

Sports Betting - Tipster Platform

Paulo Martins

paulo.s.martins@tecnico.ulisboa.pt

Instituto Superior Técnico

Abstract—Online Betting Tipsters platforms are websites where people provide betting advices, mostly with a betting analysis, in order to help other bettors to improve their results and grow their betting bank. Given the increase of online gambling all over the world (number of users and betting amount), these platforms demand have also been raising due to the need of getting relevant information about daily matches. This project enables any user to be a Tipster and to follow any Tipster to copy their tips, by placing real bets on a Sportsbook automatically. Our system will also allow Tipsters to use a Staking Plan, in order to help them and their followers to manage their bets and their Bankroll in a most profitable and secure way with the valuable help from Amazon AWS services. Additionally, our system provides unique features with the creation of two subsystems, that makes possible to hedge real bets and to select a future match (with unset odds) to be bet instantly once they are set.

Index Terms—Betting; Tipsters; Tips; Staking Plan; Amazon AWS; Automation.



1 INTRODUCTION

THIS document has the main objective to propose and present the **creation of a website, integrated with an automated software system**, which will make possible to everyone to get involved with Online Betting.

Online Sports Betting or Online Gambling (if we also consider Online Casinos, Poker, Bingo and Lotteries) had its first important expansion around 1997 [1]. — also because of the rise of Internet users which is around 40.4% of World's population [2]. Online Gambling works the same way as on physical world casinos or bingos — with the exception that this is online and every betting platform accepts a huge variety of deposit and withdrawal methods, such as Bank Wire transfer, Visa, Western Union or many Online Wallets.

There are two different kind of platforms where it is possible to place bets. The first, **Sportsbooks** (or Bookmakers — the type we will be using on this project), where you place bets, wait for outcome of a certain event and you earn (or lose) money from that bookmaker. Some of them also have a physical store, mainly in England, where any adult

may place their bets. Secondly, **Betting Exchanges**, where you place bets against other people — you earn (or lose) money from someone else — it is a person-to-person betting. There are few companies of this type that guarantee revenue for a charge of at least 5% of each winning bet. The main difference between these two kind of platforms is that it is possible to do Trading on Betting Exchanges — you can guarantee profit (no matter what outcome happens) only with the variation of odds on a certain event (even before that event begins), as it exactly happens on Financial Markets.

2 BACKGROUND

In this section, important concepts related with Betting are going to be covered in order to understand how Sports Betting works. To begin we must understand what **Odds** mean — It reflects the probability of a certain outcome of a match that may belong to any existent sport. There are some ways to represent the odds — the most commons are decimal (Europe) and fractional (United Kingdom). As Odds reflect the probability, they might be calculated by $Odds = \frac{1}{Probability}$. Hence, if for a certain outcome the odds are 2.00, it means it has 50% probability of happening.

Each **Odds are generated by Bookmakers, through their algorithms** (which are nearly impossible to discover — if they were known, there

• Paulo Martins, nr. 72976,
E-mail: *paulo.s.martins@tecnico.ulisboa.pt*, Instituto Superior Técnico, Universidade de Lisboa.

would be many researches in order to find a way to beat them). If one bets on a specific outcome, he is betting that outcome to happen and to **calculate profit or loss** of a bet done on a certain Sportsbook, we may use the following formula $Profit = Stake * (Decimalodd) - Stake$

Following the last concepts, imagine Liverpool will play against Chelsea (at Stamford Bridge, Chelsea's Home Stadium) and Away side odds are set at 3.80 ($\frac{1}{3.8} = 26.3\%$ probability for Liverpool's win) and we consider that they have 35% chance of winning ($Odds = \frac{1}{0.35} = 2.86$), hence any odds higher than 2.86 may be considered a **value bet**.

