|  |  |
| --- | --- |
|  | **2014** |
|  |  |

|  |
| --- |
| **E-MoO - Online payment system for developing world** |
| This is part of a collaboration initiative for the development of ICT solutions with the involvement of Students |

**eMoO**

# Disclaimer

THIS DOCUMENT AND THE INFORMATION IN IT ARE PROVIDED IN CONFIDENCE, FOR THE SOLE PURPOSE OF THE **e-MoO project**, AND MAY NOT BE DISCLOSED TO ANY THIRD PARTY OR USED FOR ANY OTHER PURPOSE WITHOUT THE EXPRESS WRITTEN PERMISSION OF **DAYARATHNA RASIKA** AND **ENEAS HUNGUANA**.

**Document History**

|  |  |
| --- | --- |
| Version: | 1.1 |
| Last Edit Date: | 27-May-2014 |
| Last Edited By: | Eneas Hunguana <eneas.hunguana@iwconsultants.net> |
| Replaces Version | 1.0 |

# Table of Contents

[Disclaimer 2](#_Toc388992213)

[Table of Contents 3](#_Toc388992214)

[1. Background 4](#_Toc388992215)

[2. Online Payment System for Developing Countries 5](#_Toc388992216)

[2.1 Definitions 5](#_Toc388992217)

[2.2 System Requirements 6](#_Toc388992218)

[2.3 Solution Overview 7](#_Toc388992219)

[3. Project Goals and Objectives 12](#_Toc388992220)

[4. Methodology 13](#_Toc388992221)

[5. Work Plan 14](#_Toc388992222)

[6. Project Team 15](#_Toc388992223)

[7. REFERENCES 17](#_Toc388992224)

# Background

Internet has become a platform where service providers and buyers trade in services and goods. To process the transactions, online shops resort to electronic commerce (also known as e-commerce) tools. One of the key-component of the e-commerce system is the **electronic funds transfer (EFT)**, which is basically *“the electronic exchange, transfer of money from one account to another, either within a single financial institution or across multiple institutions, through computer-based systems”*[1].

There are several companies providing EFT services for e-commerce sites. The role of these companies, also known as payment gateways, is to provide for *“for electronic payments and collection”*[2]. While several payment gateways provide mature solutions for online payment, most of them require the association of a bank card to the system. This automatically prevents the group of people without bank accounts from buying goods or services online, even if they have the financial resources to purchase the products. In another perspective, due to lack of trust, some people bank cardholder feel reluctant to associate their bank cards to online payment gateways.

It was with the intention of addressing these challenges that in 2003 kasun De Zoysa and Rasika Dayarathna from University of Colombo in Sri Lanka, proposed the eMoney Order System[3], a new payment system for the Sri Lanka post office. The system, meant to be an enhancement to the existing solution by the time, was designed to fit in the context of developing countries, where credit cards and other formal *cashless* payment methods are not widely adopted by the population, due the several reasons, including affordability. In 2008, another partnership between Rasika Dayarathna and Eneas Hunguana, an ICT professional from Mozambique, resulted in the conceptualization of another cashless Micro-Payment solution based on SMS technology [4]. In last years, Mobile telephony operators came up with payment systems that do not require bank cards. However, these systems are owned the operators, or have strong dependency on integration with operators’ core network systems, which always requires complex agreements with the mobile telecom operators.

This document presents a new cooperation project, aimed at developing the next generation of E-Money Order System, meant to be an open source platform, developed based on the combination of the concepts from the previous works mentioned above. The current project will be carried out in the format of a problem-based learning initiative, driven by projects. The resolution of the selected problem requires the understanding and use of concepts and technologies related to different disciplines of Computer Science curricula. Therefore, this constitutes an opportunity for the students to gain/widen/consolidate their practical skills, by solving a real-world problem.

The overview of the intended solution is presented in section 2.

# Online Payment System for Developing Countries

This section presents the high level description of the concept behind the intended online payment system. Because this project is mainly designed to be a learning framework for the team of students that will work on it, this should no be considered the final design. Instead, students are invited to do a critical analysis of the proposed design, identify and propose fixes to the design flaws, as well as any improvements, taking into account the practicality of the proposed solution. For a better understanding, the solution overview section is preceded by the sections presenting the definitions to be used along the document, as well as the system requirements.

