to specify a PIN. Anytime the user wants to do a transaction, the PIN is requested and if the PIN is wrong, the transaction will be denied. Thus increasing the security feature of my application.
- To make sure that it's actually a human being accessing the application, CAPTCHA a system that creates an image with texts in it and requires the user to enter the text was implemented. It only runs on the web application and not on the Java application. Before a request can be processed the user must enter the values in the image into an input box. The text is then checked against the text in the image and if they match then the request can be completed. For example before a user can register, they would have to enter this code verifying that it's a human being accessing the system. Also when

money is being transferred or requested from another user the CAPTCHA tool is utilized.

- PayPal usually allows users to add more than one bank account to their profile so that they have the flexibility of changing the bank accounts that they use for a transaction. The newest bank account is set as the default account and that will be used automatically when a transaction is initiated until the user changes it. Pay-Inn only allows the user to have only one bank account. The user is always free to change this bank account when he desires with no difficulty. The one account is always verified before a transaction is allowed to continue.

3.2. Ideal Situations and their Reasons

- Employee Payment: At the end of the month, employees have to be paid and unless these employees are paid in cash, cheques will have to be mailed out to each and every employee's bank account. If the employee base is very large, that would mean a huge cost of mailing the cheques to the different banks that they belong to. With an application like Pay-Inn with a little modification to suite the organizational structure, all the employees can be paid with a click of a button. When this is done notification will be sent automatically to all the employees informing them of the transfer. The organization gets to save on resources that may not be seen in the short term but will reflect in the long run to have been very worthwhile.

- **Payment of Goods and Services:** Since Pay-Inn is tied directly to your bank account, you have all the money you need in your pocket. Therefore paying for services will not be a problem so far as you have money in your account. If your client or service provider is on Pay-Inn, you can both agree to use it for your transactions. Once the recipient of the money accepts the money, the transaction is quickly completed and email notifications will be sent to the users. Merchants do not need to rush to the bank to deposit their money because the money has already be deposited by the application.
- **Online Transactions:** Pay-Inn can be used to pay for goods and services online or on the internet. Online shops and other service providers shoe sign up on the application can receive money from their clients to complete a transaction. This is ideal because it allows users from different places to do business. PayPal for example is accepted in so many different countries around the world, thus someone in the United Kingdom is able to buy a product from someone in Japan and the means of payment is through PayPal. Once the payment has been confirmed the product is shipped to the buyer. Users do not need to meet each other to do transactions. No physical money has to be sent to anyone, thus the risk of losing the money is cut out. With a reliable network system, this application would work close to perfect.
- **Confidential Transactions:** Pay-Inn does not at any point in time give out critical information about their users to anyone. Whenever a transaction is made the only information about the other party is his

email address. No bank account details are sent with the money transferred. Users are always guaranteed that their information is safe with Pay-Inn and will not be given out to any third party without their authorization. Users are also advised to keep their passwords and PIN confidential and not share it with other people.

4. Methodology

The methodology looks at the different parts of the application that have been developed and the steps that were followed to develop them.

4.1. Parts That Have Been Implemented

- Client Side Interface (HTML & CSS): This part of the system was developed using mainly HTML code. The main reason being that HTML is the basic language that all browsers can interpret. HTML in conjunction with CSS is used to design the page in terms of layout, color, font size and the basic look and feel of the application. There are other types of languages that can be used for example xHTML which is advanced compared to HTML but based on HTML also there is HTM which is not really different from HTML as well.
- Database (MySQL): For the database management system MySQL was chosen mainly because it's familiarity and thus ease to work around in case of problems. There are other possible database management systems (DBMS) that can be used some of them being Oracle, MSSQL and PostgreSQL. Also MySQL has the mysqli extension in PHP which loads all the functions needed to be able to use PHP with MySQL. Since

it comes with PHP the extra work of having to manually compile PHP with an extension that supports MySQL.

- **Server Side Code (PHP):** PHP is used for connecting the web server and database server. PHP is used to send requests from the client to the servers and the servers respond back through PHP as well. A reason why PHP was used was because it supports CAPTCHA. CAPTCHA depends on the GD library that exists in PHP. This library makes it possible for the image to be created and filled with the texts. In place of PHP, ASP or ASP.NET can be used. PHP is open source and is free. ASP is owned by Microsoft and users will need a Microsoft Operating System and would in order to use it. PHP on the other supports different operating systems.
- **AJAX (JavaScript & XML):** AJAX (Asynchronous JavaScript and XML) works like PHP in a way by sending and receiving data to and from the servers. With JavaScript and XML users can send requests to the server and receive a response without having to reload the page that they are on. JavaScript allows the different parts of page to be accessed and changed according to the request or response. XML is used to send requests to a PHP page that connects to the server, sends the request and gets a response. The response is sent back through the XML and then to the JavaScript and then the displaying is done.
- **J2ME Mobile Application:** The mobile application is a standalone application that has been developed for installation on Java capable

phones. The Python language could have been used to develop the mobile application but familiarity with Java pushed the idea to use J2ME instead. The mobile application has most of the features that the PC version has.

- Mobile Web Application: This part is run directly from the mobile browser. To use this, the user has to enter the url of the application in the mobile browser. This part of the application is has no drastic changes only that it is built to fit the screen size of any phone used to browse the website.

4.2. Steps Taken during development

4.2.1. Developing Requirements

In developing an application one of the first things needed to be done is the development of requirements. The requirements help to best outline the different things that the application should do and their priority.

4.2.1.1. Specification Requirements:

- Web Application:
 - Internet Connection
 - Internet Browser (Internet Explorer, Mozilla Firefox, Opera-Mini)
 - Database Server
 - Web Server
 - Computer

- Mobile Phone with internet capabilities (GPRS, WiFi, EDGE)
- Mobile Application (J2ME):
 - Mobile phones with Java Support (MiDP 2.0, CLDC 1.0)
 - Mobile phones with internet capabilities (GPRS, WiFi, EDGE)

4.2.1.2. Functional Requirements:

These have to do with the main functionalities of the application. Below are the functional requirements for my application

- User Registration: Once a new user accesses the web application, the user should be able to register on the application. There should be links on the home page that give users the chance to register with the application. Users need to specify their email address, password and other personal information in order to register. For security reasons users are required to have a password with:
 - A minimum of six characters.
 - At least one character must be a number.
 - At least one character must be in Uppercase.
 - The password cannot be the same as the username.
 - The password cannot be the user's name

Users will also be required to enter a code that will be displayed in a CAPTCHA image in order for the registration to be completed. Data should be validated as they are entered and messages should be displayed on the status of the entered data. All data must be correct before registration can proceed. After successful data entry and

submission, an email confirmation will be sent to the user with a link embedded. The user will have to click on the link and that will verify his email address and complete the registration process.