If one bets 100 Euros on Liverpool at 3.80 odds before the match and then Liverpool scores a goal during the match, we may guarantee profit at that time — it is called **Bet Hedging**. Imagining that it is now 0-1 and Liverpool odds are 2.00 to win the match, there are manual ways to assure a part of the potential profit, reducing the risk from 100 to 0. Conversely, it is also possible to reduce the loss: Imagine at half time the current score is 0-0, Chelsea is dominating Liverpool and Away's odds are 4.50. Again, manually, it is possible to calculate how much we need to bet on other outcomes to reduce our potential loss from 100 to around 20 whatever the outcome will be.

3 RELATED WORK

3.1 Tipsters Platforms

Along this chapter we will show most of the existent Tipster Platforms, presenting their flaws and their positive aspects. It is possible to qualitatively separate some Tipster Platforms, according with the features they provide, as it follows.

First, **Low quality platforms** (e.g ApostaGanha, Academia das Apostas, Betshoot, Bettingexpert or Tipster Academy). Second, **Medium quality platforms** (e.g BetAdvisor). Third, **Top quality platforms** (e.g Blogabet) And fourth, **Best platform** (not related with Sports Betting) — (e.g eToro).

These categories' features may be observed on the following Figure 1 where Blogabet is the best one, showing more features than its competitors along with eToro — the only one that has the **Automated Betting** feature.

* - Requires additional payment.

** - It is only a spot with the last tips inside each Sport, it isn't like Facebook, where you can share a tip and everyone are able to comment.

*** - Yes, but that platform has nothing to do with Sports Betting, only with Financial Markets.

	BLOGABET	APOSTA GANHA	BETSHOOT	TIPSTER ACADEMY	BETADVISOR	BETTING EXPERT	ACADEMIA DAS APOSTAS	ETORO ***
FOLLOW TIPSTERS ANALYSIS	YES	NO	NO	NO	YES	YES	YES	YES
BETTING FEED	YES	YES **	YES **	YES **	YES **	YES **	DEPENDS	YES
E-MAIL NOTIFICATIONS	YES *	NO	NO	NO	YES *	NO	NO	YES
SMS NOTIFICATIONS	YES *	NO	NO	NO	YES	NO	NO	YES
NO DELAY ON NOTIFICATIONS	NO	---	---	---	YES	---	---	YES
PRIZES FOR TIPSTERS	YES	YES	YES	YES	YES	YES	YES	NO
USERS COST	DEPENDS	NO	NO	NO	YES	NO	NO	NO
AUTOMATED BETTING	NO	NO	NO	NO	NO	YES	NO	YES ***

Figure 1. Existent platforms and its features

3.2 Staking Plans

After analyzing most well-known betting staking plans, we have split them into two different groups, highlighting their strengths and weaknesses.

First, **Loss chasing** (e.g Martingale, Fibonacci and D'Alembert) [15] — This strategy is related with Casino, however it could also be used through Sports Betting and it is the worst, because after each loss, the bettor should bet twice of the previous bet and it is relatively easy to get into a run of consecutive losses. This strategy may fit on a short run of losses for someone who holds a huge quantity of money, but on long runs of losses, it tends to lead to bettor's bankrupt. Also, Bookmaker's maximum bet allowed may dangerously interfere here in a bad way — not letting that bettor bet that higher amount [3].

Second, **Bankroll maximizing and controlling** (e.g 1326, Kelly's Criterion, Percentage and Secure) — Although the remaining staking methods do not have all the same way of working, all of them aim for maximizing and better control the betting bank. 1326 and Kelly's Criterion main goal is to maximize the winnings (but also the losses) in short runs, however for long runs as analyzed before, even with their little bank protection, they are not the best. Kelly's Criterion might be good to use in conjunction with another system, however using it alone is hard to be done and requires expertise — that is why it would not be beneficial to implement it to be used by overall Tipsters.

In overall, **Secure Betting staking plan** it is the best for long runs of bets [5] and the easier to use for most people. This tool should be implemented on my project, so that if a certain Tipster is willing to bet on Odds around 2.00, the system would

automatically place that bet with 5% of confidence; if odds are 14.00, bet's confidence should be 5% * 30% = 1.5% of the betting bank.

