

ChessHub

A Chess Tournament Management Tool

Autumn 2018

Laboratory of Systemic Modeling, LAMS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Optional Project in Computer Science

By
Colin Branca
&
Jules Courtois

Prof. Alain Wegmann
Gil Regev, Project Supervisor



Acknowledgements

We would like to express our deep gratitude to Gil Regev, our research supervisor, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. We also wish to thank professor Alain Wegmann and all the LAMS lab for their interest in our work throughout the semester, for their useful advices and more generally for having allowed us to work with them. This was a wonderful experience!

We would also like to extend our thanks to all actors from the chess world that helped us during this project: Luigi Forlano (VEGA), Roberto Ricca (FIDE), Maurice Gisler (FSE) and the Nyon Chess Club.

Abstract

It has been observed in the environment of chess tournaments that there are some usability or efficiency pitfalls, often caused by a lack of a good digitalization of a previously existing manual mechanism. Currently, tournament organizers use licensed pairing programs during the competition, these programs allow them to create the pairings for each round and generate rankings easily. Even if the existing software are quite hard to take in hand and lack some quality-of-life features, they do the job quite efficiently.

However, none of them feature automatic import of the participants in the system. For now, each organizer needs to have their own registration system (Google Form, email, phone, ...) and import the players in the system manually.

Exporting the results for transfer to the concerned chess federations is a feature usually suffering from a lackluster implementation and must also often be done by hand.

This project aims to correct a part of these pitfalls to provide a saner environment and reduce time consumption for tournament organizers, players, and federation officials.

Keywords

- Organizer: The organizer is the entity who manage a tournament. Most of time it is a chess club, but a federation can also organize some event.
- Player: The player is a lambda participant to the event. He only wants that the tournament is well organized and is only here to play.
- Federation: In chess, there is two level of federation the international one (*FIDE*) and the national one (*FSE* for Switzerland). Players have so two independent *ELO rankings* even if they are computed using the same system. In general, an organizer only handles with the national one if the event is internal and talks with both if the tournament is international.
- Pairing program: To be used in an official competition, a pairing program must be endorsed by the FIDE. There are actually five accredited programs: VEGA, SWISS SYS, SWISS MASTER, SWISS MANAGER and SWISS-CHESS.

Table of Contents

Acknowledgements	1
Abstract.....	2
Keywords	2
Table of Contents	3
Table of Figures	5
Introduction.....	6
Project Timeline.....	7
Inquiry	9
Entering the Chess World	9
Chess Federations	10
How A Tournament Works	10
Usage of a Pairing Program.....	11
Operation of the Chess World	14
Organizers' Opinion	15
Contextual Inquiry	15
First Project Idea.....	17
Goal	17
Difficulties.....	18
Explanation of ChessHub.....	20
Overview.....	20
Objectives	21
Business Part	22
Roadmap.....	22
External Objectives	22
Internal Objectives.....	22
Business Requirements.....	23
External Situation As-Is	23
Business Internal As-Is	23
To-Be Selection	25
Alternative 1.....	25

Alternative 2.....	25
Solution Selection	26
To-Be Description	27
Evolution.....	28
<i>Digital Strategy</i>	<i>30</i>
Management Summary	30
Strategic Challenges.....	31
Development	33
<i>Implementation.....</i>	<i>34</i>
Database.....	38
User Interface	40
<i>Conclusion</i>	<i>42</i>
Achieved Results.....	42
Future Development.....	42
Insights.....	43
Research Question.....	43
<i>References.....</i>	<i>44</i>
<i>Appendices</i>	<i>44</i>

Table of Figures

Figure 1: Nyon Active Chess	9
Figure 2: Little example on how a pairing works.....	11
Figure 3: Tournament Creation Page	12
Figure 4: Page for adding players into the tournament.....	13
Figure 5: Results Entering Page	13
Figure 6: Ranking Page	14
Figure 7: Echallens Active Chess	15
Figure 8: Database Schema of First Project Idea	17
Figure 9: Place of the initial idea in the system.....	18
Figure 10: As-is situation	24
Figure 11: First Alternative Diagram	25
Figure 12: Second Alternative Diagram	26
Figure 13: Application and Services Structure	37
Figure 14: Database Schema	38
Figure 15: Prototype: Profile Page.....	40
Figure 16: Prototype: "My Tournaments" Page	40
Figure 17: Prototype: "All Tournament" Page.....	41
Figure 18: Prototype : "Create Tournament" Page	41

Introduction

Information technologies and solutions have surged in the past years, with products like 3G connection and smartphones, or the development of new languages and frameworks. The challenges engineers are facing in the modern system have mutated from overcoming theoretical and practical implementation difficulties to optimizing user experience in a real-world situation.

The Chess game has been in the center of a major internal intelligence improvement in the nineties, above all *Deep Blue* who defeated the world champion Garry Kasparov in 1997. Unfortunately, the chess world did not continue developing alongside the emerging technologies. Currently, a lot of popular tools are deprecated, suffer compatibility issues, or nonexistent. This lack of development is especially noticeable around the service of organizing chess tournaments.

The described project consists in an in-depth analysis of the chess tournaments global system. Using methods like contextual inquiry and SEAM modelling, the goal is to identify what solution or solutions could be provided to help tournament organizers in their tasks: handling player bases, real-time tournament management, results transfer...

We will first talk about the studies done to correctly identify the different actors and their gaps. The second part of the report will describe the potential solutions analyzed and evaluated. Finally, the selected solution will be described more precisely and from different approach points such as the digital strategy, the business part and explanations about the implementation.

Project Timeline

This section will act as a reminder of the timeline of the project during the semester, and the different thought processes and phases.

Preparatory work: follow ESOA course, get project idea from personal experiences, define project scope and objectives, prepare project description, strike agreement with LAMS for a formal project opportunity, preparatory study of the environment

Week 1

Work on the initial project idea: on one hand, understand pairing algorithms and parameters to prepare development of the pairing software, determine current state-of-the-art algorithms and software for this process.

On the other hand, develop a model for the system and try understanding the different actors and their gaps. Contact federations, chess clubs, tournament organizers, collect data from a poll on the current usage of the available programs. Understand how FIDE regulates and certifies software and algorithms.

The idea for the project has now evolved to a distributed system with user accounts, which currently does not exist (or very minimally). Defining different ways to modernize and provide future-proofing. After polling, the idea of distribution and web-based software widely regarded as undesirable.

Week 5

Contextual inquiry at a chess tournament in Echallens. Lots of new knowledge on how the system works, two major points: the idea of a non-local software is completely dropped because of general impracticability (connection and hardware issues). Realization that the user gap is not as expected and as polled: although users complain of the general user interface and user experience of the pairing programs, they are objectively functional. Realization that development of a pairing program implies wide competition and massive development efforts.

Although, a new gap has been spotted. A large loss of time and wasted efforts go into subscriptions to tournaments. Organizers must organize their own publicity, create their own registration systems, handle payments by hand, and input players into the pairing programs from asynchronously downloaded player databases. Analysis of all the new knowledge.

New information on the general status of the Chess world, political implications of the FIDE and its core members, and the state of crisis caused by a sudden change in president and loss of financial security.

Week 6

Definition of a new direction for the project: developing an API and a web application for tournament management and registration. Contacting the FIDE with the goal of offering to officially integrate said API into their system, for use by all players.

Research into what features and what challenges await this direction: user verification, large data structures, worldwide accessibility...

Week 8

Mid-semester presentation with LAMS members, feedback from multiple points of view.

After lack of positive answer from FIDE, contact with pairing software organizers and design of an alternative to the project: instead of a major partnership with FIDE, provide a service completely external to the federations with one-way synchronization.

Contact different pairing software developers for ideas and more information on possible partnerships. Positive feedback from Luigi Forlano, developer of Vega. Refinement of the second alternative for a concept with a strong partnership with Vega, offering direct import into the software.

General research into implementation alternatives and choices to make.

Week 10

Refinement of the new project alternative. Research into business potential of the idea, development of the BRD and DSR documents.

The new alternative is now looking to be the more promising of all, getting generally positive feedback from most sources. More time is now spent on thinking about how to insert it into the market, how to draw users to it, what could be its place in the near and far future and how to reach strategic goals.

Support from Vega is a key point in the reflection, providing an already very large and international user-base. Strong support from its developer implies good publicity for the market penetration, and independence of choice and development from the lack of partnership with FIDE implies shorter development time, less drawbacks, and less inertia from the quantity of actors in the project.

Week 12

Development of a prototype of the third alternative. More research into the actual development choices and strategies, required infrastructures...

Preparation of scholar project finalization and writing of the report.

Inquiry

Entering the Chess World

In Switzerland, there are around 6000 chess players who play for more than 200 clubs. Most clubs organize annually one or more tournaments open to all players, alongside their internal tournaments where only the club members face each other.

All these clubs are overseen by the Swiss Federation (*FSE*) which handles the players ranking (*ELO*) and organizes national tournaments such as the Swiss Championship. Team competitions are also organized by the Federation. There are two main team competitions with five leagues for each. The first league is semi-professional and some of the best players in the world play there.

Each club has one or more teams registered in each of these competitions. If the smallest leagues are composed of regional groups, the bests are composed of teams from all the country.

For this project we were in contact with the Nyon Chess Club, one of the best clubs in Switzerland. Vice-champion of Switzerland, Nyon has 10 teams engaged in different competitions (for adults and juniors). The club also organizes one of the biggest active-chess¹ tournament of the country where more than 130 players battle on the chessboard. This kind of event is composed of multiple sub-tournaments, usually one per categories of age. This rally takes weeks of work to set up: defining the prices, advertising the event, handling subscriptions...

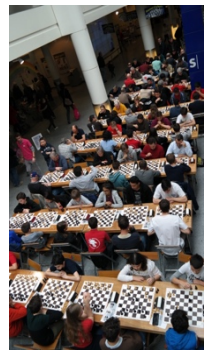


Figure 1: Nyon Active Chess

The Nyon Chess Club gave us all the required information about the organization of a tournament. They also provided contact points to some others big clubs in western Switzerland such as Payerne, Echallens, Geneva or Lausanne. These new contacts allowed to extend our knowledge and to gather multiple opinions about tournament management.