- **User Login:** Once a user has registered he can now login to the application. To login a user needs their email address and their password. After the data has been entered, the user clicks on the logon button after which his credentials are validated. If the email and password are found in the database, the user is logged in successfully otherwise an error message will be displayed asking the user to enter the correct email and password.
- **Password Reset:** users that forget their password should be given a link that will allow them to notify the application. On this page the user will be required to enter their email address. The email is checked for availability in our database and if it exists a new password will be sent to the user via their email. Once the user logs in with the new password, they will be required to change the password.
- **Add a Bank:** Once a user joins the application, they will need a bank account to be able to do any transactions. There should be a button that users can click and then add their bank account to their profile. Users can only have one bank account in their profile. They are allowed to change their bank accounts as they wish. To add a bank account, a user will need to specify their account number, choose their bank from a list of banks that have signed up for the application. If the user's bank is not available then the user will not be able to join the

application. There is also a CAPTCHA image on this page to validate the data that is entered.

- **Send/Request Money:** After a bank account has been added, a user can now send money, receive money, request for money and receive requests for money. Before a user sends money, he would have to specify the email address of the recipient, the amount to send, his PIN and then a CAPTCHA code. The recipients email is checked and a validation on his bank account is done. The senders account is validated and the balance is checked against the amount specified to be transferred. In the case where the money specified is greater than the balance in the account, the user will be asked to reduce the amount. On receipt of money a notification will be placed on the home page of the recipient and the recipient will have to approve or deny the transaction. On approval the money will transferred to his account and a notification will be sent to the sender. On Deny the money will be sent back to sender's account and a notification will be sent. The same is done for requests.
- **View Transactions:** Users should be able to view past transactions that they have made. The user will be able to filter the transactions based on type i.e. Payments Sent, Payment Received, Requests Sent and Requests Received.
- **Log Out:** Once a user has finished a transaction they can logout. Logging out will mean that the user will have to enter their email and

password before they can use the application again. Once a user logs out their session will be destroyed.

4.2.1.3. Non-functional Requirements

- **Reliability:** The system will be reliable in the sense that all the information that will be provided will be accurate and will always be current. The database for the system is going to be updated when changes occur. Updates and reminders will be sent at the end of a transaction to the users involved. The system is going to be available to the public 24 hours a day and 7 days a week
- **Performance:** Performance of the client system depends on a number of factors. Some of them being the network speed, bandwidth and speed of the device being used. Performance may vary from system to system based on the factors mentioned above. On the web server the system will be built so that database accesses will be fast and error control will be very effective.
- **Usability:** The system is going to have a fairly simple graphical user interface so that it will be easy for its users to use. The system is going to look at other existing systems and pick the common terms that are used to describe and label functions. This will help users relate better to the system since they will be familiar with these terms and will know exactly what functionalities they perform.
- **Multi-platform:** The system is going to work on any platform i.e. operating system provided there is a web browser present. It will also

be able to work on mobile phones that have the capability of browsing the internet.

4.2.2. Design

4.2.2.1. Interface

Designing the interface has to do with designing the parts of the application that the user comes into contact with physically. The interface design is very important because it determines whether or not the user stay's on the website or not.

In coming up with the design, Jakob Nielsen's 'Ten Usability Heuristics' [4] was consulted for guidelines. Some of the usability heuristics that was considered in the design include

- Aesthetics and Minimalist Design: Jakob states that "Dialogues should not contain information which is irrelevant or rarely needed". In the design, labeling of the dialogues are kept simple and straight to the point. The information being asked from users are short and concise.

Pay-Inn

| My Account | Send Money | Request Money |

Before you start this transaction, check if

- Your Recipient has a valid Pay-Inn Account
- Your Recipient has a valid Bank Account

Fields Marked with * are required.

Email*

Email Address of the recipient

Amount*
GHC
Amount of money to send

PIN*

Your PIN will be used to verify the transaction

Re-Enter PIN*


Enter the value in the image above

Send Money

Figure 1 Aesthetics

- **Help users recognize, diagnose, and recover from errors:** For the web application, there is an AJAX mechanism to inform the user when he enters the right or wrong data. When wrong data is entered, a notification is put by the side of the input field. If the user does not see and requests the page to be submitted, the errors are made visible by changing their color. The error messages are in plain and simple language stating the error and suggesting a solution. See Figure 2.

Fields Marked with * are required.

Email*
kad Not a valid email
Email Address of the recipient

Amount*
GHC ▾ tr Numbers Only
Amount of money to send

PIN*
 Your PIN should be a four digit NUMBER
Your PIN will be used to verify the transaction

Re-Enter PIN*
 Invalid Input

Figure 2 Error Reporting

- **Visibility of system status:** The system tells the user whether or not he is logged in or not. See Figure 3

Pay-Inn

| [My Account](#) | [Send Money](#) | [Request Money](#) |

Logged in as: Kofi

Figure 3 System status

If the user has not added a bank account, a notification is displayed. If the user has a bank account added, part of the account number is displayed. See Figure 4.

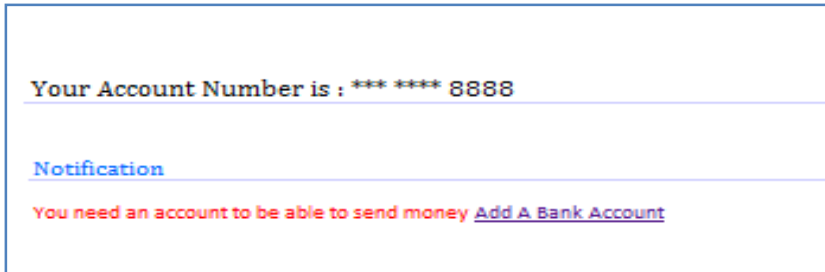


Figure 4 Notifications

- **User control and freedom:** The main menu is always visible to users on each page that they are on. A user has the freedom to access any part of the menu no matter what page they are on. See Figure 5.

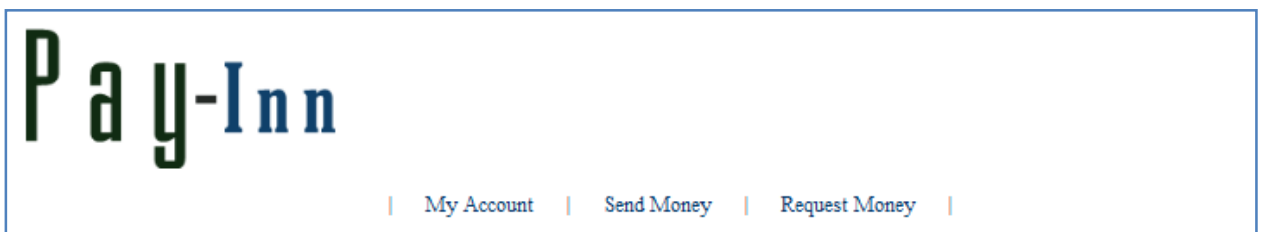


Figure 5 Main m1enu

- **Consistency and standards:** The colors used on a page need to be consistent. Error messages are in red. Notifications are displayed in blue. Menu options are displayed in a lighter shade of blue and on hover the colors change. The sub-menu options are displayed in button like containers with the same colors. Labels are displayed in black. The page layout is consistent across all the pages with buttons located in the same place.
- **Error Prevention:** Some input fields are constrained to the number of digits that they can take. For example the PIN input fields are limited to taking only four digits. The bank account input field on the 'Add/Change Bank' page is limited to eleven digits which specify the number of digits that the account number should be.