3.3 Scientific Research

3.3.1 *Efficiency in Horse Races Betting Markets : The Role of Professional Tipsters*

Based on an huge data-set of around 11.000 predictions, Deschamps & Gergaud, a forecasting model [6] was developed in order to test and detect exaggeration on the published tips — the model consists on assigning scores to each Tipster, according to how many horses they correctly picked to be in the Top 5 (or any other Top, according to how many horses were running), getting extra points when a Tipster predicts correctly the Horses place by arrival order. As there are a significant number of Tipsters competing, some of them tend to pick some unexpected horses as winners, trying to differentiating them from the rest, getting a big advantage — this fact may also affect this analysis.

The conclusion they met is that most Horse Racing Tipsters tend to make risky predictions, in order to highlight themselves among other Tipsters, exaggerating on their Tips more than expected. This event will make markets not being totally efficient, because bettors will follow these professional Tipsters, placing huge amount of money on their outcomes [7]. This will generate betting opportunities to the more conservative and cautious Tipsters (and bettors), because of these market's inefficiency. Horse Racing was the analyzed Sport, however this happens on every single Sport — Mainly secondary football league matches or Tennis matches where odds vary often with tips release.

Regarding our project, this paper shows Tipsters influence and significance, which creates the necessity to organize Tipsters statistics the greatest possible, in order to show possible followers all betting strategy information about them. Otherwise our Users may follow certain types of Tipsters that they did not really want to.

3.3.2 *Testing semi-strong efficiency in a fixed odds betting market*

In this paper [8], Bernardo, Ruberti & Verona choose four major European Football championships (classified as **semi-strong** markets, because everyone has access to the same information, being the odds adjusted immediately and efficiently to any important news. [9]). To do this, they choose

to examine the following matches after a coach replacement on the middle of a season. From 2006 until 2013, they found 203 cases over these four leagues and their analysis lead to an increase of performance on all these leagues with Spain having the highest percentage (87%) of better results.

To test Markets' efficiency (i.e if Odds are fairly set), they will try to apply a betting pattern to get profit — They tested this event by simply betting on the first four matches after the arrival of the new team manager. After summing all leagues and all seven seasons, they reach the payoff of around 9%, which makes this potentially a good method to earn money on long term. Following the last paper [6], on these semi-strong markets (where everyone has access to the same information) is still possible to find inefficiency on Markets and consequently apply a betting pattern as this one that would bring profit.

Regarding our project, this specific event is just one of a dozen and shows that a profitable Tipster do not need to have inside information to keep making profit (finding Markets inefficiency). As this one, there are many unexplored perspectives that may be successful on Online Sports Betting.

3.3.3 *Doubling Revisited: The Mathematical And Psychological Effect Of Betting Strategy*

Along this research Turner & Horbay [10] examine and simulate tests using different types of Staking Plans on Casino. Most gamblers state that doubling their bets after a loss (Martingale or other Loss-Chasing strategy [15]) improves their chances to profit. As we stated when analyzing Loss-Chasing systems, they also may lead to a fail in a spectacular manor and may also lead to gambling problems [11]. Not only people who increase their stake after a loss, but also people who do it after a win may suffer gambling problems, because they tend to treat earned money as "Casino's money" (when someone treats earnings as money they are not afraid to lose by betting more).

Then, they simulate 1000 bets using each Strategy (e.g Martingale, D'Alembert [4]) for Roulette and for Slots, presenting graphs for each one with the evolution of profit and loss with number of bets made. There are few occasions (only two in a total of twenty) where a certain bettor would get profit after these 1000 bets on Casino.

Regarding our project and our Staking Plans analysis made before, this paper strengthens our

point of view about them — Loss-Chasing strategies are not suitable, especially for long runs of bets; Also, other Systems as 1326 [16] or D’Alembert [4] may be less aggressive, but still not optimal. This way we may reinforce that Secure Staking Plan is the most appropriated strategy suitable for the overall users and for long runs of bets.