But before talking about how tournaments are working, let's investigate the federations and how they operate.

¹ An active chess is a tournament where players have little time to play their game (i.e. between 10 and 20 min for each player). An active chess tournament often takes place during only one day.

Chess Federations

Like in almost all sportive or associative domain, there is a national federation in each country. All manage their associations the way they prefer and all have a different national ELO ranking, often based on the same system but that can also differ. All these federations follow the general directive given by the international one (*FIDE*). Like the others, the FIDE handles its proper ranking, which is the international ranking that can be used in all international events. The FIDE also decides which program can be used to organize a tournament. If some details in the regulations can vary from a country to another, the main rules are made by the FIDE. So, they are also in control of the referees' formation and licenses.

Any club can organize an international event, they just need an agreement from the federation (easy to have) and to pay a license. Like the national federations, the FIDE also organizes some tournaments but only the biggest events: Candidates tournament², World Chess Championship, ...

Lately the FIDE has had some internal troubles. On April 2018 the federation saw their bank account shut close by UBS. "According to FIDE officials, UBS is terminating its relations with FIDE because of its president Kirsan Ilyumzhinov's presence on the sanctions list of the U.S. Department of the Treasury."³ This was followed by the election of a new president on October, which is contested by a lot of people due to Vladimir Poutine's support. Together, these changes made the FIDE difficult to get in touch with and not very cooperative during this project.

How A Tournament Works

Even if plenty of variants exist, the large majority of tournaments follows the Swiss System. Each player plays the same numbers of games. At each round, they will face a player with the same number of points. If they win, they will earn 1 point, $\frac{1}{2}$ for a draw and 0 for a loss. At the end of the event, a ranking is done which takes in account the player's number of points and their *BuchHolz*, a nice German word which means the sum of your adversaries' points. This is a nice way to numerically describe the difficulty of a player's competition route. If BuchHolz are not enough to decide between two players, other tie-breakers are used. However, we will not talk about them because they are more or less complicated to explain and not really useful for the following. This still shows that it is nearly impossible to deal with all those features by hand, even for a little tournament.

Pairing players for each round is also quite difficult to do manually. After each round, an intermediate ranking is done, which is required for the pairing process. In the simplest way, this is done by taking sub-rankings for each categories of points (i.e. all players with 1, 1.5, 2, ... points), dividing the table in two parts and making play the first player of table 1 with the

² Tournament which decide the contender who will play the defending world champion, more information on [Wikipedia](#).

³ See full article on [chess.com](#)

first one of table 2, the seconds against each other and so on. This is shown on the schema below, where players with 1 point are paired against each other.

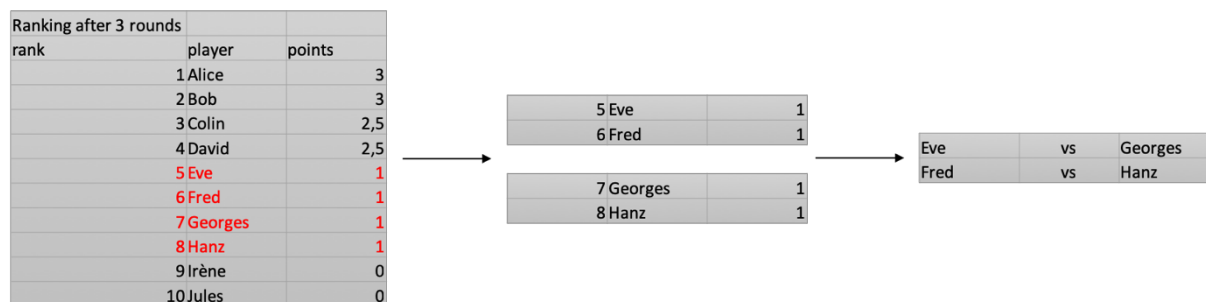


Figure 2: Little example on how a pairing works

Unfortunately, this is not so simple. Players who have already played against each other during the tournament cannot play a second time, even if colors are switched. Colors are also important; a player must play as many games with whites than with blacks⁴. If there is an odd number of players in a sub-ranking, someone will play against a player from another sub-ranking (often with someone with a half point more or less). This makes large tournaments impossible to handle by hand, pairing programs are thus there to help organizers, and can even be considered a requirement when the number of players grows to a few tens.

Usage of a Pairing Program

In this section, we will only talk about SWISS MANAGER, which is the most popular pairing program in Switzerland. The logic behind all those programs are the same, explaining how one works is enough to understand how all of them work in general.

The first step is to enter the tournament information: name, referees, dates, ... This information will be useful because it appears on all files relative to the event and can easily be communicated to the federation. Other, more technical details are also asked, such as time control (how much time a player will have for a game), categories, if the tournament uses national or international rating or both, ...

⁴ Most of the time, tournament an odd number of rounds. In this case, players must only play one more time in a color than in the other one.

Tournament Data Dialog (Swiss System) X

General Tiebreaks Lists Board List FIDE Title Other Arbiter

Tournament Name: EPFL Cup

Remarks:

Organizer(s): Swiss Federal Institute of Technology, Lausanne

Website: www.epfl.ch Email: noreply@epfl.ch

Time Control: 1h30 + 30m ko

Tournament Director: Colin BRANCA

Chief Arbiter: Jules COURTOIS Deputy Chief Arbiter:

Arbiter:

Federation: SUI Switzerland

Location: Lausanne

Rounds: 5 Date from (yyyy/mm/dd): 2018/10/10 to: 2018/10/13

Age Groups: 8,10,12 Cutoff Date: 2018/01/01 Rtg.min for Rtg-Ø: 1000

Sort start rank automatically ☒ Check board order ☐

Replays (e.g. Double RR = 2): 1

Points for the Bye Player: 0

Sorting/Display:

- ☐ National rating
- ☐ International rating
- ☒ Int. rating then Nat. rating
- ☐ Rating maximum (Nat./Int.)
- ☐ National rating only
- ☐ International rating only

Color for home game:

- ☒ White
- ☐ Black
- ☐ All boards the same

Pairings according to:

- ☒ Game points (1, 1/2, 0)
- ☐ Match points (2, 1, 0)
- ☐ Game points (3, 1, 0)

Rated FIDE: ☒ Yes ☐ No

Rated national: ☒ Yes ☐ No

Match points for bye team:

- ☐ 2
- ☐ 1
- ☒ 0

Copy tournament data

? Help ✓ OK ✗ Cancel

Figure 3: Tournament Creation Page

The tournament information set, the organizer must now enter the participants. This must be done by hand. The program proposes to download rating lists from some national federations and from the FIDE, so if a player is listed in the federation, entering his license number suffices to extract all their information. But even with the existence of this feature, this step can be quite long when entering a lot of participants.

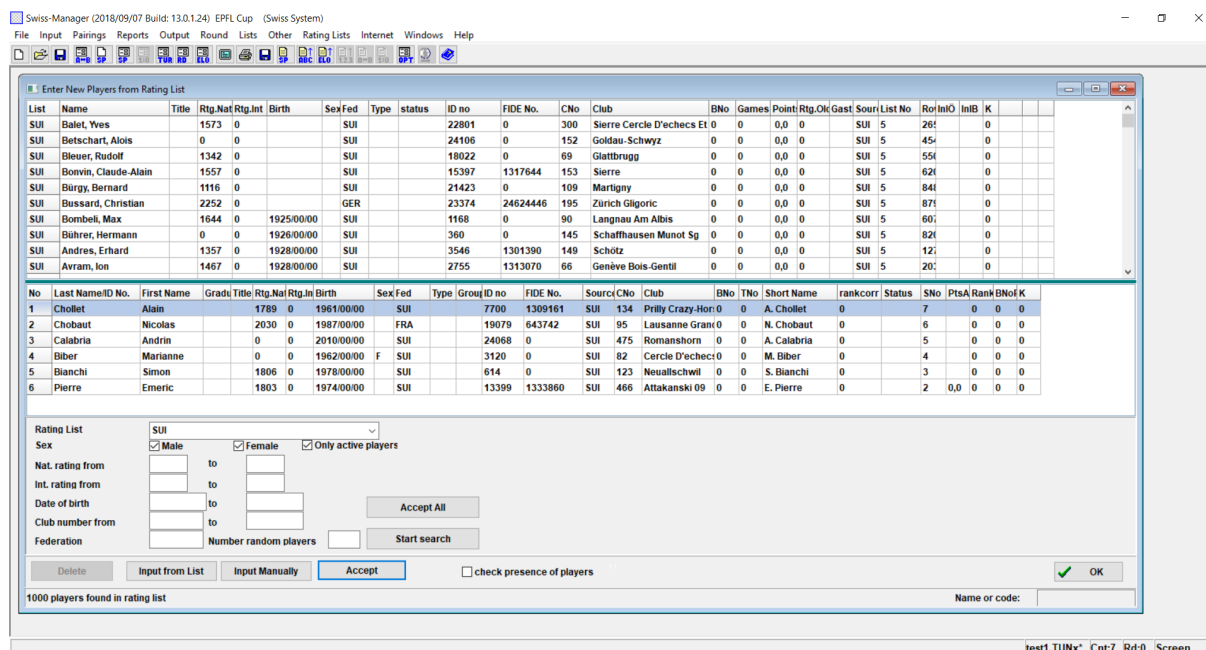


Figure 4: Page for adding players into the tournament

This step done, the tournament can begin. The starting list, where participants are sorted by their ELO rating, is used as initial ranking for the first round.

At the end of each round, the results must be entered manually on the program, this is in theory very simple, but it needs in practice a lot of concentration. Indeed, when 100 players are waiting for the next round and asking questions to the referees when they are entering results, it can be difficult to not make mistakes.