Most of the designing was first done with a pen and paper. The main focus of the design was on the layout of the page and how things were going to be positioned on the page. The designing did not take a lot of time as some queues were taken from the PayPal website. The design is a very simple one with the layout clear to navigate. Different colors are used to distinguish one group of links to the other. There are also different colors used for displaying messages. Red colored texts are mainly for errors and blue colored texts are notification or confirmation messages.

The general outline of how the pages connect to each was also drawn up. During implementation some of the pages were implemented using AJAX so some of them do not have dedicated pages to them. The page flow diagram of how the pages connect to each other is below.

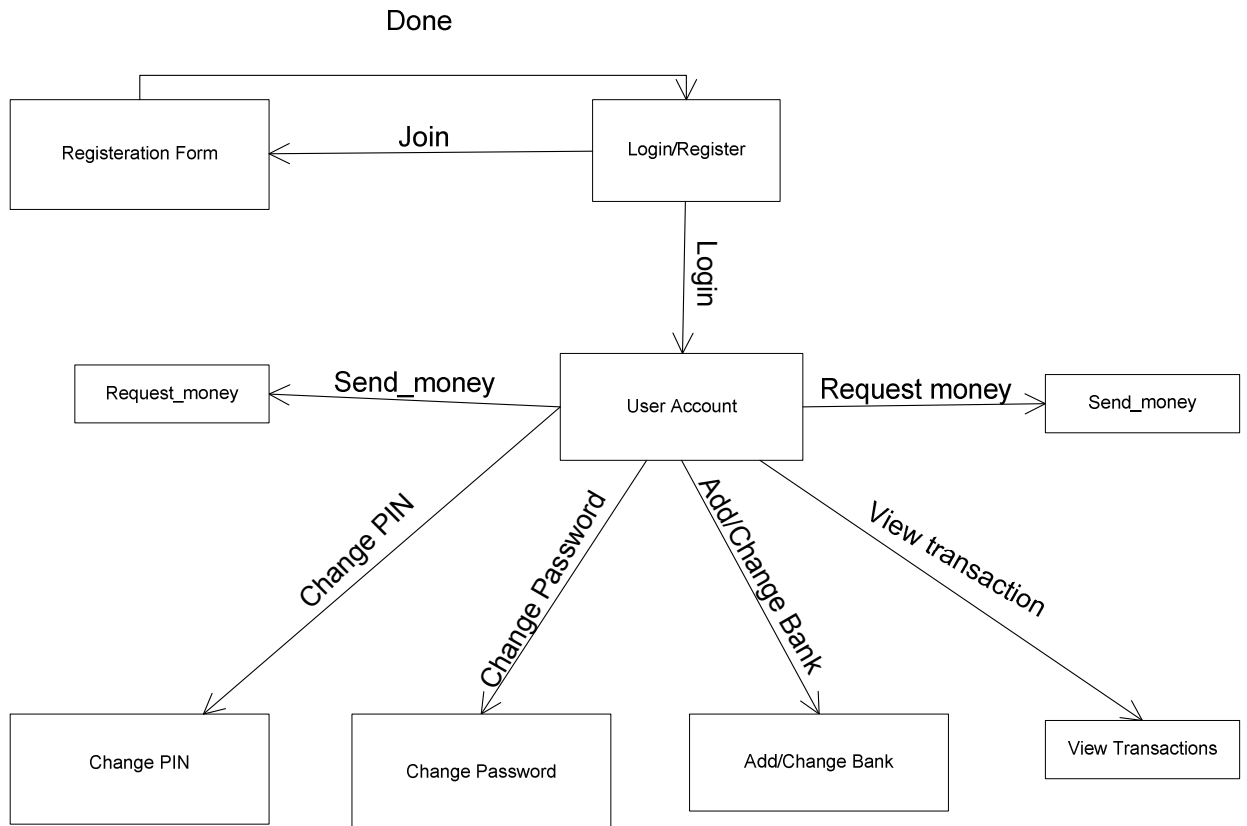


Figure 6 Page Flow Diagram

4.2.2.2. Database

The database is another part of the system that has to be designed. The design is needed to make the implementation of database easy. Without the proper design the designer is going to run into problems implementing the database.

Most of the designing was done with pen and paper. The tables are the first to be created. The next thing to do is to fill the tables with the necessary attributes then make sure that the table is normalized with the primary keys and foreign keys and candidate keys identified. After that the relationships

among the tables need to be drawn up to match how the system is going to work.

The application has two main databases and each of them has a number of tables in them.

- Banks – This database has two tables

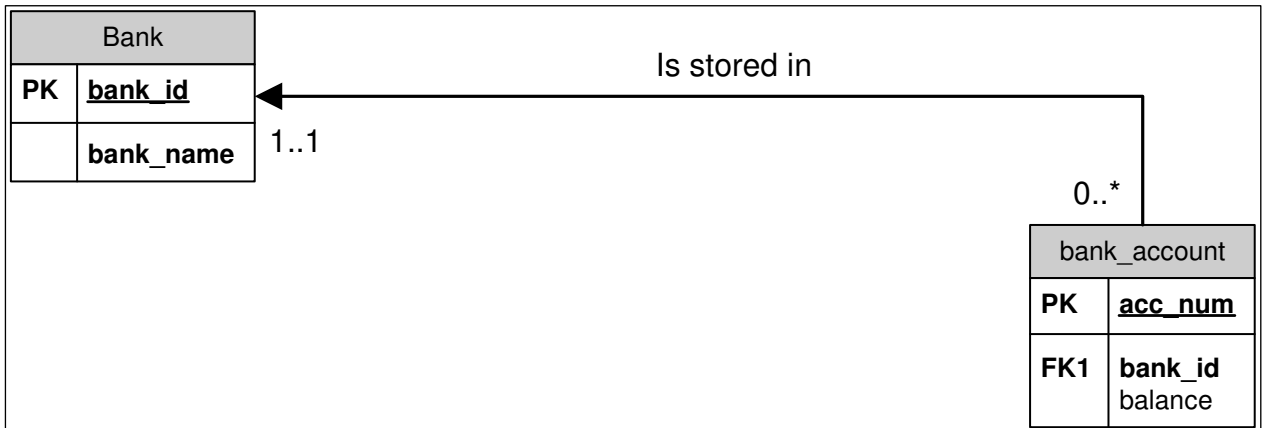


Figure 7 Bank Database schema

- bank – This table contains data on the different banks
- bank_account – This table contains data on the different bank accounts that the different banks have. It contains the account number, balance and the bank id.

