



## **goMLM - Software Documentation**

### **NAME**

go MLM - Version 0.9.3

### **PURPOSE**

Business software solution for Binary based Multi level marketing.

### **PLATFORM**

Server – Apache/IIS

Operating system - Windows/Linux platform.

### **SOURCE LANGUAGE**

Coding language - PHP

Database - mysql

### **DESCRIPTION**

go MLM is a web based application which provides complete solution for all challenges raised by the Multi level marketing industry. The software experts in visualizing the user graph tree (binary genealogy, down line), calculating periodically payments (Binary, Spillover, awards etc.), generating epin, e-wallet and complete maintenance required for the administrator.

### **INPUT**

- First time visit: Registration information (to be stored in database) is to be filled by the user in the join user form.
- After: User Login information to be provided in login page.

## **OUTPUT**

- **Binary tree** (member details in genealogy format)
- **Down line tree** (complete down line member details)
- **Invoice** (Income Bill generated automatically by the software)
- Binary pay Commission details
- Spill over pay commission details
- Edit profile (Change user contact, bank and nominee information details)
- Change Password (change user login and epin password)
- Check epin (view epins bought by the user)
- Logout page clears the user session and he will be logged out.

## **USE**

It is a normal web application. All it needs is the user login information at first and then, on it produces a complete user friendly interface, with which the user can manage his account online.

## **HARDWARE REQUIREMENTS**

VPS or a private web server for optimized performance.

## ABOUT PHP & MYSQL

### **What is PHP?**

- PHP stands for **PHP: Hypertext Preprocessor**
- PHP is a server-side scripting language, like ASP
- PHP scripts are executed on the server
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP is an open source coding language
- PHP is free to download and use

### **What is a PHP File?**

- PHP files can contain text, HTML tags and scripts
- PHP files are returned to the browser as plain HTML
- PHP files have a file extension of ".php", ".php3", or ".phtml"

### **What is MySQL?**

- MySQL is a database server
- MySQL is ideal for both small and large applications
- MySQL supports standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use

### **PHP + MySQL**

- PHP combined with MySQL are cross-platform (you can develop in Windows and serve on a Unix platform)

### **Why PHP?**

- PHP is an open source language
- PHP runs on different platforms (Windows, Linux, Unix, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP is FREE to download from the official PHP resource: [www.php.net](http://www.php.net)
- PHP is easy to learn and runs efficiently on the server side

### **Where to Start?**

To get access to a web server with PHP support, you can:

- Install Apache (or IIS) on your own server, install PHP, and MySQL
- Or find a web hosting plan with PHP and MySQL support

## **Software development process**

A **software development process** is a structure imposed on the development of a software product. Similar terms include software life cycle and *software process*. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider a lifecycle model a more general term and a software development process a more specific term. For example, there are many specific software development processes that 'fit' the spiral lifecycle model.

### **Overview**

---

The large and growing body of software development organizations implements process methodologies. Many of them are in the defense industry, which in the U.S. requires a rating based on 'process models' to obtain contracts.

The international standard for describing the method of selecting, implementing and monitoring the life cycle for software is ISO 12207.

A decades-long goal has been to find repeatable, predictable processes that improve productivity and quality. Some try to systematize or formalize the seemingly unruly task of writing software. Others apply project management techniques to writing software. Without project management, software projects can easily be delivered late or over budget. With large numbers of software projects not meeting their expectations in terms of functionality, cost, or delivery schedule, effective project management appears to be lacking.

Organizations may create a Software Engineering Process Group (SEPG), which is the focal point for process improvement. Composed of line practitioners who have varied skills, the group is at the center of the collaborative effort of everyone in the organization who is involved with software engineering process improvement.

### **Software development activities**

---

#### **Planning**

The important task in creating a software product is extracting the requirements or requirements analysis. Customers typically have an abstract idea of what they want as an end result, but not what software should do. Incomplete, ambiguous, or even contradictory requirements are recognized by skilled and experienced software engineers at this point. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect.

Once the general requirements are gathered from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document.

Certain functionality may be out of scope of the project as a function of cost or as a result of unclear requirements at the start of development. If the development is done externally, this document can be considered a legal document so that if there are ever disputes, any ambiguity of what was promised to the client can be clarified.

### **Implementation, testing and documenting**

Implementation is the part of the process where software engineers actually program the code for the project.

Software testing is an integral and important part of the software development process. This part of the process ensures that defects are recognized as early as possible.

Documenting the internal design of software for the purpose of future maintenance and enhancement is done throughout development. This may also include the writing of an API, be it external or internal. It is very important to document everything in the project.

### **Deployment and maintenance**

Deployment starts after the code is appropriately tested, is approved for release and sold or otherwise distributed into a production environment.

Software Training and Support is important and a lot of developers fail to realize that. It would not matter how much time and planning a development team puts into creating software if nobody in an organization ends up using it. People are often resistant to change and avoid venturing into an unfamiliar area, so as a part of the deployment phase, it is very important to have training classes for new clients of your software.