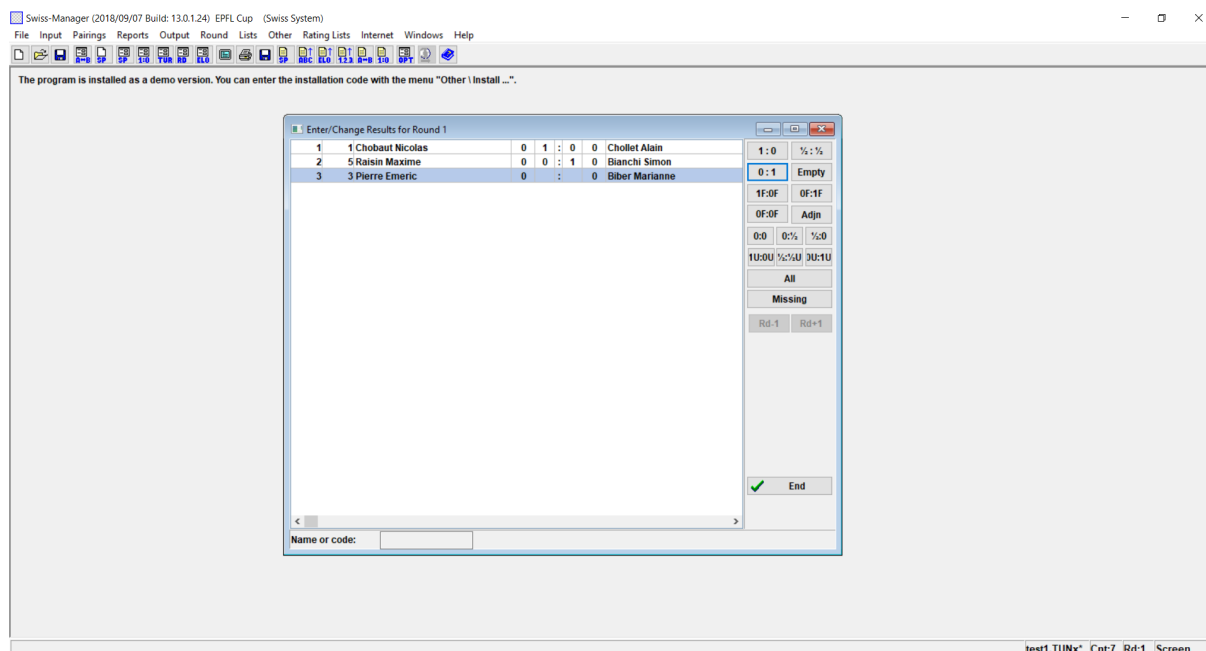


Figure 5: Results Entering Page

Clicking the END button allows us to compute the next round. At each round, an intermediate ranking containing all wanted information, can be generated and printed.

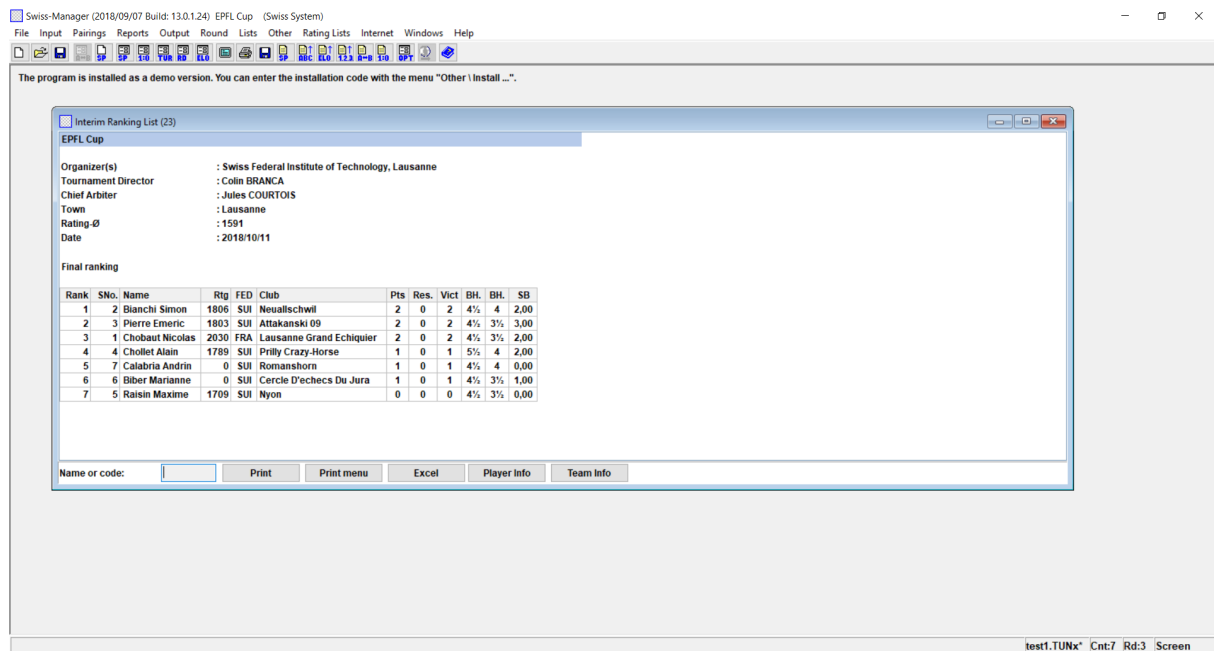


Figure 6: Ranking Page

At any time, inputs can be modified but this would have an impact on the different outputs. Therefore, some modifications are difficult to do, like introducing a new player in the middle of the event.

Operation of the Chess World

To fully understand the needs and gap of each actor, our first step was studying the behavior of the system: how chess tournaments are organized before the event, how the event is managed and ran, and how the different actors act and interact.

We identified and chose to differentiate four main actors with different roles and interests.

- Players: they are maybe the less concerned by but also the main reason of such a project. Indeed, a participant only wants the tournament to go well and is not interested by the backend, but on the other hand, there were no event without participants.
- Tournament organizers: they are the ones to be convinced by the project, being the major targeted sponsors. They are the most affected by the lack of useful tools. Currently, they must enter every player by hand in the system, this can take a lot of time for a big event, especially because this should be done 3 days or less before the event, to ensure all rankings are up-to-date.
- Pairing programs and their developers: organizers use one of the endorsed pairing programs to manage their tournament. The developers form the third important actor. As will be explained later, they would have been competitors in the first project idea but will be partners in the final solution.
- Chess Federations: the FIDE (World Chess Federation) is the highest instance and decides which pairing system can be used. The international federation has a database

of all players with their FIDE ELO ranking. The national rankings are stored by each national federation.

When a tournament is created, the organizer must announce it to the national federation and the international one if he wants his event to be worldwide. The federations will not interfere in the organization, they will only ask for the results of each played game at the end of the tournament. The organizer must now make his publicity and handle the subscriptions in the way he wants. He will have to enter each player in the pairing program he decided to use⁵ and can then begin his tournament. At the end of the event, the organizer must go through the process of exporting the pairings and outputting them in the desired format to the federations, in theory this part is quite automatic but in practice this always take hours.⁶

Organizers' Opinion

Knowing that the major objective of the project is to convince the tournaments organizers, whom who have defined as the sponsors, their opinions have a large weight and must be considered. They are also in the center of the interactions between all the other actors and so have the most universal point of view. To interrogate them, a survey has been done and sent to all presidents of chess clubs and tournament organizers of western Switzerland and neighboring France. A total of 16 answers were collected, which was enough to draw some conclusions.

A large majority are using SWISS MANAGER and are quite satisfied with it. According to them, its only defects are a bad user interface, hard to take in hand at first, lack of possibilities for modifying pairings but above all the impossibility to import participants.

They were also asked about the missing features, four were heavily represented: automatic import of participants, possibility to share pairings and results online, automatic handling of special prizes (first woman, best performance, ...) and direct access to players' information.

A large majority also do not want an online solution. This can be surprising especially in our days where quite all is done online.

Contextual Inquiry

The final step of the study was a contextual inquiry. The Echallens Chess Club had the amiability to accept us for their active chess tournament on October 7th, 2018. The observations have been done during two key moments of the event: before the first round and between the last and the awards ceremony.



Figure 7: Echallens Active Chess

⁵ In Switzerland, only SWISS MANAGER is used for big events, SWISS-CHESS is rarely used and only for little tournaments. Other ones and quite never used.

⁶ Notice that the practical aspects such as finding the material, a playing room, finding the referees, ... are not discussed here because they are out of the project field.

Those two phases are the most important when organizing an event and the most stressful ones.

At Echallens, only half of the participants registered online before the event. The others came one hour before the beginning to subscribe. There was no possibility to pay the subscription fee online, so every player had to come to the referee office to pay it. In all, these factors make this moment a key one that must be well managed to begin the first round on time. Although this was quite well handled by the organizers, the tournament began 20 minutes late. Some players who came at the last moment and some last-minute changes caused it. At the end of the day, this delay is not significant, however the first arrived had to wait for more than 1h30 and the delay was a source of stress for the organizers all event long because they wanted to catch up to it.

Once the first round was launched, no difficulties were encountered. The tournament was moving forward. At each round, the pairings were printed and shown to the players. There were four categories at this event, so four sub-tournaments and so four different pairings. After having seen the pairing, each player went to the table he was indicated and played their game. When the match was completed, the winner (or the whites when in the case of a draw) would go to the referee table to announce the result. The following round would begin a few minutes after the end of the last game of the current round.

After the end of the ninth and last round of the tournament was the most important and stressful section. The goal was not to make to players wait unnecessarily and so to begin to award ceremony as soon as possible. Even if the rankings of each categories were available automatically, each category had special prizes and those were not handled by SWISS MANAGER and so had to be done by hand. If the program could have dealt with those prizes, it would have saved 15 minutes to the organizers.

Although the event was a great success, a lot of details can be corrected to facilitate the organizer's lives. A better subscriptions management is strongly desired by the tournament director. This is also the case for the possibility of handling special prizes.

Even if it would have been benefic for such an event, a distributed system which allows to send the pairings to the players and to receive the result would not be possible in this playing room. No WIFI was indeed present and the mobile web access was really limited.

First Project Idea

Goal

The first project idea was to create a brand-new pairing program, more user-friendly, distributed and directly connected to the different federations. The server would be directly connected to the federations and could import automatically the players information into the server's database. Each event would have been stored definitively and made accessible online. The program would also handle player subscriptions via a dedicated web page where each user would have a profile page. The whole event could be managed online or locally.