- Pay-Inn – This database has 4 tables

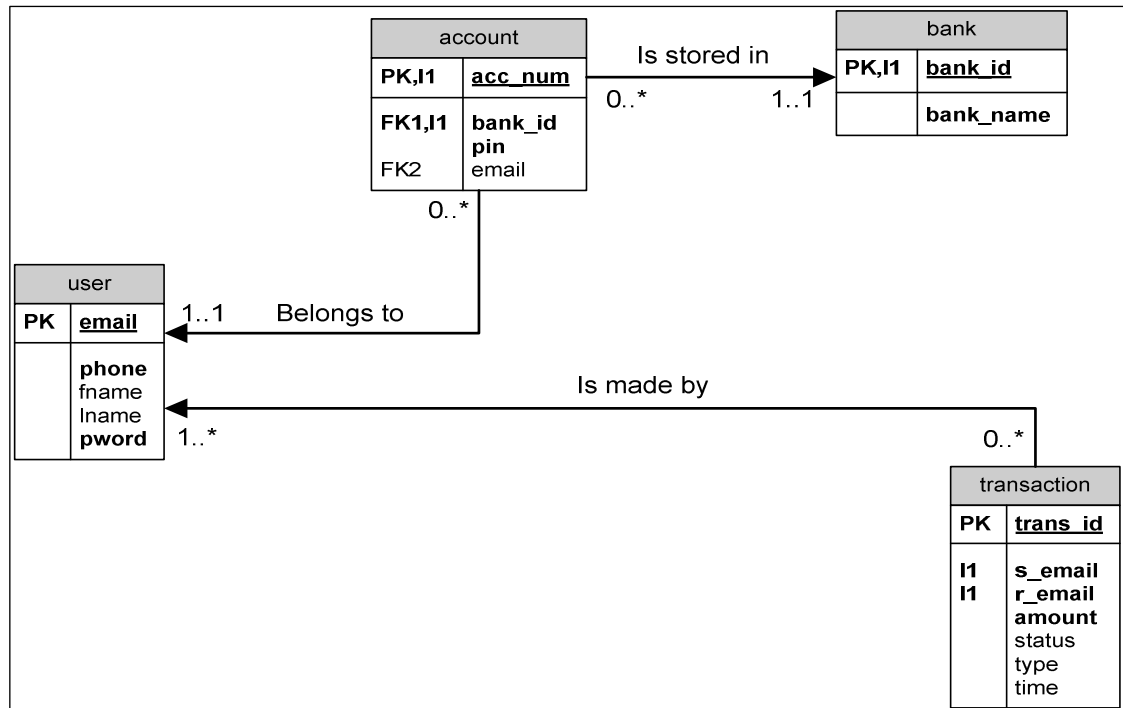


Figure 8 Payinn Database schema

Account – holds the account details (email, acc_num, bank_id, PIN) of each registered user with a bank account

- Bank – this contains the bank information
- Transaction – a table to store the transactions that users make
- Users – a table to hold the information on registered users

4.2.2.3. Classes

Classes are created to divide the code of the application into different objects. These objects have functions or methods that are called in different

parts of the application to provide some service. Different classes were designed to cater for each module of the application. A class was designed for the user, accounts, bank and database. See figure 9.

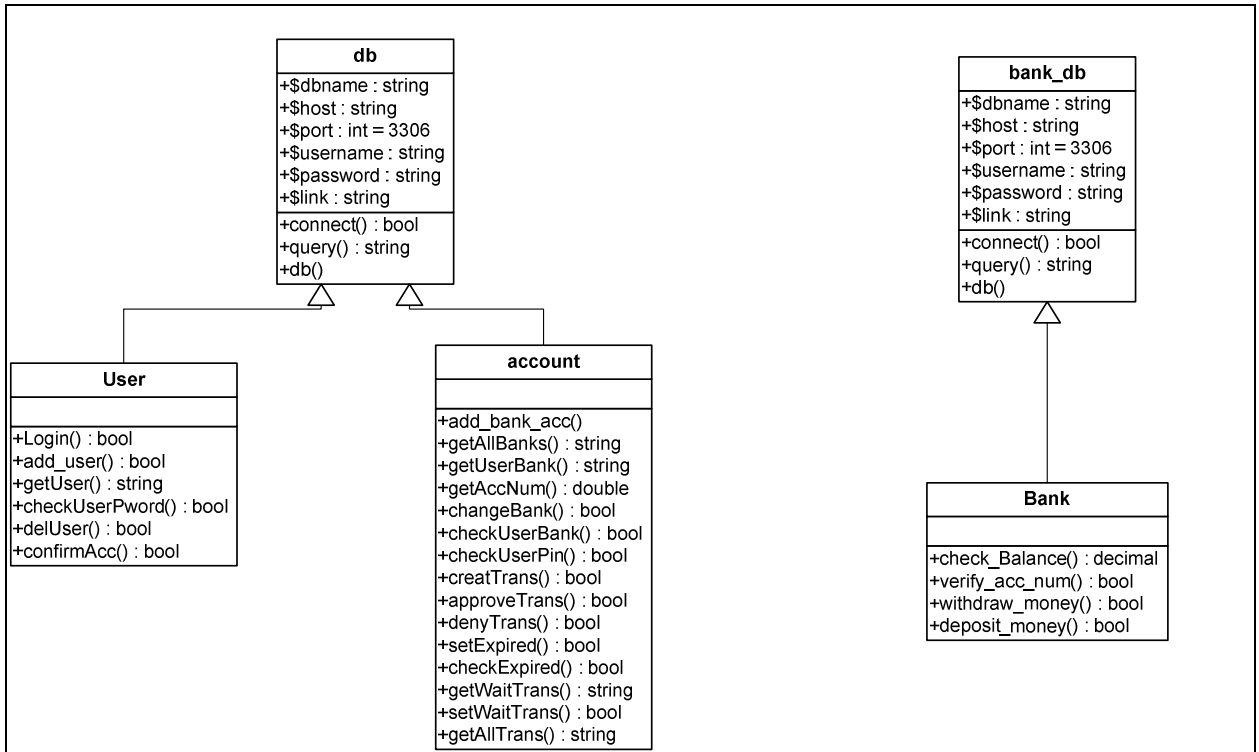


Figure 9 Class Diagram

The db class is responsible for connecting to the database and querying the database for data.

Both the user and account classes extend the db. By extending the db class the user and account classes can send queries using their functions to the database to send and retrieve data.

The user class represents the user object and therefore does operations related to the user for example registering and login.

Account class is responsible for the users account on the application. It adds and changes bank a user's bank account. It's also used to log and view the transactions that a user makes in relation to his account.

The bank class deals with the external bank account and connects to the bank database to retrieve the users account number and the users account balance. This class extends the bank_db class which does all the connections to the bank database.