Maintaining and enhancing software to cope with newly discovered problems or new requirements can take far more time than the initial development of the software. It may be necessary to add code that does not fit the original design to correct an unforeseen problem or it may be that a customer is requesting more functionality and code can be added to accommodate their requests. If the labor cost of the maintenance phase exceeds 25% of the prior-phases' labor cost, then it is likely that the overall quality of at least one prior phase is poor. In that case, management should consider the option of rebuilding the system (or portions) before maintenance cost is out of control.

Bug Tracking System tools are often deployed at this stage of the process to allow development teams to interface with customer/field teams testing the software to identify any real or perceived issues. These software tools, both open source and commercially licensed, provide a customizable process to acquire, review, acknowledge, and respond to reported issues. (Software maintenance)

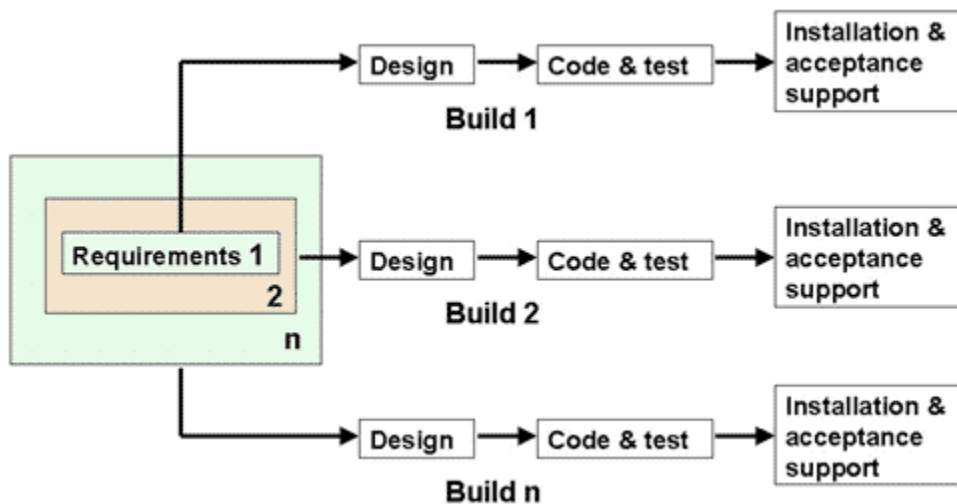
## Software Development Models

---

Several models exist to streamline the development process. Each one has its pros and cons, and it's up to the development team to adopt the most appropriate one for the project. Sometimes a combination of the models may be more suitable.

### **goMLM was built by Iterative and Incremental Development**

**Iterative and Incremental development** is at the heart of a cyclic software development process developed in response to the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between.



### **The Basic idea**

---

A common mistake is to consider "iterative" and "incremental" as synonyms, which they are not. In software/systems development, however, they typically go hand in hand. The basic idea is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing the developer to take advantage of what was learned during the development of earlier portions or versions of the system. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and

implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of a task from the project control list, and the analysis of the current version of the system. The goal for the design and implementation of any iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not dictated by the interactive approach. In a light-weight iterative project the code may represent the major source of documentation of the system; however, in a mission-critical iterative project a formal Software Design Document may be used. The analysis of iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, usability, reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

### **Iterative/Incremental Development**

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross-discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

- Inception identifies project scope, risks, and requirements (functional and non-functional) at a high level but in enough detail that work can be estimated.
- Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
- Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
- Transition delivers the system into the production operating environment.

Each of the phases may be divided into 1 or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full.

### **Implementation guidelines**

Guidelines that drive the implementation and analysis include:

- Any difficulty in design, coding and testing a modification should signal the need for redesign or re-coding.

- Modifications should fit easily into isolated and easy-to-find modules. If they do not, some redesign is needed.
- Modifications to tables should be especially easy to make. If any table modification is not quickly and easily done, redesign is indicated.
- Modifications should become easier to make as the iterations progress. If they are not, there is a basic problem such as a design flaw or a proliferation of patches.
- Patches should normally be allowed to exist for only one or two iterations. Patches may be necessary to avoid redesigning during an implementation phase.
- The existing implementation should be analyzed frequently to determine how well it measures up to project goals.
- Program analysis facilities should be used whenever available to aid in the analysis of partial implementations.
- User reaction should be solicited and analyzed for indications of deficiencies in the current implementation

## Multi-level marketing

---

**Multi-level marketing (MLM)** is a marketing strategy in which the sales force is compensated not only for sales they personally generate, but also for the sales of others they recruit, creating a down line of distributors and a hierarchy of multiple levels of compensation. Other terms for MLM include **network marketing**, **direct selling** and **referral marketing**.

Although the products and company are supposed to be marketed directly to consumers and potential business partners by means of relationship referrals and word of mouth marketing, critics have charged that most MLMs are pyramid schemes.

### Setup

---

Independent, unsalaried salespeople of multi-level marketing, referred to as distributors (or associates, independent business owners, dealers, franchise owners, sales consultants, consultants, independent agents, etc.), represent the company that produces the products or provides the services they sell. They are awarded a commission based upon the volume of product sold through their own sales efforts as well as that of their downline organization.