The goal is to have a user-friendly tool, with all existing features and especially most wanted ones such as subscription handling, results sharing, special prizes computation and direct access to player profiles.

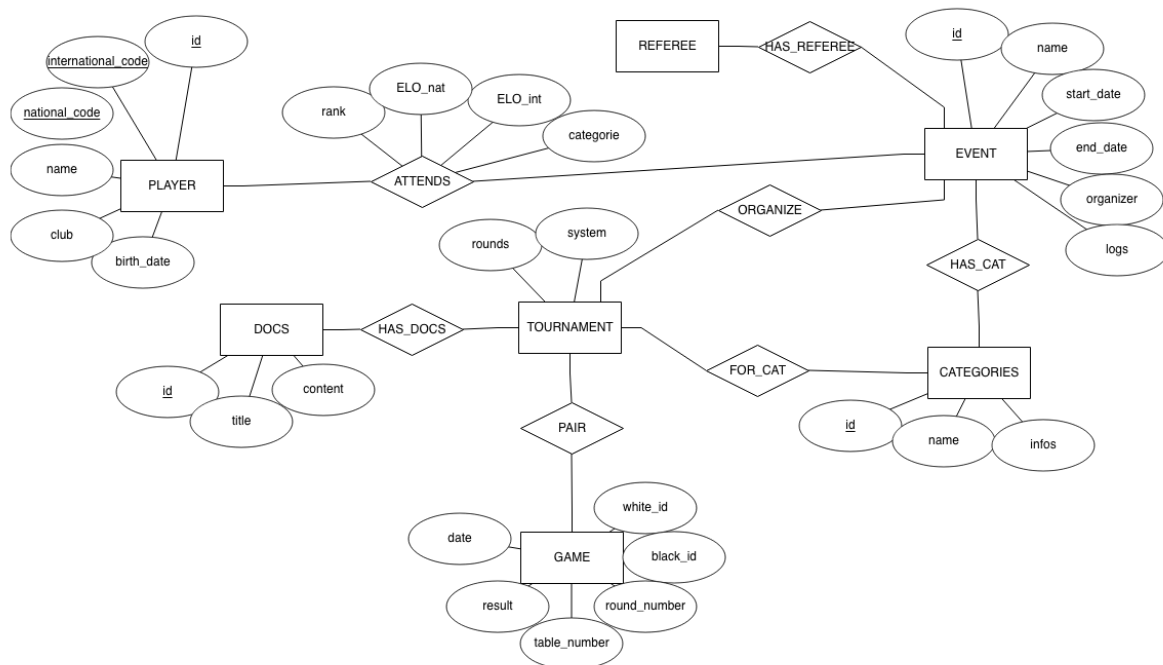


Figure 8: Database Schema of First Project Idea

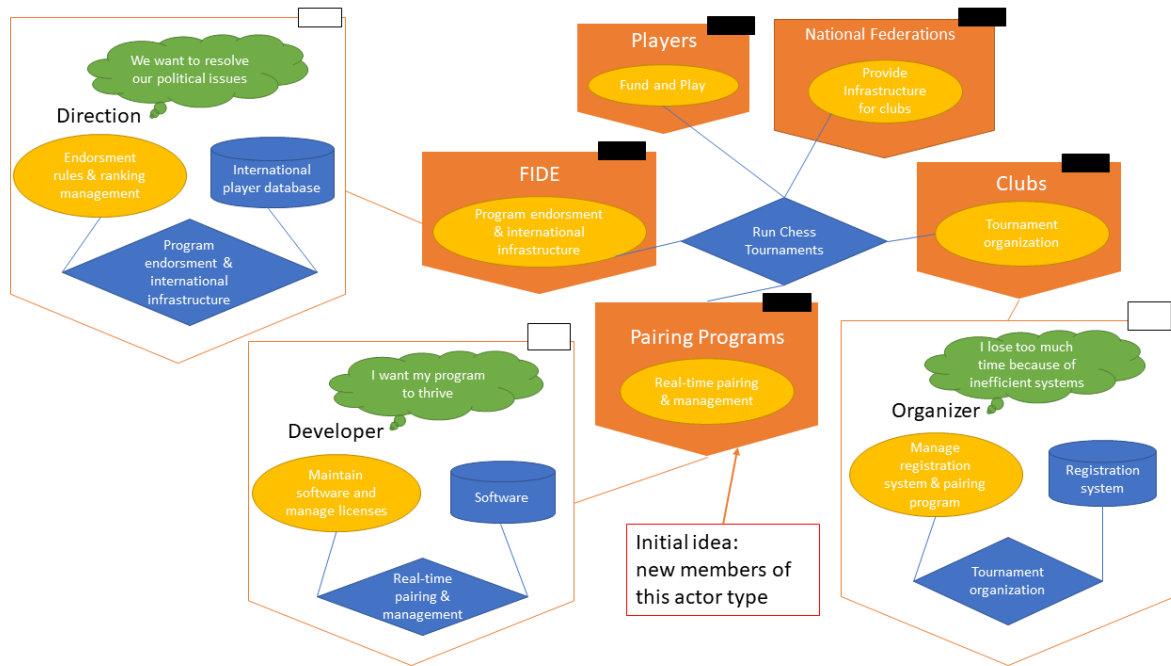


Figure 9: Place of the initial idea in the system

Difficulties

The pairing engine implementation was a far more complicated and regulated process than expected. Only two engines are currently being used: JaVaFo which runs on java and bbpPairings on C++. The requirements are not explicit on the FIDE's web page. Contact was established for further details and information; the FIDE simply provided a redirection to the official but unclear pages. After careful evaluation of the two possibilities (i.e. developing a complete, authorized pairing engine or using a pre-existing engine), it was chosen that integrating an existing engine was the right course of action. The larger turning point for this choice was the fact that, as opposed to real-time tournament management software like SWISS CHESS discussed above, pairing algorithm engines are well-implemented, rather recent (especially bbpPairings, 2016), and although they would require a strong effort to integrate into the codebase, it would still be a much lower development toll than creating and integrating from scratch an engine.

Another big issue was the firmness of the federations regarding new pairing engines. Only the FIDE can endorse a program and were not cooperative. The requirements are high, as can be expected from an international organization, and it was noticed that their behavior was leaning towards conservatism of the current state of things. M. Herzog (SWISS MANAGER developer) was also contacted to see if a collaboration could be possible to potentially create an extension to their product, providing distributed features, but they did not answer.

The lack of answer from the international federation is also partly due to their crisis. On the October 3rd, 2018, the new FIDE president (Arkady Dvorkovich) was elected after some financial problems linked to the previous one. This nomination is questioned in particular because of Vladimir Poutine's implication, Dvorkovich being a former Russian deputy.

Finally, some studies showed that a large majority of tournament organizers manage events every year and even if SWISS MANAGER is difficult to use at first sight, it is made for regular users. Those pairing programs are maybe not ideal, and perfecting them would be an interesting potential solution, but entering competition was not: developing from scratch a market position and finding innovators to sponsor it, although possible, was definitely far from an optimal way of sanitizing the system at play.

Explanation of ChessHub

Overview

All these encountered difficulties brought to light what is really missing and caused a change of direction in the project. Pairing is not the real problem, however a tournament management tool that link players, federations and organizers could be welcomed. The idea is therefore to develop a web tool and an API that can do this job. This new idea falls precisely in line with the core strengths of the development team: new technologies providing ease of implementation and interesting features, allowing for more focus on the user interface and experience. Being a complete new actor to the system, this concept also provides strong attractiveness and a head start compared to potential competition. With enough agility in the system, it can safely be described as a full-market opportunity, and even a market monopoly until potential competition started developing another tool.

Each player creates an account on the *ChessHub* site and give their information, in particular their different license numbers. The server is directly connected to the different federations and so can check if the given information is valid. It will also periodically update the player's ELO ranking.

Their account made and verified, the player can now subscribe to the available tournaments for which they are eligible and pay the subscription fees online, securely, in advance.

Furthermore, for each tournament he attends, the player receives some information such as rankings, pairings or simple communication.

For his part, the organizer creates their tournament as usual on their favorite pairing program. They can afterwards publish it on the *ChessHub* platform where some other information can be added: image, date, place, prizes, categories, maximum number of participants, ... The tournament should also be creatable directly from the web application.

At any time, the organizer should be able to import the participants list to his pairing program. During the event, they can also publish rankings, pairings and other information online so that it can be accessible by the participants.

At the end of the event, the results are transferred automatically (or semi-automatically, maybe requiring a simple user confirmation) to the concerned federations and can also be shared online.

To make it possible, existing pairing programs must do some implementation to communicate with *ChessHub*. A clear, well-documented API with strong and expansive features should be available to make it as easy as possible to find an agreement with developers.

Objectives

Even if this idea is not revolutionary, such a tool does not exist in the chess world. Some tournament organizers, players and federation commissioners were asked about the usefulness they would find with a tool like *ChessHub* and the answers were all positives. This shows that the target is well defined and that if the project is well conducted its result can be used by a lot of people. It remains now to convince every actor to work with this program.

Players will only use the web platform. To convince them, the UI must therefore be user-friendly, simple to use and secure (payment will be done via the platform). Older or more conservative players will be difficult to persuade but if the young ones, more familiar with this kind of applications and systems, use the tool the number of users can only grow year after year. More explicitly, early adopters would be more modern tech-oriented players and tournament organizers.

ChessHub will be free for players but not for organizers, the real sponsors. The tool is clearly made for them, so it must satisfy them as much as possible. To do so, all interactions with the tool must be easily doable via their pairing program and on the web app. So, partnerships with the different pairing programs owners are key to the success of the software. Users of partner programs would be the innovators, being offered facilitated access and potentially advertised the idea by the software they already use.

For now, only VEGA's developer Luigi Forlano answered positively. His program has the largest national market share in four countries: Italy, Spain, New-Zealand and Australia, which represents more than 200 tournaments per year. This can be a strong start and a way to receive usage data and perfect the idea before finding agreements with other pairing programs.