4.2.3. Development

4.2.3.1. Tools

- Adobe Dreamweaver – used to write the main code
- Adobe Photoshop – used to design some banners and the logo
- xampp – the sql server and web server are installed here
- PHPMyAdmin – used to manage the database

4.2.3.2. Developed Parts of the Application

- Database: The database is used to store information on users, banks, bank accounts, transactions and other important information used in the application.
- User Registration: This module is important for adding new users to the application. With this page the user can register to use the application on both the PC and the mobile phone.
- User Login: This module allows the user to login into the system and to do transactions. It works on both PC and mobile

- Adding a Bank: Registered users will need to add banks to their profile in order to do any transactions. This feature is limited to PC only. See
- Send Money: this is the heart of the application. Without it the application loses its purpose. It allows users to send money to other registered users on both PC and mobile.
- Request Money: an additional feature that allows users to request for money from other users. This feature is available on both PC and mobile.
- View transactions: users will be able to view their old transactions to know who they sent money to. This can be used as an account statement. This feature is limited to the PC version of the application.
- Change bank: users who have bank accounts already added to the application can change their bank accounts. This module will become useful if a user needs to do a transaction but has no money in his default account. He can change it to some other bank account that has money in it. Only on the PC version.
- Notifications page: this page is used to display notifications about new transactions and old ones as well.
- Validation: Validation is done on the server side with the use of AJAX. The data entered into fields are validated as it is entered into the fields. Incorrect data is not submitted to the database.

5. Testing and Results

Through the development process various tests were done to check whether or not the application works. At each point of the development phase there were different things that were being tested. Below are some of the test that were conducted and the results they produced.

- Connecting to the database: The system has most of the data needed for it to work stored in the database therefore it is very critical the database always and it is much more critical to make sure that the queries being executed are the correct ones.

There are different modules of the application that have to connect to the database to read and write data. Testing of these modules was done to check whether they were connecting to the database and reading or writing the correct data from and into it respectively.

- Reasons:
 - To make sure that the database is working
 - To make sure that the application was connecting to the database
 - To make sure that the queries being sent were correct
- Results: Generally the database was working and application was connecting to it perfectly. The problem was with the queries that were being sent. There were errors in the queries being sent to the database therefore the desired results were not being achieved. After the queries were corrected the application

had no problem writing and reading to and from the database respectively.

- Data Validation: During user registration, the user is required to enter some data. As the data is being entered, the AJAX on the page is supposed to read that data send it to the server and make sure the data entered is valid. The test was done for different types of data ranging from email addresses, phone numbers and names.

Validation on the mobile side is done partly on the mobile and partly on the server. Things like PIN validation and password validation are done on the server. Comparing passwords the have been entered match is done on the mobile phone. Checking for whether all necessary fields are filled is also done on the mobile. The validity of the data entered is checked on the server and the appropriate messages are sent back to the phone.

- Reasons

- To check whether the validation code was working
- To check whether the AJAX is working on both mobile and PC
- To check whether the right data is being validated.

- Results

- AJAX was working perfect with no errors on PC
- Two types of mobiles were used to test the mobile web application. An iPhone and a Sony Ericsson S500; the iPhone run the Java with no problems but the Sony

Ericsson could not support it. The possible reason could be that it does not support AJAX or JavaScript. This is an error that has not been rectified.

- At a point in the testing, the correct data was being sent but to a different page. Therefore the results were not correct. This occurred because the path to the desired page was not correct. Once the path was changed, it started working fine.
- A few errors occurred with the validation code. The email validation code had some errors in it and so did the password validation. The problem occurred in the `ereg()` function that was being used to validate the data. After a few modifications of the code, it started working perfectly.
- CAPTCHA: the code that used was gotten from an external source [5]. The details of the owner are stated in the PHP files. Understanding it was not too difficult since it was straight forward and it had examples that could be followed.

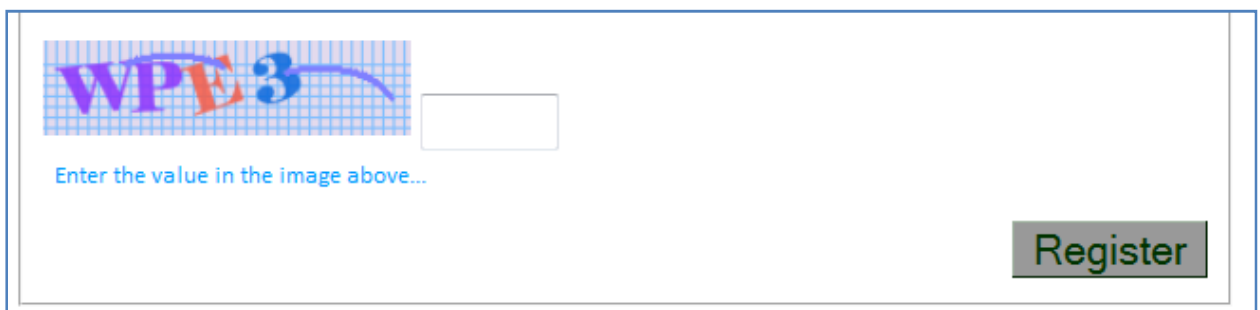


Figure 10 CAPTCHA image

- Reasons
 - CAPTCHA is used to verify that it's a human running the page and not a bug or a virus.
- Results: CAPTCHA worked fine the first time it was implemented. The main problem that was encountered was mainly due to the fact the code was borrowed. An understanding of the code cleared all the problems that occurred. For CAPTCHA to work, GD library in PHP has to be installed or enabled. The library allows for the image to be created and displayed. The testing server had the library installed and CAPTCHA worked fine.
- Email: whenever a user completes a registration process, an email is sent to confirm the email address they provided. This is done when a user registers by mobile or PC. To test this, a local email server was setup on the web server and dummy email accounts were set up. Users were registered using the dummy accounts and the emails were sent successfully.
- User Registration: User registration testing was done gradually as different modules were added. First phase was connecting to the database. Then validation of data, then CAPTCHA and then email. All these came with their problems but were solved accordingly. See Appendix: Figure 12.

- Adding a bank account: Adding a bank account was not easy. This is because this part of the application was connecting to two different databases at the same time. The main thing being tested were
 - Validating a user account number: Before a user can add an account, the bank account is verified with his specified bank. If the bank account exists, the account is added if not the transaction is denied.
- Sending money: This module was also connecting to two databases at the same time. The main parts of this module
 - Validation: the user's bank account and the user to which the money is going to be transferred bank account will need to be validated. The balance of the senders account will have to be checked for enough funds for the transaction. PIN validation also occurs to make sure the user is the authorized user of the account.
 - Notification: on successful transfer of funds, a notification is sent to the recipient. The notification appears on the home page of the recipient and the user is given the opportunity to accept or deny the transaction.
 - Deposit and Withdrawal of Money: when a user sends money, the amount should be deducted from his balance in the bank. In the case where the transaction could not be completed the money will be deposited back into the sender's account. When a user receives money and accepts the transaction, the amount is

deposited into the user's account. If the user denies the transaction, the money is deposited into the senders account.