4 SOLUTION

We will present our designed Architecture and Implementation, along with the decisions we had to do about them in order to make our system the most functional, usable and secure possible. To start, we will present the changes (and additions) on planned functional requirements, then we will show and explain every detail regarding our Architecture along with Implementation algorithms and used technologies.

4.1 Functional Requirements

Following the set of requirements, we needed to be implemented we decided to design a Flowchart diagram (Figure 2) with the main objective of allowing each to understand and visualize how our System really works [12], depending on User’s desired action and how it is logical split on three different Subsystems: Tips, Autobetting and Betting Interface.

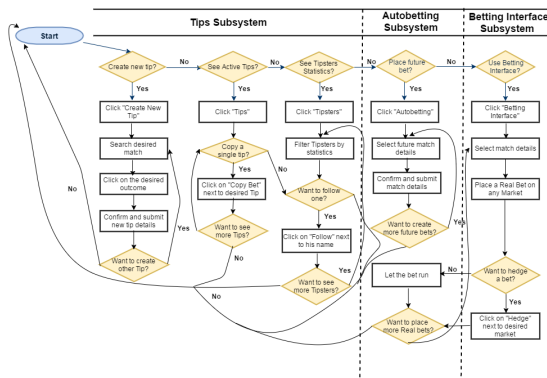


Figure 2. Flowchart Diagram

These set of features were implemented the most simplistic way possible, being user-friendly, so that each user can perform any action with a couple of clicks.

4.2 Architecture and Implementation

The developed system uses the well-known Three-Tier Architecture which is divided in three different parts [13]. The first one, **Client Tier/Front-End** — the part that contains everything (HTML

Elements) a user can see and interact, accessible on any browser through any device connected to the Internet, known by Graphical User Interface (GUI). Second, **Web Server Tier/Application Server** — This part is fully developed with PHP, Javascript and JQuery. It is the bridge between Users and system requests to the Back-End (or even to its own) and its responses to the Front-End, exposing necessary data to GUI. Along with Back-End this is the Tier that will process every single feature. Third, **Database Tier/Back-End** — The last piece is where everything is stored and handled. It includes the Drupal’s MySQL database, the filesystem or the Cron Jobs Scheduler.

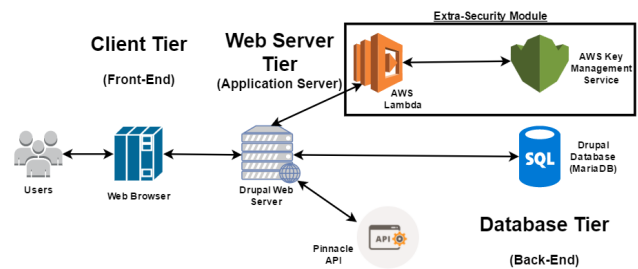


Figure 3. System's Architecture

Figure 3 shows the complete Architecture for our System, including the technologies and external services we decided to use and which part communicate with each-other. We can see our Web Server is using Drupal (a CMS — Content Management System, similar to the Worldwide known Wordpress). Pinnacle API, Amazon AWS Lambda and Key Management Service as external service to help placing bets and securing User’s data, respectively. These external services will be approached in detail over the next chapters. As mentioned, Drupal Web Server works as a bridge between himself, Database and Web Browser that exposes all visible information to the user.

4.2.1 Front-End

Regarding Client Tier, as the website uses Drupal, every developed page uses a theme that we choose to automatically renders each page using its own CSS rules.

We have not explored Drupal Front-End features too much, we preferred to create our own Elements and design them the way we like. Every page information is retrieved directly from its PHP file or from JQuery Ajax responses (after User interactions or automatic system scripts) and, depending on the type of information to be shown, we decided to

use different Elements and different CSS Rules, constantly trying to make it the most simple and usable possible. Some figures included on Appendixes attached to the main document shows some examples of our interface.