Finally, the different federations must be considered. Every federation gives a public access to its players base. Unfortunately, the way to get them is not generalized so the import script will be different for each one. For now, the FIDE does not seem to be interested by a collaboration to universalize the players base access and the results transfer. The FSE is more interested asked to be kept up to date concerning the project.

For all these actors, the objectives and difficulties can be split in two domains: the business part and the technical part. They will be approached in the next chapters.

Business Part

Roadmap

Concerning the business part, the project must progress in two different dimensions: negotiations with providers and marketing.

Concerning the providers, an agreement has already been found with VEGA. The first steps of the project will so only be done with their collaboration. The objective is to obtain a working tool used in the different countries where VEGA is used. This step done, other pairing programs developers will be approached to also find an arrangement to extend the *ChessHub* userbase.

To increase the number of users, first a “top down” marketing approach will be used. VEGA users will be contacted to be briefed about the tool. VEGA will also offer organizers to register the tournament on the *ChessHub* server. When using the app, organizers will ask the players to also create an account on the program web page. The chess world being a small environment, if some big events use it, a large majority of players will be registered.

This step done, the marketing approach will be changed to a “bottom-up” one. Indeed, if players are convinced by *ChessHub*, they will ask to the non-client organizers to also use the tool. Some discussions with chess players showed that they would appreciate a simple and universal tool that allows them to subscribe to any tournament.

External Objectives

The short-term goal is *ChessHub* to be used in the majority of tournaments in countries where VEGA is used (Italy, Spain, New-Zeeland and Australia). An agreement with Heinz Herzog (SWISS MANAGER) is another objective, it could allow to enter the swiss market. The swiss market entered, some negotiations with the FSE would be facilitated.

In a long-term view, an agreement has to be found with all FIDE approved pairing program. This would make *ChessHub* an universal and international tool. With this, talks with every chess federations, especially FIDE, would be possible.

Internal Objectives

Currently, only two developers are working on the project. This is sufficient to achieve the wanted results. If the project progress as wished, another employee could enter the team to oversee maintenance and scripts writing.

In three years, the objective is to be fully established in the four countries where VEGA also is. Partial establishments in some other countries like Switzerland and Austria are also wanted. The goal is to be implicated in more than 150 tournaments with more than 20 people per year.

In seven years, establishment in more than 20 countries and to implication in more than 500 tournaments per year are desired.

Business Requirements

External Situation As-Is

Because of recent events (October 2018), an important external actor has been revealed in the form of banks and international politics. Because of allegations against the previous president of the World Chess Federation (FIDE), all the federation's accounts at UBS have been closed. This is a drawback, requiring much attention from the new president and committee to keep the federation running while administrative and financial solutions are being found. This situation influences the possibility of cooperating directly with the FIDE and might reduce its potential as a sponsor to the project.

The FIDE also provides a service to retrieve a list of all official players, as well as some information like their ELO and unique identification number. This is done through a simple HTML webpage. Tournament results can be sent to the federation by email for manual input into their database. The Federation also endorses tournament organizing software according to precise requirements and guidelines.

Clubs use tournament software to create, organize, and run tournaments. There are currently 6 endorsed programs with relatively small differences, which all fill their roles correctly. The most notable, for the scope of this project, are Swiss-Manager (SM) and Vega. Such programs are developed and sold independently from any other system, by the developer to clubs. They all run locally, with some distributed solutions for pairing (SM) or entering results (Vega).

Clubs also receive tournament registrations using an external method (ex: Google Forms, simple email registration, dedicated website...) for manual input into organizer programs.

Organizer programs also output a tournament's result file which clubs must send by email to the FIDE upon completion.

National federations (like the FSE in Switzerland) provide services to retrieve the list of players. Such a list can be used in the programs to lighten the load of manual input and simply retrieve player info using a search query of the license number or name

Business Internal As-Is

The injection of our solution is at the crossroads between Federations, clubs and players, offering a centralized tournament registration system. In the current state of being, Federations provide their list of players independently. Organizers must design and provide

their own system to receive tournament registrations, which players must be aware of and follow to successfully register. Finally, organizer software provides the link between Federation and organizers, through the local importation of a list and manual comparison.

In the current state of things, there are no official ways to distinguish a real subscription from a malicious one, except by two checks from the tournament organizer (first, the player does belong to the federation he claims to belong to, then the player truly is who he claims to be). This is a massive security pitfall. In practice, the second check is almost never made, which opens the door for falsification of player results.

Currently, there is no tool similar to *ChessHub*. According to Luigi Forlano, developer of Vega, this project has been an idea of his (but at a national scale, within the Italian Chess Federation) for multiple years but never went into development. We did not hear from any other project like this. Even if a similar program exists, none of the people we talked with knows it, this can confirm that *ChessHub* does not seem to have adversaries neither in Switzerland, France nor Italy.

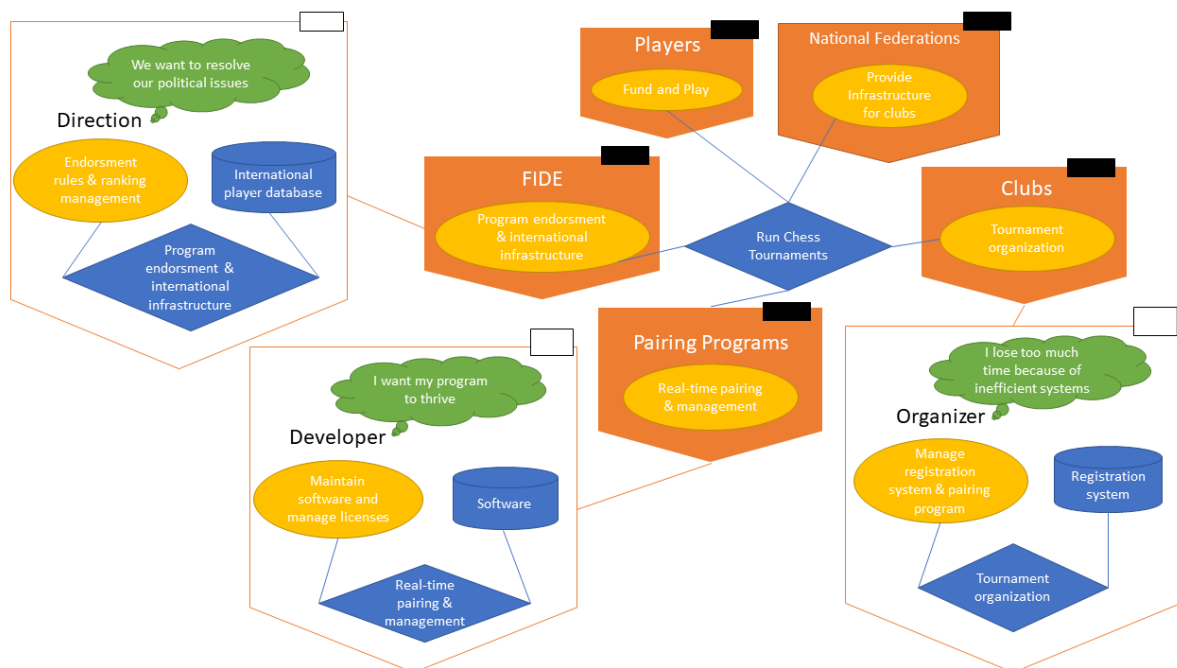


Figure 10: As-is situation

To-Be Selection

Alternative 1

The first solution alternative describes a situation where the FIDE does not wish to partake in the development of the solution. An API will be developed which would allow users to authenticate themselves, organizers to create tournaments, and players to subscribe to tournaments from any platform in contact with the API. The creation of such a platform (as a Web-app), would be part of the solution, and would also come with extra features like online registration payment.

A new API will be created, running permanently and independently from any other service, program, or federation. The API should automatically pull new data from the different Federations' services. Player authentication will not be guaranteed because of the lack of cooperation from the FIDE. Instead, organizers will have an authentication system to create, manage and import tournaments.

Organizer programs need to create a link to the newly created API. This allows them to directly import entire tournaments, and completely diminish the load of this operation when opposed to the current state. After discussion with Luigi Forlano, developer of Vega, this seems to be a feasible implementation and extension to organizer programs.

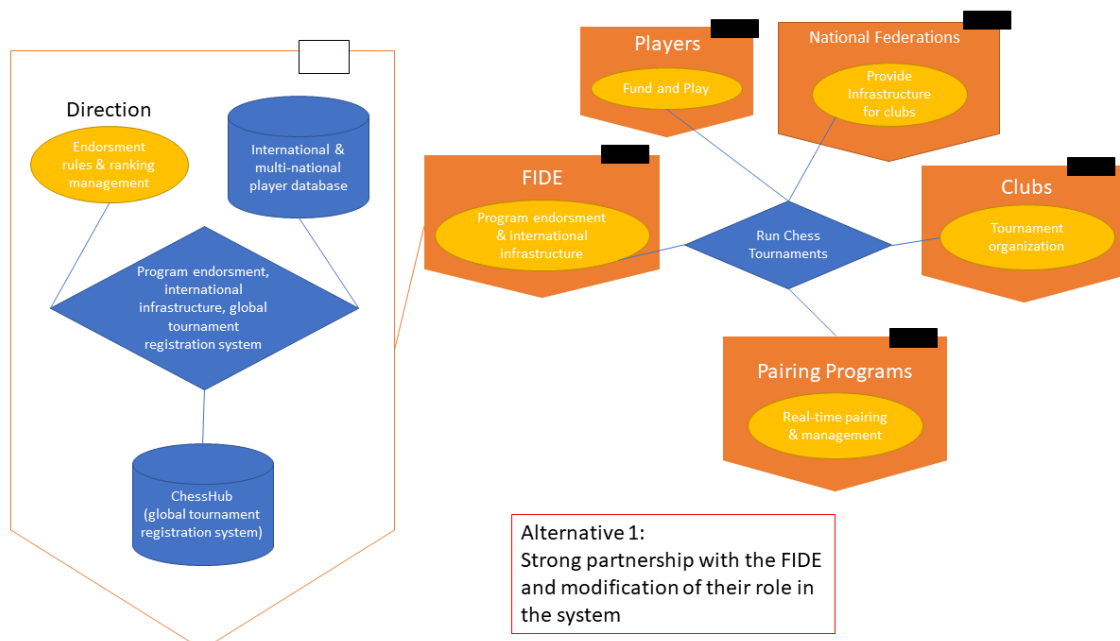


Figure 11: First Alternative Diagram

Alternative 2

This solution is akin to the first alternative, except a collaboration has been struck with the FIDE.