- Transaction timeline: each money transfer transaction takes has a validity period of seven (7) days. After the validity period if the recipient of the money does not respond to the notification, the transaction will expire and the money will be deposited into the senders account and a notification sent.

6. Discussion

This application is a not so much of a new application being developed for the Ghanaian and African market but it comes with a new approach that allows users to tie the application directly to their bank account which means that any transaction they make is reflected directly in their bank account.

6.1. Approach

The approach taken to build this application and the solve the problem of people going to the bank whenever they needed money proved to work at what would be termed a testing level. The main focus of Pay-Inn was to move money from one account to another. This is done basically between the two databases. If this system was being built to suite the real happenings in the banking sector, there will be far more stringent procedures to follow. Pay-Inn in its current state is not a harmful because it does not touch real money. It only deals with perceived money and perceived bank accounts.

Security is key is such a system and thus must not be overlooked. Pay-Inn was not built to tackle the security aspect of such a system therefore the

security implementations are minimal. Without proper security, people's data may be at risk of being modified, stolen or deleted from the records. For such a system, an occurrence like this may cost someone a large amount of money. The system should be secure such that

- Data that is stored in the databases are encrypted with AES.
- The application should be deployed over a secured that has all the data passing through it encrypted and the key kept safe.
- The keys used for encryption should be changed frequently.
- The physical location of the servers that deploy the application should be in a place that is fully protected. The server room should always be under lock and key and should be accessible to only authorized personnel. The server room should be such that it is difficult for natural disasters that occur to touch the servers.
- Back up is very important in a system like this. It always good practice to have a backup server in a location far from where your main servers are located and also to have more than one backup server. In case of server breakdown the servers can be switched automatically and there will be not data loss. A system like this requires frequent backing up in order not to lose any information on any transaction.
- Since data can be lost due to network failures and other things, it is always necessary to make sure that the data sent is always correct. One way to use this is by implementing checksums or by hashing the data using a particular function like `'md5()'`. The data that is transmitted is hashed and then added to the raw data. The combined

data is encrypted and sent to the server. Once the server receives the data, it hashes the raw data and matches it to the hashed data. If they are the same then the data is correct and the data is processed otherwise the server will request for the data to be retransmitted.

Pay-Inn does not cater for all the security measures above. It deals with security mainly on the user side of the application which is also very important. With all the security features mentioned above implemented, it will be useless if user passwords can be figured out easily or details of a users account can be obtained without difficulty. Pay-Inn mainly handles security by

- Making sure that a user's password meets the minimum criteria.
- Users have PIN that they use for transactions
- Personal user information is not transmitted to other users except their email addresses
- CAPTCHA is used to make sure that it is a human accessing the system.
- Data entered is always validated before it is stored into the database.
- User bank accounts are always validated and checked for the necessary balance required to complete a transaction.
- Running the application on a secure network.

7. Limitations & Conclusion

The limitations that were faced during development were

- A suitable banking interface to use for the application: The main plan for development of the system was to build the application to suite a real banking environment. This was not achieved because a suitable banking interface to connect to was not available. Through research the ISO 8583 standard which is used to transfer information between ATM and the bank was discovered. The PHP API for the standard was found but was of little help since its operation could not be fully figured out. Research to find a more in depth API proved futile. The idea was therefore changed to use a pseudo banking environment developed using different databases. One for Pay-Inn and the other for the banks. If the ISO 8583 standard was used it would have given a true banking feel to the application. It would have represented a banking environment and that would have fulfilled the main plan of the project. The ISO 8583 standard provides a more secure way of transmitting data from one server to another. The data transmitted is divided into three different parts [6].
 - The Message Type Identifier: this is a four digit number that identifies the type of transaction being made. The first two digits identify the type of transaction. For example 02xx specifies a financial transaction [6].
 - The Bitmap: The bitmap is used to specify which data elements are available in a message. The first set of bitmaps range from 1 – 64 and are represented in 8 bytes of binary data or 16 hexadecimal characters. A data element is present if the bit

representing it is true or 1. In a 8 byte binary number 10010010 and counting from the last digit and starting from zero the data elements 1, 4 and 7 are present in this bitmap. These data elements have specific data types that they store and transmit [6].

- The Data Element: The data element is used to transmit information across the servers. It holds the main transaction data. The receiving server processes the message and sends a response in the same format [6].
- Introduction to mobile programming was a late in relation to this project. Therefore coding in J2ME was a new thing and needed much practice to get used to. With the knowledge of how Java code runs and the use of the API it was not difficult to figure out how to get some things done. The J2ME mobile application has been developed to the level which allows user registration, login, send money and request money. These are working but with some errors along the line. Being a novice at J2ME it was quite difficult to figure out how some functions performed. A lot of testing and redevelopment was done compared to the web application most of them running into the same after corrections were made. These cost a lot of time as there were other parts of web application that could have been finished in that time. Nevertheless, the problems that were being faced caused the learning of new ways of doing things and bringing the understanding that the code will work they one writes it. Instead of the means that seemed

logical but did not work, other means were developed to do slightly the same job with a few modifications but all at the expense of time. An example being that when data was sent from the server to the application, there was some difficulty trying to read the data and work with them so the approach taken was to return different data with different lengths and then instead of reading the content of the data the length is read and used for something. This worked fine for all of the pages that requested data from the server. Another challenge with J2ME is that it takes a lot of coding to get a single thing done unlike Python where a relatively few lines will do the same thing. Validation was also a problem as most of the major validation needed to be done on the server and not on the phone thus increasing the amount of time a user would spend doing a transaction if the connection was slow. And forcing the user to re-enter data.

Overall the time spent to develop this application has been worthwhile. Most of the requirements that were drawn up have been achieved. From the tests that have been conducted so far, these are the modules that work,

- The application works on both PC and Mobile.
- Users can add bank accounts to their profile
- Money can be transferred between accounts successfully.
- Requests for money can be made successfully
- Users get a full list of all the transactions that they have made
- User registration is working on both Mobile and PC
- Email confirmation.

- Data validation
- User login
- Changing a bank account
- CAPTCHA

These are the parts that do not work

- Adding a bank on mobile
- Data validation on mobile
- Changing password and PIN on both Mobile and PC
- Forgotten password on both mobile and PC

This application if developed well with the security features can be a very big and profitable one. The mobile phone is just evolving into a tool with many uses and this application is one of many that will utilize that power. The use of the mobile phone is widespread and as time goes on most developers will be focused on developing more for the mobile than the PC because with the mobile, users can work from anywhere and anytime.