4.2.2 Application Server

Following our System's Architecture presented before at Figure 3, this tier may be divided in a couple of different parts. We opted to approach them by analyzing sequentially (following the order we performed its implementation into our System), in a minutely way, as it follows.

4.2.2.1 Betbrain: Although Betbrain is not an element on our Architecture diagram, getting Betbrain's iFrame (to show all available Odds for all matches) to work was our first objective as we needed it to create a new tip. Betbrain's iFrame and the integration with PHP, Javascript and JQuery have a steep learning curve, hence we spent a long time figuring out how to use these important technologies in the implementation phase to make it work perfectly.

Firstly, we tried to print the iFrame directly (using a specific HTML iframe tag) to recognize a user click on it, however it always generated Cross-Origin Resource Sharing (CORS) problem. After trying few workarounds, such as using third party frameworks, none was successful. **Hence**, instead of printing the iFrame, we tried to scrap each page and print its HTML code — all content — on our website. It worked, but with some bad formatted elements, some images missing and nearly zero buttons working. **In fact**, before we print the content, we managed to change all button clicks destination (because we noticed how their links work) to request a new page from Betbrain. When "Tennis" was clicked, the system would get the content of <http://betbrain.com/tennis/?portalId=1351>. And when a some odds field was clicked, a pop-up was shown with all correct information.

However if odds were gathered from another page of Betbrain's iFrame (i.e getting from the specific match odds page, instead from Live Matches page), it would be a nightmare, because all DOM elements were differently positioned or with different IDs.

The final solution encountered to fix this long and painful problem was to mix a little bit of these two last tries. We begun by getting

<http://betbrain.com/?portalId=1351> HTML content and remove some elements, in order to reduce design's complexity and possible errors. **Afterwards**, we created different Javascript onClick functions to treat different clicks (redirect to another page or select an outcome for a match) and gather all different information, depending on which page the user is.

4.2.2.2 Pinnacle API: After Betbrain fully implemented we moved into Pinnacle API (RESTful service) integration, by trying to place bets using PHP. Integrating Pinnacle API (RESTful service) had a steep learning curve, hence we spent a some time figuring out how to use these relevant technologies as the Application Programming Interface. After it, the API looked simple to use and almost every response can be retrieved as JSON or XML. On every API call, the user parameters (username and password) should be passed on header following the format "username:password" base64 encoded and along (after) with the string "Authorization: Basic".

In what concerns automated betting, after we get all the match details, such as team names or sport, we are using the endpoints *sports* to match the sport, *textitleagues* to match the league and more data. After getting all these IDs, we are using *bets/place* endpoint to place the bet automatically.

4.2.2.3 Security: Tips are being created and posted, bets are being placed automatically, therefore securing User's Data was now the main challenge in this project, as we must uphold all the system's security requirements.

The most important aspect of this project is to secure each User's Pinnacle Username and Password — The hardest matter to deal with is the need to be a **Two-Way** encryption — because we need to encrypt and decrypt data automatically — instead of only One-Way (this type of encryption do not require to decrypt data), which makes it non-trivial to develop.

After analyzing the potential threats on this type of encryption and possible improvements, we found out that three different unfavorable behaviors may occur. Hackers can access **Database**, **Filesystem** (Source Code) or **inject scripts** on the website.

Eventually, if a hacker gains access to this three different sources, he would be able to discover any Username and Password from Pinnacle, because they would know the complete Encryption Algorithm we designed.

We edited an Encryption file that uses *key stretching* (simply by applying a function that uses the standard PBKDF2 derivation, to make the data longer and stronger), hides the *Initialization Vector* in order to create scrambled up keys every time and does *HMAC verification* of source data (because HMAC output is an hash of the concatenation between the generated key and the data to encrypt). **Perfect solution** would be composed by three different keys: **Server Key**, **User Key** and **User Private Key**, where the first two would be stored somewhere in-between database and/or file system, and the **User Private Key** would be an input by each user to retrieve the sensitive data.