Independent distributors develop their organizations by either building an active customer base, who buy direct from the company, or by recruiting a downline of independent distributors who also build a customer base, thereby expanding the overall organization. Additionally, distributors can also earn a profit by retailing products they purchased from the company at wholesale price.

## **BINARY PLAN**

Binary plans are so named because they are built on a matrix of two. You can sign only two people onto your first level. Everyone else goes beneath those people. One of the good features in binary plans is that you are not paid based on levels at all, but instead, on a specified, maximum purchases (over some period of time, often weekly). It pays on that amount regardless of how many levels down you need to look to reach that total. So the number of levels between you and a given recruit is not especially relevant. The thing that determines whether you get a commission from that person is not what level he is on, but rather how much purchase volume is generated in the levels between you and that person.

Put differently, a matrix plan (not a binary plan) pays off on all purchases that occur through a specified maximum number of levels, regardless of how small or large a cash amount that might be. Binary plans turn that around. They pay off on all purchases that occur through a specified maximum amount, regardless of how many levels away you have to look to accumulate that total.

The advantage of binary plan over the standard matrix is that you can easily benefit from sales that occur many levels away from you, if that's where the volume is. The disadvantage to the other differentiating feature of a binary system is the "balancing act."

Since you can only sign two people directly under yourself, you end up with a structure that has two "legs" - two spots that in turn have all the other spots under them. In any given pay period, the two legs will likely be unequal (one will have generated more sales than the other). A binary system pays based on the volume of the weaker of the two legs. There is usually some mechanism to carry the unused volume of the stronger leg forward to the next pay period.

Even though this structure encourages spillover (that is, you will likely have people placed under you by your upline), those people generally end up in one leg, and you must balance that volume with recruits of your own (in your other leg) in order to benefit from that spillover.

Another common feature of binary plans is that there is usually some mechanism by which you can re-enter your own downline. This allows you to create a second position in the matrix that shares one leg with your initial position. For that position to receive commissions, you only need to build one new leg, since it would have its first leg essentially pre-built for it from the efforts of maintaining your primary.

If you join a binary system in a group that you feel is likely to produce spillover, it is a good idea to start with two positions if possible. If you place your own recruits mostly on the two legs of

your second position, you will probably maximize the income from those people as they balance each other; and their combined total will all end up under one leg of your primary position which will, in theory, largely be balanced by the spillover from above that will end up in the other leg of that first position.

## **Our Binary MLM Software Features**

- Binary Software 2XN
- Auto Spillover
- E-Wallet: Fund transfer member to member & admin to member. Buy New Products.
- Print Commission
- Real-time Commission
- Product Manager
- Tools: Backup Database, Restore Database, Banner Ads
- Genealogy Tree View
- Network Tree View
- Withdrawal Report
- E-Pin Module: Card Center, Serial No & E-Pin No.
- TDS Module
- Automatic Commission Processes.
- SMS integration (carries user login information and payment updates).

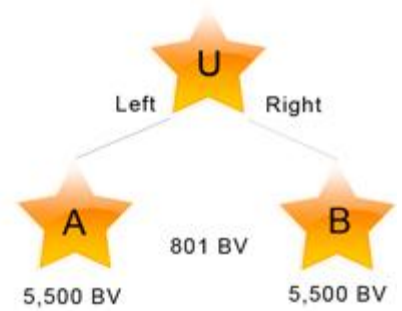
## **Types of Commission Modules**

- 1. Spill over Commission
- 2. Binary Pair Commission
- 3. Awards (on level basis)

**(Any other commission shall be added as per the client requirements)**

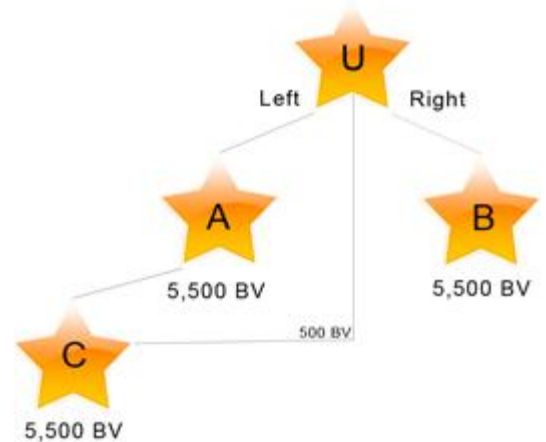
## Binary Income

- \* First Pair (1:1) (A & B is compulsory)
- \* All Future Pair (1:1)
- \* Pair Income
- \* Capping amount
- \* Carry forward allowed



## Spill Over Income

After completing the A & B, if you sponsor any distributor under your down line, you get additional spill income of each displacement. This amount is not limited by capping amount as well. So you can introduce as many spills you want and earn your dream income without any limitation of time or depth.



## Awards

User will be given award prices for every admin specified level of binary pairs he completes.

Check our website for latest updates

<http://www.ummstudios.com/>