## Definitions

The following terms will be used in this document:

eMoney Order Account – A virtual money account hosted in the online payment platform;

Remitter – The person who deposits physical money in an *agent*, in order to load an E-Money.

Agent – An entity (physical shop) that receives cash from the *Remitter* and credits it to an e-money account.

Beneficiary – The money receiver. It’s a person to whom the funds transferred by the *Remitter*.

e-Payer – This is the *beneficiary* when paying for online products/services using eMoney system.

Merchant – An entity that sells products/services online, and accepts payments via E-Money order system.

Mobile Operator – The owner/operator of cellular telephony network.

e-Voucher– An alphanumeric code equivalent to a specific amount of physical Money;

## System Requirements

Below are presented the different requirements that the system must comply with:

* + 1. Functionality

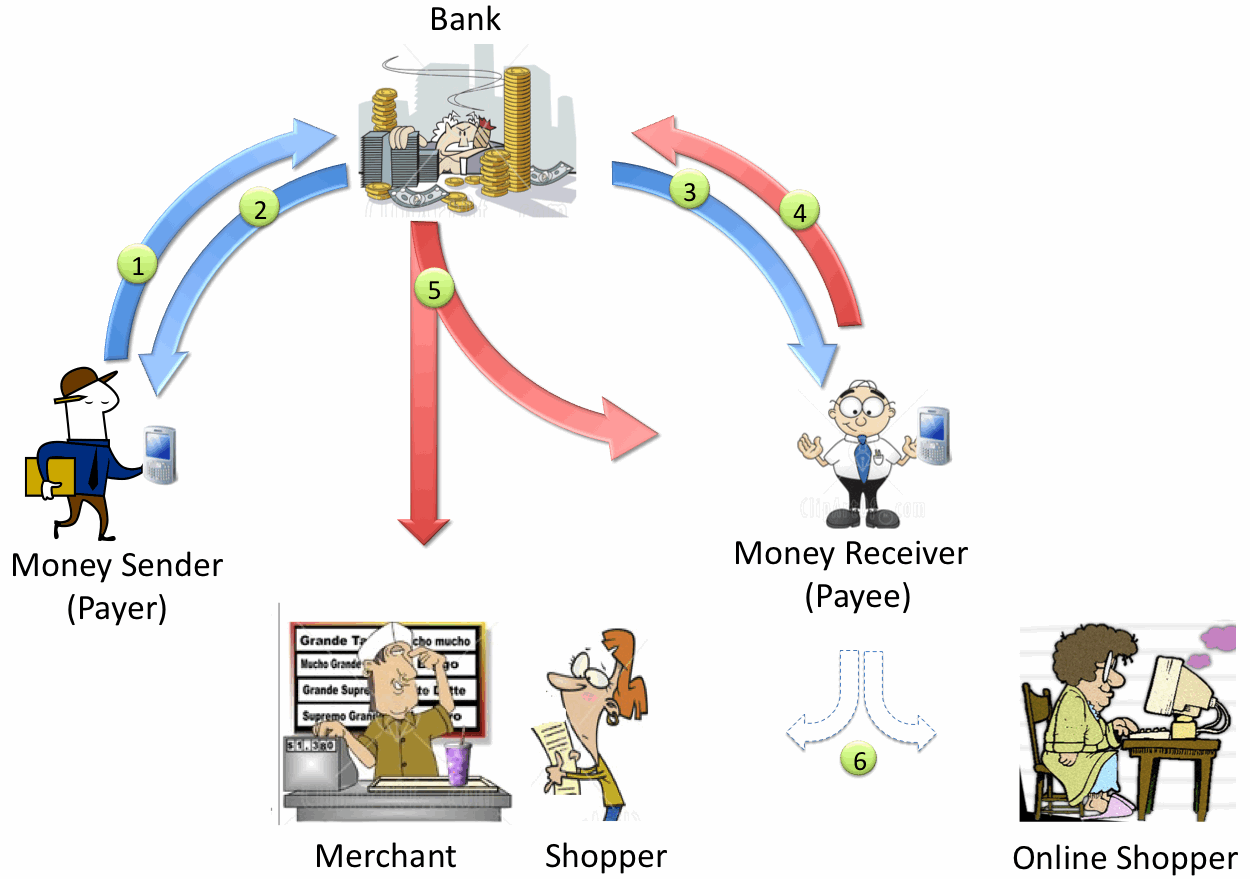
1. Online payment without the need of a bank card;
2. Funds transfer to/from individuals without bank accounts;
3. On-the-fly creation of *beneficiary* accounts (without requiring user to register);
4. Manual creation of *agents* and *merchants* accounts;
   * 1. Usability
5. Account top-up and balance enquiry (for end-user);
6. Account balance enquiry for service providers (merchants who use the system as a payment platform);
7. Transaction records storage;
8. Instant notifications (both for *Remitter* and *beneficiary*) upon the realization of a transaction (successful or not);
9. Must allow payment Order requests via SMS or Internet;
10. …
    * 1. Customer (project Owner)
11. Run on top of open source Operating System Linux;
12. Built with free and open source tools;
13. The interface for communication with *agents* must be web-based, therefore not requiring the installation of third-party applications;
14. The API for communication with*merchants* must use standard protocols (e.g., HTTP);
15. Support for any type of phone handset;
16. …