However, as the system must decrypt it **automatically** without User's permission, this third key should be retrieved (automatically) in a different way — The solution was to have something about encryption outside Drupal that could be accessed **only** from Drupal when we needed to retrieve them to place the bets automatically, for instance.

Then, we found out **AWS Amazon Services**.

"Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow. (...) to build sophisticated applications with increased flexibility, scalability and reliability."

From their big range of services, and after an intensive research on their numerous alternatives we figured out that a service called **Lambda** was able to run single pieces of code:

"AWS Lambda is a compute service where you can upload your code to AWS Lambda and the service can run the code on your behalf using AWS infrastructure."

AWS Lambda allows us to run code remotely and securely, because nobody ever will get access to source code of this part.

In conjunction with this service, we are also using KMS service to manage encryption keys:

"AWS Key Management Service (KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data, and uses Hardware Security Modules (HSMs) to protect the security of your keys."

This code would handle to part of the encryption algorithm, making it harder for hackers to decrypt our sensitive data.

The key used to encrypt and decrypt is the result of concatenation between three different keys: one has to do with each e-mail (stored directly on database), other we can call **User Key**

(on HEX format, half split on a file on file system's root and the other half on database) and **Server Key** (the only way to get is through AWS Services).

AWS KMS has a Master Key that we use to encrypt and decrypt our **Server Key**, that is stored on our Filesystem (encrypted and formatted on HEX format). Whenever we need to perform any operation that needs User credentials, we will send that encrypted key to AWS Lambda (which will send it to AWS KMS to get decrypted). The response will be the **Server Key** decrypted.

Besides these requirements, SSL (Secure Socket Layer) protection is a must, and should take part of the Future Work, in order to avoid Man in the Middle attacks when these kind of information is requested.

After everything was greatly working on this part, we felt like some new great features, hence we decided to create a **Betting Interface** with some tools to help all bettors.

4.2.2.4 Betting Interface: Following the needs of new bettors and even experienced bettors, we built a page where each user may place bets on Pinnacle using our platform.

The big differences are that for each market we calculate the payout percentage which can be useful to know how much percentage Pinnacle is earning with that market. That payout percentage is calculated by:

$$\frac{1}{\left(\sum_{n=1}^n \frac{1}{nOdds}\right)}$$

Also, we have added the possibility to hedge every bets on every market you choose, in order to guarantee profit or reduce the possible loss. After an high number of tries and research, we finally discovered how to calculate the amount needed to place to **Hedge** a certain bet.

If the first bet was placed on Team 1, risking r euros to win pp euros. Hedging a bet means that the profit on both outcomes should be the same, hence we should solve this equation (x is the value to be calculated): $pp - x = (odds * x) - r$ Which will become: $x = (pp + r) / odds$

This way we are able to know how much we should bet to Hedge and we can easily calculate how much we can guarantee by now calculating $Hedge = pp - x$.

4.2.2.5 Cron Job: Cron job makes possible to run specific commands (or group of commands), automatically from time to time [14] — it can be set

to run at any interval of time, being the minimum one minute.

This feature can be useful for a big variety of purposes. Following the definition of value betting, we decided to add a new great functionality. Imagining we know two Tennis players (or any other Sports teams) — Lesia Tsurenko and Dominica Cibulkova — we know their value and we saw their performance on those matches. With some experience we can predict what should be the odds of each player on the next round when they play against each other.

After this, we guess Tsurenko L. has at least 27% of chance to win the match against Cibulkova D. which means any odds higher than $1/0.27 = 3.70$ have value.

Odds for this match were released at 31 August 21:17 on Pinnacle with odds 4.05 on Tsurenko's Win. If we only notice this on 2nd September in the morning, we can see her odds were already around 3.58, which did not have any value for us already — it represent less 18.21% of potential profit.