With complete collaboration from both the FIDE and organizer programs, most of the internal project would be designing the API and plugging it into the service offered by the FIDE. The creation of a user interface would still partake in the process of an efficient environment.

This collaboration would take shape in the creation of a full-size authentication system: players would now get accounts upon their registration to the FIDE. This would necessitate a password recovery system, password delivery system, potentially modifications to the way registrations are currently validated and digitalized. This system could be developed either by the FIDE, or by an independent developer, but would probably necessitate a dedicated employee at the Federation.

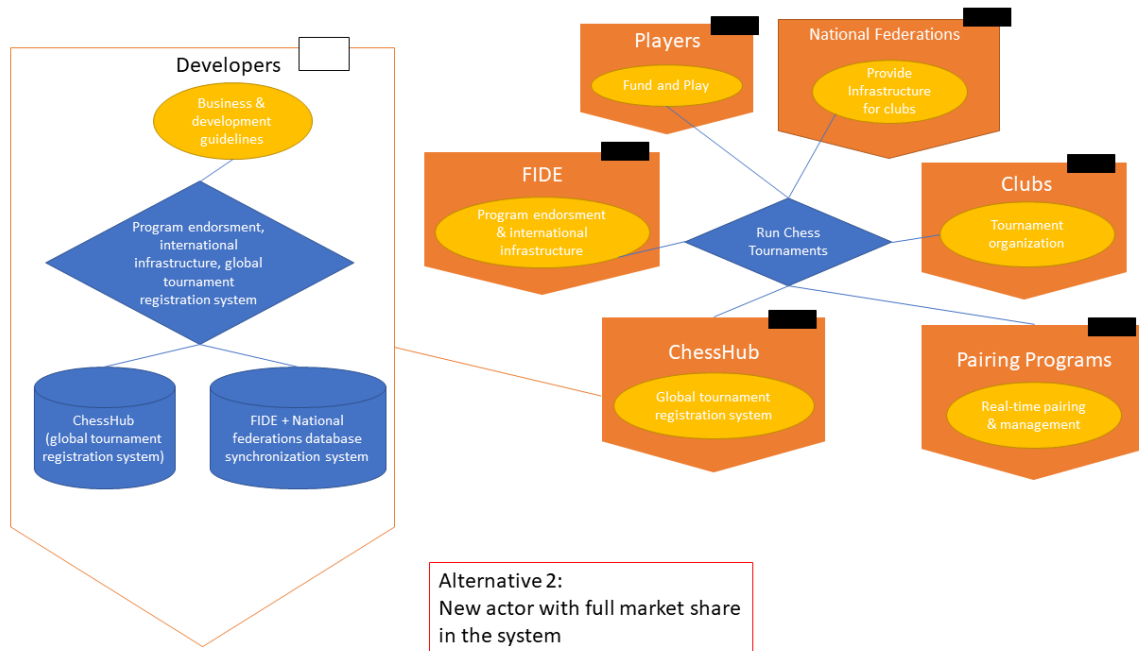


Figure 12: Second Alternative Diagram

Solution Selection

The first solution was selected. Although being a harder technical challenge to overcome, this selection mainly came to be because of the difficulty of striking a good agreement with the FIDE. Apart from obvious communication barriers (language, distance, experience...) there were also, as explained before, strong events which the FIDE had to handle with higher priority than the described extension of their information system.

The main advantage of the second alternative is that it does not exclude a future partnership with the federations, this would be done with help of our API that can already connect to their web services to get the players lists. Another communication with their server in order to upload tournaments results or other information can be done. Our prototype is already implemented with an architecture that can integrate this feature in the future.

The development of the first solution, without any limitations on the implementation, will have an estimated development time of at most 100 man-hours. Such a workload can be delegated as a semester project for a student, reducing development costs to nearly nothing. The largest part of the investment will come from the rent of the cloud services.

Three business models could potentially be implemented:

- Pay-per-use: every time an organizer wishes to create a tournament, he has to pay a certain amount depending on the maximal number of participants. Although this would be the most flexible and with the most benefits, we would risk losing clients of certain types: small clubs who organize many small tournaments would find it too expensive, or large organizations who organize massive tournaments every year for long durations.
- Subscription: this is the less likely to work, although it could potentially interest the clients described above, subscriptions for online services tend to discourage clients for the fear of losing their investment.
- License: at the opposite of the subscription, a timeless, one-time-payment solution could be the good halfway point between profit per client and overall number of clients.

What we are thinking of truly developing would be offering a choice to the organizers between pay-per-use and timeless license. The idea is also to provide a light version for free, that would allow to manage a tournament with a maximum of 20 participants in which some features, like communication with players would be missing. This would let tournament organizer testing our tool before investing money in it and also convince them that they need this software.

To-Be Description

Concerning the external business part, an IT service will be provided for each actor. As said before, each party has its requirements. In order to be used by the organizers, our tool has to satisfy the needs of each actor. If only one actor does not take part to the project, the purpose would be totally defeated. The organizers want to work with a program that allows them to communicate with every other parties, if they are asked to use another tool to, for example, send take the subscriptions because players do not want to use our web interface, *ChessHub* won't be used. The objective is therefore to satisfy every actors.

- FIDE / National Federations: The Federations will hold all player data. With no partnership or affiliations with any federation, these actors will have no direct influence on the project. Their webservices are usually public, accessible through http, and allow the retrieval of rankings, names, and identification numbers. The greatest interaction will be having to update the software if the webservices are modified, since there is currently no regulation, feed of updates, or standard for the services.
- Pairing programs: Support for the *ChessHub* API will be directly integrated into the pairing programs. This implies building a solid Rest API and good documentation, as well as providing an authentication tokens system. We have struck a deal with Vega. Thanks to the cooperation of Luigi Forlano, this will rapidly be done in his software. This first agreement will allow us to test our product in real conditions and to have a solid reference when negotiating with the other developers.
- Clubs: The clubs will be our clients, and our primary, if not only, source of return on interest. We can safely suppose that at some point in time, a club organizer will have access to the Internet and will use a modern browser, no matter their device or operating system. As such, to ensure the widest possible range of clients, we will need

to make the software accessible through a WebApp compatible with as many browsers as possible. This can be done with software like React.

- Cloud-based services provider: Supposing a provider akin to AWS, most management is done through a responsive and functional WebApp. This actor will hold all the data of the application and will handle incoming and outgoing connections from the clients and the federations' webservices.

As was described above, since all services are distributed, there aren't any completely internal actors or IT services. An important point would be developing a strong workflow diagram. Most cloud-based services allow for automated recompilation when a push is detected on *Github*. Such a feature would be very important for an always-up, production application, but would require training of new developers to the workflow. Resident developers will have access to the repository, and the WebApp will allow for a bug-reporting form to be sent. This will ensure that we detect bugs as soon as possible (for example, as said earlier, federation API changes) and can fix them under short notice.

Evolution

Before diving into this section, it seems important to bring some details to light. All the strategy described below will certainly not apply with this exact explanation in each country. As in every domain, each country has a different chess culture, other ways to structure tournaments organizations, ... What follows describes the strategy for most European countries but the type of tournament targeted can change from a country to another and the strategy can be adapted for each one.

Knowing that there are few communications between national federations, or just few players that attend tournaments in multiple countries, the first phase of our development strategy is to grow locally in different countries. The international reputation will exist only if *ChessHub* becomes the baseline tool of some nations. The roadmap will therefore follow this schema:

Firstly, we need to find an agreement with a pairing program developer and work on the integration of our tool in his software. For each country where this program is used, an import script must be implemented in order to get the players affiliated to the local federation. Contacting this federation must also be done in order to make our product be known by them and to potentially work with them to propose a result exporting feature.

Once this step done, the most difficult tasks lie ahead: convincing organizers to use *ChessHub*. We can categorize chess tournaments in four main classes. The little regional tournaments reuniting between 20 and 50 players, mid-sized national tournaments with around 100 players, big national competitions of 200-300 players and international ones with also more than 200 players.

Our strategy will be inspired by the theory elaborated by Geoffrey A. Moore in his book *Crossing the Chasm* (1991, revised 1999 and 2014). Which consists of dividing the costumers

in five groups: innovators, early adopters, early majority, late majority and laggards and focusing on only one group at a time.

If at first, small tournaments will certainly have a minor interest in the tool, we think that the mid-sized ones should be the first targets. They have enough participants to find *ChessHub* useful and also have enough budget to invest in it. The first step of our marketing will therefore consist of targeting this kind of tournaments to make them our early market (innovators + early adopters). Organizers of this kind of tournament would be advertised by their pairing program and also by email, a fast research on the web allows to collect email addresses of a good amount of tournament organizers since all events are listed on their federation's website.

In order to avoid falling in the Moore Chasm, which is the transition between visionaries and pragmatists, we will need to convince big national tournaments to use and trust our tool. Even if this step can be tall, if a lot of mid-sized tournaments use *ChessHub* and are satisfied by it, then transition could be done really smoothly. Regarding the fact that the chess world is a small domain and that finding motivated people to work and organize event can be hard, medium tournament organizers take very often part in the big events management. If someone can push the steering committee from the inside, the chances of seeing the tournament using the tool increased.

Like in other domains, there exist some online chess influencer. They are often chess streamers who are getting more and more views and who begin to organize some events for their fan base. *Blitzstream* in France or *Chess Brah* in the USA are two good examples⁷. Contacting them and trying to find an agreement with them for the publicity can also be a good way to make our tool known.

Concerning the coming three years, we want *ChessHub* to be used in the majority of tournaments in countries where VEGA is used (Italy, Spain, New-Zeeland and Australia). We also want to find deals with other pairings programs so we can extend our number of clients. We mostly want an agreement with SWISS MANAGER so we can enter the swiss market.