## Solution Overview

The proposed payment system builds on the concept of the previous works [3,4] on the matter. For a better understanding of the evolution, a brief explanation is offered below.

* + 1. Evolution from the previous works

In the previous works, banks and/or post offices played the role of collecting the physical money, verify users information and generating the eMoney order codes, worth a specific amount of money meant to pay for pre-defined services or to transfer money between individuals.



**Figure 2.1: Micro-payment System for underprivileged Communities [4]**

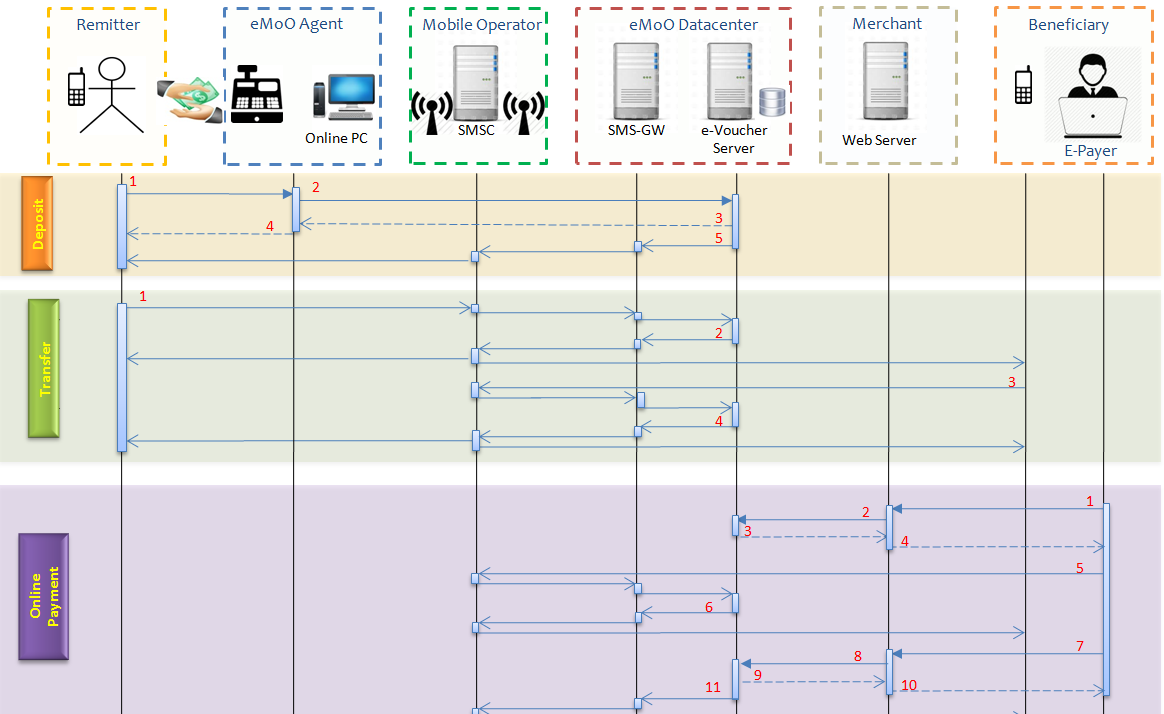
The new system will focus on the concept of *electronic money order accounts*, and dynamic generation of *payment order codes* with variable amount.