Then we create a Cron Job using the command: **curl -s http://eyebetting.com/cronSearch.php** that executes the script on the PHP file every 5 minutes. The script will search for any previously inputted matches on the Database Table *drcq_Autobetting* and check if the match and the market selected are already available to bet using Pinnacle API. If opening odds are bigger than the odds a certain user selected for that match, a bet will be placed automatically at the best odds available at that time, guarantying the maximum possible profit.

4.2.3 Back-End

After understanding well how PHP really behaves, we created a handle file (PHP written) to Ajax Post all kind of necessary requests from the Front-End. Almost every response contains HTML code to be completely printed somewhere on the website page, but there are also responses that contains an Array, that will be handled using Javascript or JQuery on the origin request file (Front-End).

As the most important component on our Back-End is our Database, we have designed each table to have an ID as its Primary Key (PK) and most of them a **uid** as Foreign Key (FK). *drcq_users_field_data* table is the one holding each users data, such as his name, followers, username, e-mail or encrypted Pinnacle username and password; *drcq_Tips* is responsible for storing all Tips details, so the system can place bets automatically,

along with the first table's help; *drcq_Autobetting* is similar to Tips table, however its goal is to be used by the CRON Job to place future bets; And on *drcq_Realbets* will be stored all real bets a certain User placed through our System (from copying Tips, from our Betting Interface or from our Autobetting subsystem). Most of relationships are one-to-many, since each User may post many tips, for example. The exceptions are the relationships between Autobetting and Tips with Realbets, because it is possible for a certain User can have some Real Bets with the necessity of having a single Tip or Autobetting entry.

5 SYSTEM'S EVALUATION

As we can not measure system's performance, we decided to test its usability and its functionality, to make sure everything was developed and designed the simplest and the most effective way possible. Our tests are based on the **User Test Protocol** [17], utilized on Interface Development Process.

We started by explaining the testers what is Online Betting along with some concepts included in our Background section. Then, we explained the purpose of our System and its main features. They were asked to fill out our Initial Questionnaire and we showed how the system worked for some minutes. Then they were allowed to explore themselves the website for a couple of minutes, before doing the test. After finishing it, our Testers were asked to evaluate our System according to its easiness of use and understanding, as we will display below.

Following Figure 2 of Flowchart Diagram, for each Subsystem we developed created Test Cases (the tasks and sub-tasks each User will need to perform, a total of 15) in order to evaluate their Usability with some Users.

5.1 Results

We have calculated to minimum, maximum, mean, median and mode of each sub-task according to how many seconds each User took to perform each task. Most of our Testers (73%) did not have any betting knowledge, however almost everyone completed every single task as expected — There were some tasks where we can see few Users took more than expected, however some others took much less.

Overall, we may conclude that our platform is showing great levels of Usability as almost every

User perform everything successfully with relatively easiness.

Table 1
Overview Questionnaire Statistics

Test Case	1	2	3	4
Statistic	Mean	Mean	Mean	Mean
Easy to Use	3.81	4	3.59	4.23
Easy to Understand	3.86	4.14	3.14	4.09

Table 1 show the statistics of our Overview Questionnaire, where we can observe the Mean values (from 1 to 5) gathered from our 22 Testers evaluation of Easiness of Use and Understanding from each Test Case. It is possible to visualize that the values from both types of evaluation are similar along the four Test Cases, and always positive and higher than the medium value (3).

6 CONCLUSION AND FUTURE WORK

This project was successfully developed fixing the existent flaws on Tipster platforms, by **following a Staking Plan** when **placing** (or copying) **bets automatically**, bringing advantages to the bettors who use it. This way the process of copying a bet from other bettors is fully optimized, replacing the need to place bets totally manually, as it happens nowadays. Our project also contains other two subsystems that are intended to ease some betting aspects, with the use of Autobetting page to automatically place bets upon odds release and Betting Interface which allows any user to place and hedge bets on Pinnacle Sports.