In a long-term view, we want to have an agreement with all FIDE approved pairing program. We also want to find some deals with different national federation and provide them a tool that can simplify their ranking system.

⁷ If you are interested we suggest you to visit their pages: [Blitzstream](#) and [Chess Brah](#).

Digital Strategy

Management Summary

As said before, our project is intended to be an intermediate between multiples services. On one hand we want to provide a user interface for participants and organizers. On another one we want to communicate with chess federations to imports user's information such as ranking, club, birth date, ... Finally, the program must also talk with pairing programs.

The UI must be user friendly, as easy to use as possible. For our tool to be used by players, it has to simplify the registration process and also has to be totally free for them.

Today, if a player wants to register to a tournament, he has to fill a form where he has to specify his personal information, to tell in which category he wants to play and he will pay the inscription fees with cash on the D-day. All this process can be simplified by stocking the user information on our database and letting him pay online. The registration process would be done in two clicks: one to register and one to pay. For organizers the UI must also be easy to use, creating a tournament must be done in as few steps as possible. The tool being principally designed for him to gain time, the organizer will have to pay to publish his tournament.

To have complete and up-to-date information our program must import ranking lists from the chess federations. Unfortunately, a unified way to import all lists does not exist. Every federation, national or international, has its own web service to access the data we want and has its own way to classify the data. So, we will have to develop a specific importer for each federation.

The communication with pairing programs is maybe the most critical point. For our tool to be useful it needs to be integrated to the used pairing programs. Its indeed assumes a bit of development on their side. We would like that those programs firstly implement the importation of the subscribed users. Other options maybe harder to implement, like publishing the tournament directly on our platform or sending pairings and rankings to participants, can be discussed in the future. We already have an understanding with Luigi Forlano (Creator of VEGA) who is motivated to integer this tool in his program.

For now, we only have an agreement with VEGA, so the objectives are to make our product used in countries where VEGA is also used: Italy, Spain, New-Zealand and Australia. In those countries, VEGA is used to organize more than 200 tournaments per year. The first goal is to make our tool used in the majority of those tournaments. To make the organizers aware of our product, we will pass by VEGA which will propose this offer and also contact the different federations to talk about it.

The development of the tool has already begun with the building of a prototype. Its goal is to get some feedback from different chess actors (players, organizers, federations, developers, ...) so that we will be able to clearly define the scope of the project and define which features need to be implemented first.

Currently, the first version of the program needs to have the following features:

- A user-friendly web app for player subscription and tournaments details including
 - An authentication system
 - A secured payment system
 - Up-to-date player profiles
 - Tournament profiles
- An optimized server and database synchronized to the different federations
- Import scripts for federations
- Import/Export scripts for providers
 - Subscriptions, rankings and pairings can be exchanged between the pairing program and our server easily

Again, we don't have funds to pay a developer for this task. We so have two possible ways to achieve the development. The implementation can first be done by ourselves but it could also be an interesting bachelor project for a motivated student, which is our preferred option.

Concerning the providers, we already have an agreement with VEGA. We will first work only with them. The objective is to obtain a working tool used in the different countries where VEGA is used. This step done, we will approach other pairing programs developer to also find an agreement with them so that we can extend our clientele.

Finally, we have to convince tournament organizers to use our tool. For this we will do some marketing. We first want VEGA to advise its users to use ChessHub, the chess world is a small environment so if some big tournaments use it, smaller ones will do the same.

We will also directly contact these big tournaments organizers to convince them to use the tool. We know that our biggest showcase is online, this is why having a nice and user-friendly web-app is necessary.

Strategic Challenges

Concerning the project IT part, the app architecture will be separate in two different parts: client and server. For the moment we will have two different types of client, the pairing programs and the web-app. We want our web-app to be implemented in Ruby on rails. The framework contains a lot of libraries that will be useful for us. So, we will use already implemented plugins to handle authentication and payment. This will save us a lot of work and guaranty security.

On the other hand, the pairing programs will communicate directly with our server and only have some set/get methods that will permit to create a tournament, import participants, ... All imported and exported files will be in .csv, their format will vary according to the program.

The server itself is coded in java, it will be the only one able to talk with our database. Its role is to do some security checks and to support parallel usage of data. Having seen that all federations and all pairing programs have different ways to send and receive files, the server will provide particular import/export scripts for each actor. We want this to be as modular as

possible. The database, as for it, is implemented in SQL. The actual database schema can be seen in the implementation part.

Finally, we have to think about the web hosting. We will host the web-app and server in Switzerland it will cost 10.- per year. We will also have to buy a domain name. We want to have a *.com* site web because of our international objectives. It would cost us too much to have one domain per country. It would also be problematic for international tournaments with players from multiple countries. *mychesshub.com* is available for 10.- and will probably be the one we will buy.

Development

As said before, we want our codebase to be as modular as possible. The objective is to add functionalities as time goes by, so the more our codebase is “future-proof” the easiest it will be to implement new features. We also want modularity because of the import/export scripts. As told before, each actor with whom we will work has its proper way to send or receive data. Since we don’t want others work for the integration of our tool, we will have to make different script for each entity. If we want it to be easily maintainable, we also must make it the most modular. Ruby on rails and Python Flask being quite recent frameworks, we will have to be constantly up to date concerning plugins we will use and will also always look at possible better alternatives.

A solid agile focus strategy needs to be a core part of the project. In the future, it is possible that the FIDE may settle down from their internal, political troubles and reach out for a partnership if the software becomes popular enough. With this idea in mind, having a flexible codebase and implementing with the potentiality of one day having either complete synchronization with the FIDE’s database, and maybe even officially handling tournaments and player licenses and registrations is a requirement to not suffer massive deficit from development, loss of trust and to overcome expectations from federation officials.

We will have to work on our marketing skills. We already have some notions but if we want to sell our product in multiple countries, we will have to learn how to do it. The strategy can be totally different from a country to another so there will be a lot of work to determine a good publicity campaign for each country. We will also have to create a support service. At first it will only be an email communication. If we see that the support is used a lot, we will use a ticket manager to be more efficient.

We have to improve our contact with the different chess federations. We first want to have some federation as client. So, we can propose an automatic ranking handler synchronized with tournaments results. This would also allow us to reduce our hosting costs because all those federation data could be stored in their server. We also have to take contact with other pairing program owners to find an agreement with them. This will be the step after the first test with VEGA. If the test works well, it would be easier to convince other suppliers to integrate our tool in their program.

As was described in the previous chapter, we will have to convince tournaments organizers to pay to use our tool. This will be the most important part for durability, seeing that server and codebase maintenance incurs periodical and often non-negligible costs. The difficulty is also increased because we will have to work with people from different countries, with different cultures and also different purchasing power.

Implementation

The choice of implementation was a recurring question during the study. Although our initial approach was a client-side, native application, the paradigm switched when the second idea took shape. The major cause of this change was that the new software didn't have the objective to facilitate running a tournament at a specific point in time, but rather facilitating the organization of the tournament beforehand. As such, instantaneous and flexible availability was not a crucial point anymore. With this constraint removed, a much more optimal implementation approach would be a web service and API, which we chose to implement using one of the major technologies at time of writing, [Flask](#).

Flask is a microframework for developing web applications in Python. Using this paradigm not only drastically cuts the development time for the client-side of the software, but also highly increased compatibility: any HTML5-compatible browser should be able to access the service and profit from complete functionality. This is an important point, anchoring the application in the development of portable technologies: users can choose to manage their tournaments or subscriptions on-the-go as well as from a desk. The cost of such a choice are the limitations in performance and capabilities of HTML and browsers, but the software is extremely light-weight and those limitations are compensated by the benefits.

Other software used in the prototype and which should be kept, or a variant of, for the final implementation will be listed below:

- [SQLAlchemy](#): a database toolkit and object relational mapper (ORM) which simplifies by many factors the development and maintenance of a database system in a web application. It allows for simple creation of schemes, relationships between objects, handling of language-specific constraints, and a strong built-in security and optimization of queries.
- [FlaskLogin](#): a user-session management toolkit. Helps building more secure and reliable session management systems.
- [WTForms](#): a HTML forms management toolkit, for better handling of forms and more security from malicious users.
- [Stripe Checkout](#): the current leading software for simple implementation of checkouts and credit card payments in Flask. This comes with the downfall of an expensive pay-per-use system, but a very simple implementation. Depending on the development budget, [WePay](#) would be a much safer alternative, providing insurance coverage in case of legal issues in case of use of the system for fraudulent payments between users abusing the system. In all cases, a strong security system should be put into place to prevent any fraud. User verification should be done by hand for the case of receiving users (tournament organizers), and checks should be implemented to make sure paying users have consistent and natural behavior. The system should also not allow

for large payments to be completed, bounding the maximal amount for registration to a tournament to a value around 30 to 50 euros.

- [Celery](#): a framework to simplify scheduling of tasks. Currently the most used framework in Flask for this purpose, a strong all-around tool which clearly implements the desired functionalities for development.

With such a framework in place, the rest of the implementation is schoolbook development of the different features. Flask allows for simple creation of a RESTful API in case the pairing engines like Vega wish to have direct access to the application, preventing the user from having to manually export tournaments from the browser to his local software. Although this remains an option in case pairing software developers prefer only implementing a file reader rather than a web API bridge.

Concerning application hosting, there are barely any benefits from not using a distributed platform apart potential access to the physical servers. The current leaders in the market are Amazon Web Services (AWS) with, for our case, Elastic Beanstalk (35% market share), Microsoft Azure and Google Web Services. Another alternative, of a smaller size, and which is especially friendly for newcomers and easy to take in hand, is Heroku. All those platforms offer relatively similar pricing and far more services than would be needed for the *ChessHub* application. An interesting plan could be starting with Heroku, which comes with free trial hosting with small bandwidth and general aptitudes, but which should already be far enough for the beta and early phases of release. Afterwards, AWS currently has the best price offers for startups and individuals and would probably be a good option to switch for once demands for bandwidth and general capability would start increasing. A rough estimate would be 1 to 2 years into release, or around 1000 users.