In the proposed system, any pre-registered *agent* (e.g. groceries shop) will be able to collect money and credit the money into an eMoney order account as long as it is connected to the Internet. Additionally, the eMoney order code will only be generated at the moment that the *e-payer* requests and payment order, for any amount, as long as covered by the available funds (and eventually, service fees). This flexibility expands the types of products and services take this system as a form of payment.

The architecture that makes it possible is presented in the next section.

* + 1. eMoO Solution

Figure 2.2 illustrates the proposed system architecture. The call flaw presented below is a simplified illustration of the interaction among the system components. The focus is on eMoO operations; therefore, details of the intermediate events taking place on the mobile operator are omitted at this stage.



**Figure 2.2: eMoO solution call flaw**

The diagram presents the steps required to cover all operations in a single sequence. However, they are grouped in the context of the operations. For the sake of simplicity, some details of required steps to are also omitted. These will be included in the low-level design documents, to be produced as part of the project deliverables. The summarized description of each step is presented below:

* + - 1. Deposit

|  |  |  |
| --- | --- | --- |
| 1. **Deposit** | Step 1: | The *remitter* hands in money to an eMoO *agent* and provides his/her mobile number*.* |
| Step 2: | The *agent* accesses the eMoO system via Web, types the *remitter’s* information (number is enough), as well as the amount to be loaded. |
| Step 3: | eMoO system verifies the existence of an account for the *remitter*. If it does NOT exist, it automatically creates it, and credits the deposited amount. The transaction details displayed on *agent* screen*.* |
| Step 4: | The *agent* prints a receipt with the transaction details and hands it in to the *remitter*. |
| Step 5: | eMoO system notifies the *remitter* via SMS about the top-up and the updated account balance. |

* + - 1. Funds Transfer

|  |  |  |
| --- | --- | --- |
| 1. **Funds Transfer** | **Step 1:** | The *remitter* requests the transfer of funds to the *beneficiary* by sending an SMS to eMoO system short number. |
| Step 2: | The following checks are performed eMoO system in this step:   1. If the funds are not enough to cover the transaction, a notification is sent to the *remitter;* 2. If there are enough funds but the *beneficiary* does NOT have an account, funds are debited from the *remitter’s* account and credited into a temporary transfers account. eMoO informs the *beneficiary* about the pending transfer and requests an authorization to create an eMoO account. The *remitter* is informed about the pending request. 3. If there are enough funds and the *beneficiary* has an account, the amount the debited from *remitter’s* account and credited into *beneficiary’s* account. The process then continues in step 4. |
| Step 3: | If the *beneficiary* responds positively, the account is created and funds are debited from the temporary transfers account and credited into the *beneficiary* account. The process then continues in step 4. If the *beneficiary* responds negatively or does not provide feedback within the defined feedback period, the amount is reverted to *remitter’s* account and the transaction aborted. |
| Step 4: | eMoO system notifies the *beneficiary*/*remitter* via SMS about transaction feedback. **In case of success:** the SMS contains transfer details and the new account balance. **In case of failure:** the SMS contains the reasons behind the failure. |