Works Cited

- [1] MTN. (2010, April) MTN Website. [Online].
<http://www.mtn.com/ProductsServices/MTNMobileMoney/>
- [2] Zachary Ochieng, "Mobile Money Transfer Service Gains Popularity with the Both the Banked and Unbanked Populations.," *CIO East Africa Magazine*.
- [3] PayPal. PayPal Website. [Online]. <http://www.paypal.com/cgi-bin/webscr?cmd=p/gen/about-outside>
- [4] Jakob Nielsen. (2005) Use It. [Online].
http://www.useit.com/papers/heuristic/heuristic_list.html
- [5] Drew Phillips. (2007) Securimage website. [Online].
http://www.phpcaptcha.org/Securimage_Docs/
- [6] Trusted Security Solutions, "ISO8583 Message Format Specification," vol. 3, pp. 1 - 11, Sept 2005.
- [7] Oracle. Oracle. [Online].
http://www.oracle.com/database/db_manageability.html?origref=http://www.oracle.com/us/products/enterprise-manager/index.html
- [8] Safaricom. Safaricom. [Online].
<http://www.safaricom.co.ke/index.php?id=745>

Appendix

Screen shots



Join For Free | Log in | About Pay-Inn

Pay-Inn

Sign on

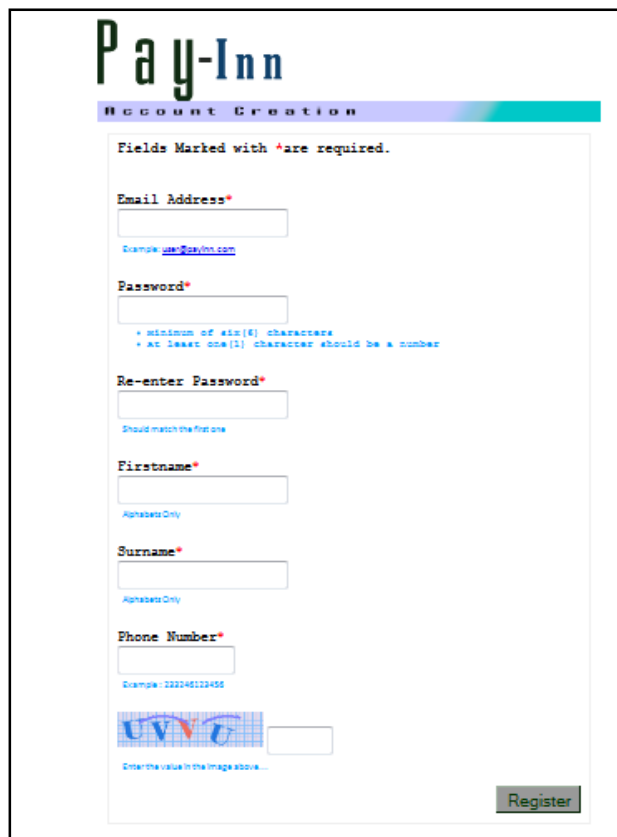
Email

Password

[I have forgotten my Password](#)

[Open an account today](#)

Figure 11 Login Page



Pay-Inn

Account Creation

Fields Marked with * are required.

Email Address*

Example: uvt@payinn.com

Password*

* minimum of six(6) characters
* At least one(1) character should be a number

Re-enter Password*

Should match the first one

Firstname*

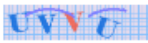
Alphabets Only

Surname*

Alphabets Only

Phone Number*

Example: 222246123456



Enter the value in the image above...

Figure 12 Registration Form

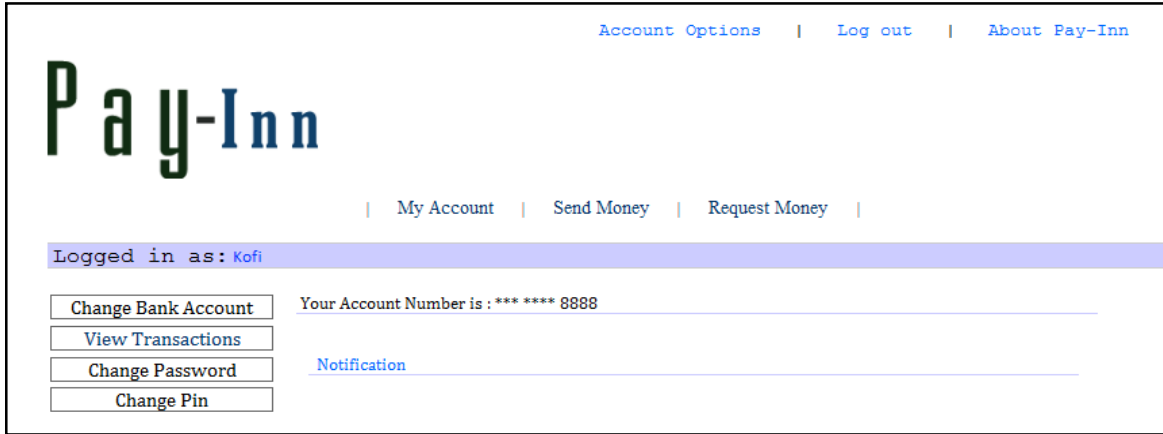


Figure 13 User Home Page

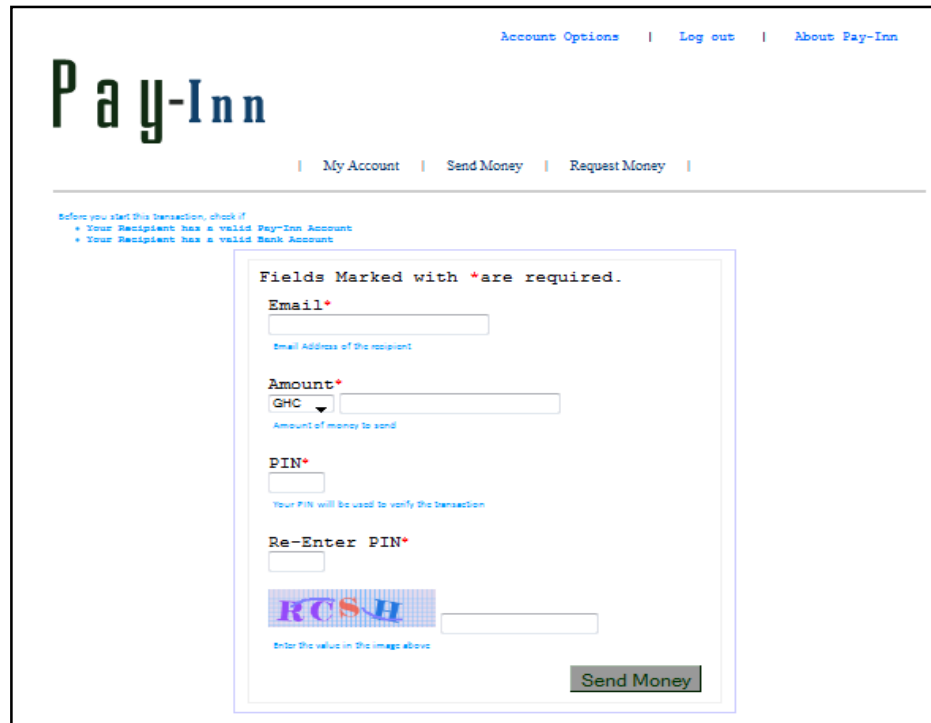


Figure 14 Send Money

[Account Options](#) | [Log out](#) | [About Pay-Inn](#)