At this moment Pinnacle Sports is the only Sportsbook used through our System, we will also integrate with other well-known Sportsbooks (e.g Bet365, 188Bet and SBOBet) that do not have any available API. This way user will be able to create Tips from a wider range of Bookmakers. At that point, we will improve Autobetting and our Betting Interface Sub-Systems with these new Bookmakers.

We are also planning to display graphs on each Tipster profile to make easier to observe their statistics over time and provide the possibility to customize more settings on their profile. Another feature we would like to implement is the creation of a Chat System similar to Facebook (each User will be able to chat with each other on 1-to-1 communication and to create chat rooms).

Last but not less relevant, in order to reward our Tipsters, we will design and develop a Reward

System based on how many Users are following each Tipster — similarly to eToro system. [?]

ACKNOWLEDGMENTS

I would like to thank Supervisor teacher José Alberto Sardinha, who helped me overtaking all barriers, motivating me and coordinating me the best possible. I am also very grateful to all my friends for their friendship and support. Last but not least, I would like to thank my family for their love, encouragement and caring all these years.

REFERENCES

- [1] Jackson, D. A. (1994). Index betting on sports. *The Statistician*, 309-315.
- [2] Number of Internet Users (2015) - Internet Live Stats, "Internet Users in the World" Internet: <http://www.internetlivestats.com/internet-users/> [last accessed: Nov. 1, 2015], 2015.
- [3] Buchdahl, J. (2003). *Fixed odds sports betting*. High Stakes.
- [4] Staking Plans, "D'Alembert Staking Plan" Internet: <http://thestakingmachine.com/dalembert.php/> [last accessed: Dec. 14, 2015], 2012.
- [5] Staking Plans, "Secure Staking Plan" Internet: <http://stakingplan.com.au/secure-staking-plan/> [last accessed: Dec. 14, 2015], 2012.
- [6] Deschamps, B., & Gergaud, O. (2008). Efficiency in Horse Races Betting Markets: The Role of Professional Tipsters. *Handbook of Sports and Lottery Markets*, 341-354.
- [7] Smith, M. A. (2003). The Impact of Tipster Information on Bookmakers' Prices in UK Horse-Race Markets. *The Economics of Gambling*, 67-78.
- [8] Bernardo, G., Ruberti, M., & Verona, R. (2015). Testing semi-strong efficiency in a fixed odds betting market: Evidence from principal European football leagues.
- [9] Croxson, K., & James Reade, J. (2014). Information and efficiency: Goal arrival in soccer betting. *The Economic Journal*, 124(575), 62-91.
- [10] Turner, N. E., & Horbay, R. (2003). Doubling revisited: The mathematical and psychological effect of betting strategy. *Gambling Research: Journal of the National Association for Gambling Studies (Australia)*, 15(2), 16.
- [11] Blanco, C., Moreyra, P., Nunes, E. V., Saiz-Ruiz, J., & Ibanez, A. (2001, July). Pathological gambling: addiction or compulsion?. In *Seminars in clinical neuropsychiatry* (Vol. 6, No. 3, pp. 167-176).
- [12] Nassi, I., & Shneiderman, B. (1973). Flowchart techniques for structured programming. *ACM Sigplan Notices*, 8(8), 12-26.
- [13] Kambalyal, C. (2010). 3-tier architecture. Retrieved On, 2.
- [14] Keller, M. S. (1999). Take command: cron: Job scheduler. *Linux Journal*, 1999(65es), 15.
- [15] Betting Resource, "But what about martingale betting system?" Internet: <http://bettingresource.com/do-betting-systems-work.html> [last accessed: Dec. 14, 2015], 2010.
- [16] Staking Plans, "1326 Staking Plan" Internet: <http://stakingplan.com.au/1326-staking-plan/> [last accessed: Dec. 14, 2015], 2012.
- [17] Fonseca, Manuel J. et al. 2012. *Introdução ao design de interfaces*, FCA-Editora de Informática, Lisboa. ISBN: 978-972-722-738-9