* + - 1. Online Payment

|  |  |  |
| --- | --- | --- |
| 1. **Online Payment** | Step 1: | The *e-payer* browses to a *merchant* website that accepts eMoO payments, agrees on the price and confirms the interest in a given purchase. |
| Step 2: | The *merchant* e-commerce application issues a payment order request to eMoO system, via the merchant’s API. |
| Step 3: | The eMoO system authenticates the request, and generates a payment order code, worth the specified amount. The code is sent back to *merchant* e-commerce application. |
| Step 4: | The *merchant* e-commerce website displays the code to the *e-payer* and requests him/her to proceed with the payment. |
| Step 5: | The *e-payer* copies the code to his/her mobile an issues a payment request*,* by sending an SMS to eMoO system short number.  (Note: The eMoO system should alternatively allow this operation to be done via a web portal) |
| Step 6: | eMoO system verifies whether the number of the SMS sender is associated to an eMoO account, as well as funds are available for the payment. Upon successful verification, it generates an *e-Voucher* worth payment value, deducts the amount from *e-payer’s* account and transfers it into a transition account (for pending payments), and sends the *e-Voucher* to *e-payer’s* mobile. |
| Step 7: | The *e-payer* submits the *e-voucher* code to *merchant’s* e-commerce website. |
| Step 8: | The *merchant* e-commerce application issues a payment confirmation request to eMoO system, via the merchant’s API. |
| Step 9: | eMoO system transfers the reserved funds associated to the *e-voucher* from the transition account into *merchant’s* account and sends a payment confirmation to the e-commerce application. |
| Step 10: | The *merchant* website confirms the purchase, and grants the contents (in case of online content being sold) or concludes the purchase (in case other products) |
| Step 11: | eMoO system notifies the *e-payer* via SMS about the purchase, the merchant and the updated account balance. |

* + 1. Additional information about eMoO features

Deposit

The design of step 2 is such that requires the *agent*’s computer to be connected to the Internet. It should be interesting to consider a design variant that allows agents without Internet connectivity to use the system. This can be accomplished by designing a standalone application to run on *agent’s* computer, able use SMS as the bearer of the traffic to be exchanged with the eMoO system.

Online Payment

The design of step 5 should be expanded in order to allow the *e-payer* to issue a payment order via a web portal. This is useful in cases that the *e-payer* travels to a country in which there are no agreements with mobile operators or Value added Service Providers, so that the request is sent via SMS. Step C-5 also incorporates a feature that allows the *e-Payee* identity to remain private, which can be an interesting feature when the person buying contents does not want to be known by the service provided. Anonymity is made possible because the *payment order* and *e-voucher* codes do not explicitly contain information about the e-*payer*. However, because it is important to be able to track this information (e.g. to settle disputes), details are kept on the eMoO system.

In step 6, if the *e-voucher* is not used after a specific (configurable) period of time, the amount must be returned to ­*e-payer’s* account and the ­*e-voucher* validity revoked. When an *e-voucher* is revoked, the *e-payer* must be notified. The system should also be allowed to explicitly request the revocation of an *e-voucher*, if it is not yet used.

In step 7 the *e-payer* is required to submit to the *merchant* a payment confirmation code (or the *e-voucher)*. The *e-voucher* is associated with the payment order request generated in step 3, and it can only be used for that specific payment. One could argue that step 7 could be performed automatically between e-MoO system and the *merchant’s* e-commerce application. That is a valid point, but it would increase the level of complexity to be added to the *merchant’s* e-commerce application, is it must be designed to support operation in server mode.

# Project Goals and Objectives

This project has the following two goals:

**a) Primary Goal:** Develop critical thinking skills in the students, as well as their ability to methodically work in teams in order to solve “real-world” problems, within a pre-defined time frame;

**b) Secondary Goal:** Implement the new of the eMoney System, as well as a use case to be used as the proof of concept.

In order to achieve the project goals, students will learn/put into practice knowledge and skills of systems design, systems integration, applications development, resorting to open source operating systems and tools. Students will also be encouraged to exercise their documentation and reporting skills. Therefore, the following specific objectives are defined:

1. Review and criticize the high-level solution description;
2. Provide inputs for Design of the required system components (eMoney order accounts, e-Voucher Generation, SMS processing logic, merchant Payment Request order, transaction historic);
3. Get familiar with of the Linux operating system (overview of the architecture, file system navigation, basic command, installing and running application, basic network connectivity tests);
4. Select the programming languages for the backend and front-end (Web-based) applications;
5. Design and implement the web applications required for communication with the *agents*;
6. Design and implement the Databases required by the solution;
7. Design and implement the HTTP API required for communication with the *merchants*;
8. Install and configure and manage a HTTP Linux server;
9. Install and configure and manage a Database Linux server;
10. Install and configure and manage a SMS-Gateway Linux server;
11. Install and configure Virtual Private Network (VPN) clients and server in Linux;
12. Implement the e-Voucher generation logic;
13. Integrate the SMS-GW with an operator’s SMSC;
14. Produce the test cases document;
15. Produce the solution documents;

# Methodology

In order to accomplish the project goals, the students will work in a learn-by-doing environment, with close mentoring. Therefore, students will be introduced to the technology topics and tools required for the project. That will happen in sessions with experts on the topics. A team of coaches will supervise their work during the project life.

Although the focus is to get the students familiar with the hands-on activities, the project will be conducted as if the students were a company contracted to develop the solution. This will only teach them technology, but as well as other relevant aspects from professional life. A laboratory environment will be made available on the cloud and there will be opportunity to do real tests with mobile operators through partnerships to be discussed.

The students will be involved in the design process. Because the overall solution is composed by different components (which require different type of skills), the group of students will be divided in sub-teams responsible in order efficiently tackle specific problems related to different parts of the system. However, the interfaces to each of the system component must be clearly defined and agreed upon. Additionally, each sub-team will required to regularly present the progress of the work to everyone, on a bi-weekly basis. Every team will have to submit weekly reports.

It is important to create balanced sub-teams, and this is coaches have a role to play. Based on the system description, each student should be able to express interest on the component He/She would like to work on. The preliminary sub-team are proposed in section 6.2.

# Work Plan

<TBA once feedback from students is received>

# Project Team

This project will involve students coached by a team of Computer Science and ICT lecture and professionals. The s

* 1. **Coaching Team**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibilities** |
| Rasika Dayarathna | Coach & Co-owner | * Supervision/Examiner of project Primary Goal * Specify and approve the solution Business Rules * Project Management (logistics, students selection & registration) * Methodology Mentoring * Review & Approval the Functional * Review of the progress reports * Advise on the usability and legal aspects |
| Eneas Hunguana | Co-Coach & Co-owner | * Supervision/Examiner of project Secondary Goal * Specify and approve the solution technical requirements * Technical coordination and Mentoring * Preparation of the lab environment * Organize the technical sessions for knowledge transfers * Review & Approval the Technical solution * Review of the progress reports |

**6.2 Students Teams**

<TBA once feedback from students is received>

|  |  |  |  |
| --- | --- | --- | --- |
| **Team A** | **eMoO accounts and e-voucher system** | | |
| **Name** | **E-Mail** | **Key-Skills** | **Hrs/week** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Team B** | **eMoO agent Web Interface and SMS processing logic** | | |
| **Name** | **E-Mail** | **Key-Skills** | **Hrs/week** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Team C** | **eMoO Merchant API interfaces & Use Case Application** | | |
| **Name** | **E-Mail** | **Key-Skills** | **Hrs/week** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Note:** Every team will be responsible for designing and building the mechanisms for direct interaction with the SMS-Gateway through the documented API.

# REFERENCES

[1] “Electronics Funds Transfer”; <http://en.wikipedia.org/wiki/Electronic_funds_transfer>

[2] “Electronics Funds Transfer”; <http://fms.treas.gov/eft/index.html>

[3] De Zoysa, K., Dayarathna, R. eMoney Order, 2003; http://....

[4] Dayarathna,R., Hunguana, E. A Micro-Payment System for Underprivileged Communities, 2009; https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CCgQFjAA&url=http%3A%2F%2Fwww.w3.org%2F2008%2F10%2FMW4D\_WS%2Fpapers%2Frasika.doc&ei=aUhtU-WOMIaY1AXsrIHQCQ&usg=AFQjCNFYf6A2Qp\_fjYAUyDivvQ\_ZC0K6Qw&bvm=bv.66330100,d.d2k