Pay-Inn

[My Account](#) | [Send Money](#) | [Request Money](#)

[All Transactions](#) | [Payments Sent](#) | [Payments Received](#) | [Requests Sent](#) | [Requests Received](#)

Date	Sender	Recipient	Amount (GhC)	Status	Type
20 Mar, 2010	we@y.com	wakhesse@ashesi.edu.gh	50087.00	ACCEPT	SEND
20 Mar, 2010	wakhesse@ashesi.edu.gh	we@y.com	200.00	ACCEPT	SEND
20 Mar, 2010	wakhesse@ashesi.edu.gh	we@y.com	12090.00	ACCEPT	SEND
20 Mar, 2010	wakhesse@ashesi.edu.gh	wilhess18@yahoo.com	1500.00	DENY	SEND
21 Mar, 2010	wakhesse@ashesi.edu.gh	wilhess18@yahoo.com	1200.00	ACCEPT	SEND
23 Mar, 2010	wilhess18@yahoo.com	wakhesse@ashesi.edu.gh	1000.00	ACCEPT	SEND
25 Mar, 2010	wakhesse@ashesi.edu.gh	damola@yahoo.com	1200.00	WAIT	SEND

Figure 15 View Transactions

[Account Options](#) | [Log out](#) | [About Pay-Inn](#)

Pay-Inn

[My Account](#) | [Send Money](#) | [Request Money](#)

Logged in as: [Kofi](#)

[Change Bank Account](#)

[View Transactions](#)

[Change Password](#)

[Change Pin](#)

Current PIN

New PIN

Re-Enter PIN

Figure 16 Change PIN

Buttons: Add A Bank Account, View Transactions, Change Password, Change Pin

Fields Marked with *are required.

Account No.*

Eleven Digit Number

Bank Name*

Create PIN*

A four(4) digit number that you will use to verify you transactions.

Re-Enter PIN*

Add Bank

Figure 17 Add a Bank

Mobile Screenshots



Figure 18 Home Page J2ME

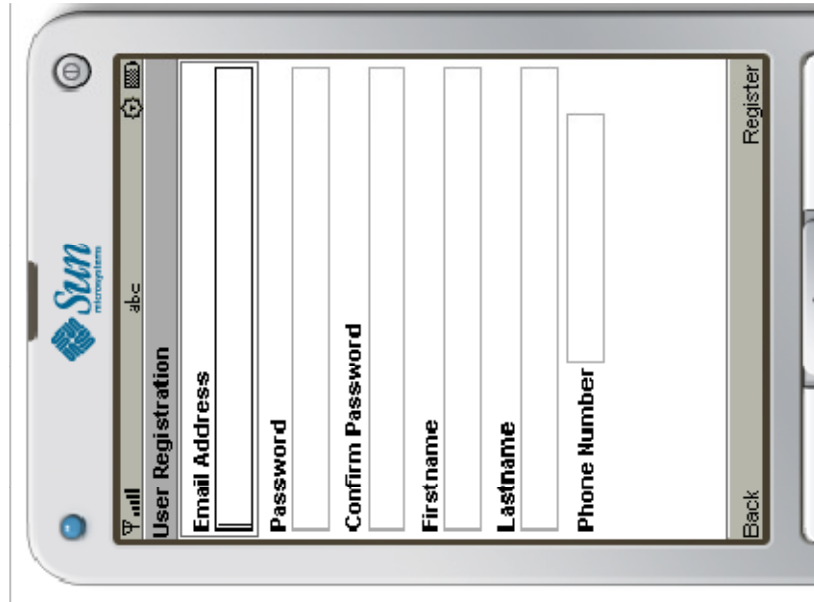


Figure 19 User Registration J2ME

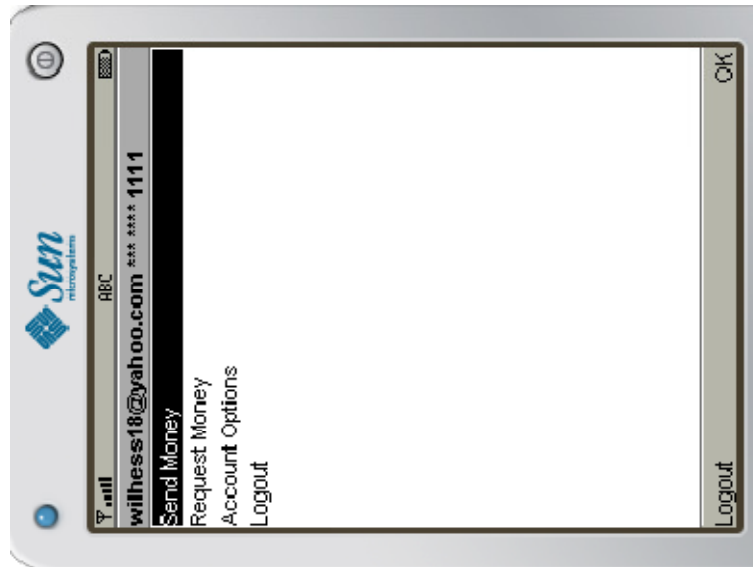


Figure 20 User Account J2ME

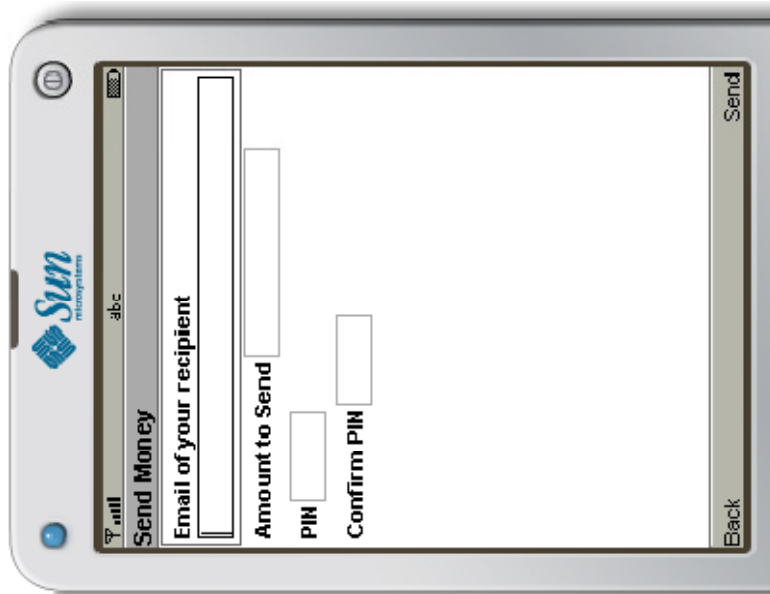


Figure 21 Send Money J2ME