The following figure details the application structure and the different services provided. A further explanation of the services also follows:

- Organizer details management: allows users to provide payment details and id verification (to prevent fraud). Only with those elements inserted and verified can a user create a paid-registration tournament. As discussed above, this is to prevent fraud. Also allows organizers to review and edit their license model or status.
- Player profiles management: allows users to review and update their different license numbers for each federation they belong to.
- Tournament registration management: allows players to view the status and info of each of tournament they have registered in, practice payments if required, and other management tasks like un-registering.
- Tournament registration: allows user to register or reserve a spot while delaying payment to a tournament.
- Tournament creation: allows organizers to create tournaments
- Tournament management: allows organizers to edit their own tournaments, review the current registration statuses, and export to pairing software-readable formats.

- External database importation: synchronization with the different federation's databases for player statuses and rankings.
- Latent database updates: current time/date dependent database updates, like tournament completion or registration availability.
- User & sysadmin alerts: general email alerts in case of either tournament dates approaching, tournament information updates... for users and server failures or fraud detection for system admins.
- Pairing software API: generic API which provides automatic export of application-tournament format to local pairing software format through csv access. Requires special software credentials (ideally implemented with OAuth).

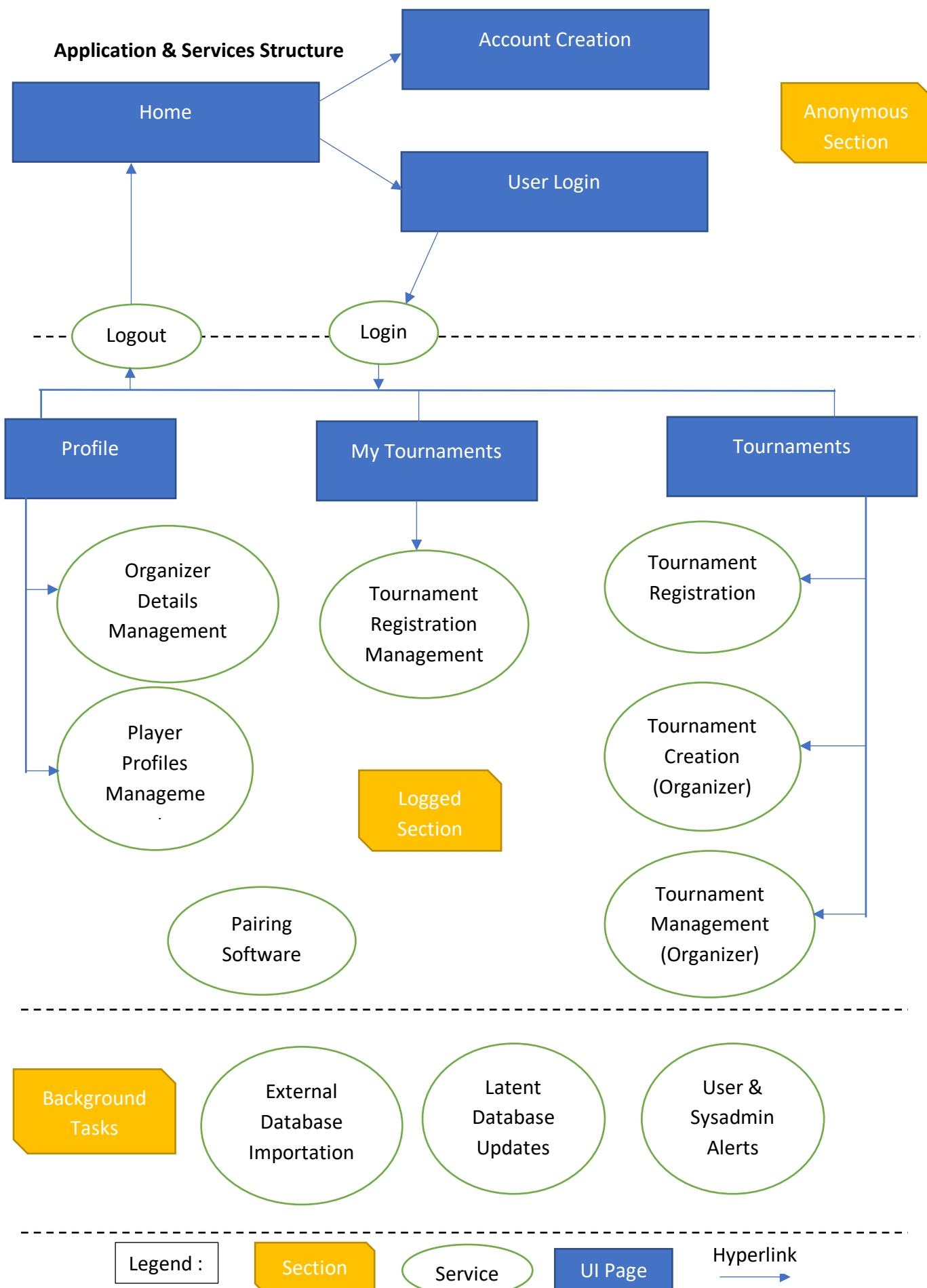


Figure 13: Application and Services Structure

Database

The following figure shows a scheme of the database.

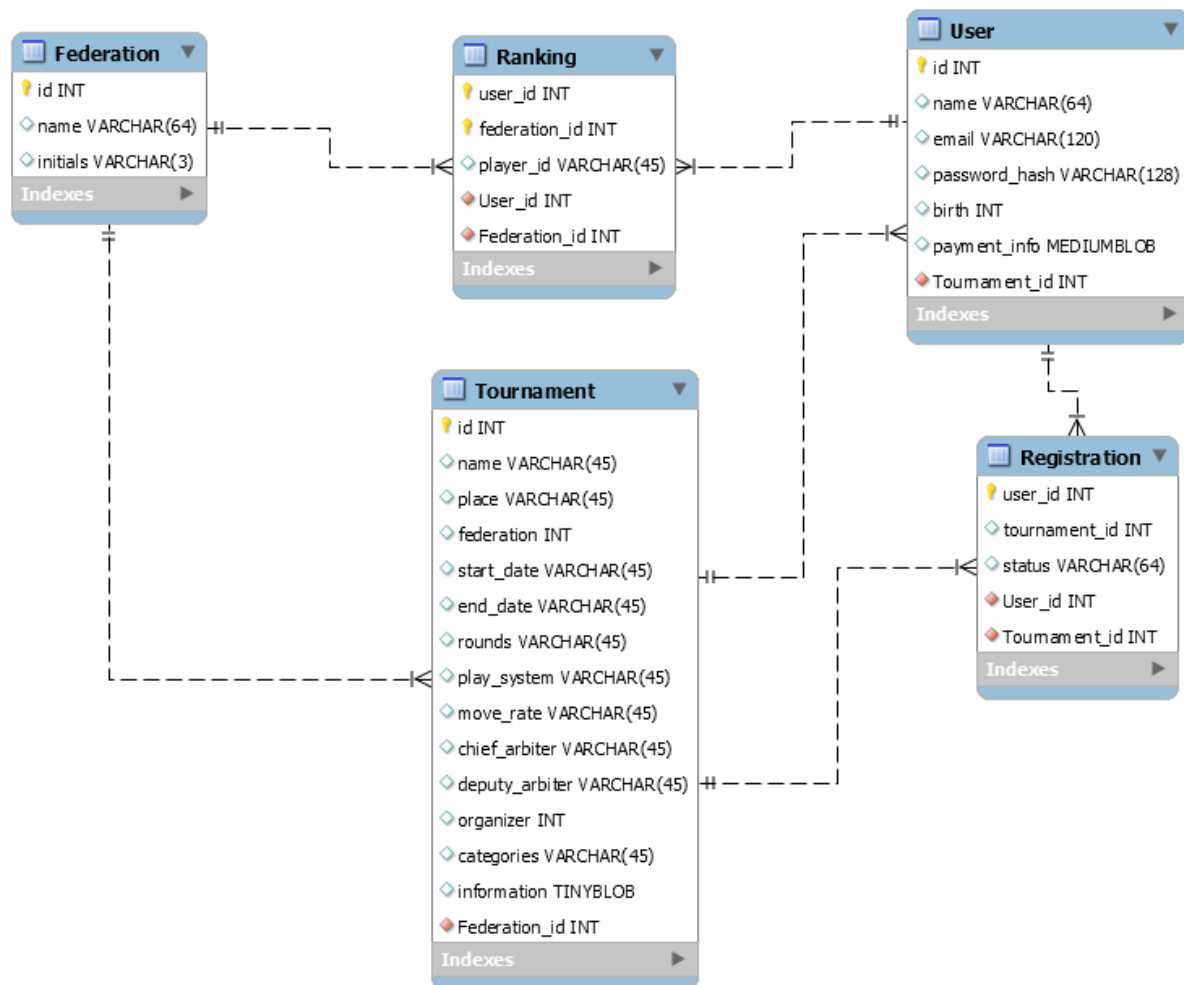


Figure 14: Database Schema

Note: both N-to-N relations (Registration & Ranking) have been made as explicit tables for clarity. The 1-to-N relation between User & Tournament is for the organizer field.

An important challenge to the project will be populating the database and synchronizing it with external databases from the different federations. The most important, the FIDE, updates its database once per day with the new player rankings. The rankings files can be accessed via a rudimentary API, a simple HTTP link, then unzipped and processed. Not all federations have explicit APIs, but the French Federation (FFE) has a complete database of all players of all federations, complete with all their different rankings. Although no contact could be established with their system admin, their website can be crawled through HTTP requests and all data can be extracted with the careful use of a HTML parser like Jsoup. This was not implemented in the prototype, because a partnership or a link to an API would greatly simplify the synchronization process, drastically cut development time, and provide future-proofing and robustness. An ideal solution would be hooking webhooks to a web service provided by the FFE (or the owner of the database, if it doesn't belong to the FFE), but would require a

strong collaboration and discussion. This would still probably result in less development time but might incorporate a payment for the development of the service on their side.

User Interface

The following screenshots demonstrate a rough potential user interface. It follows general guidelines of user-friendliness and has the aim of being usable by non-initiates of web applications and interfaces, staying in line with the community of chess players.

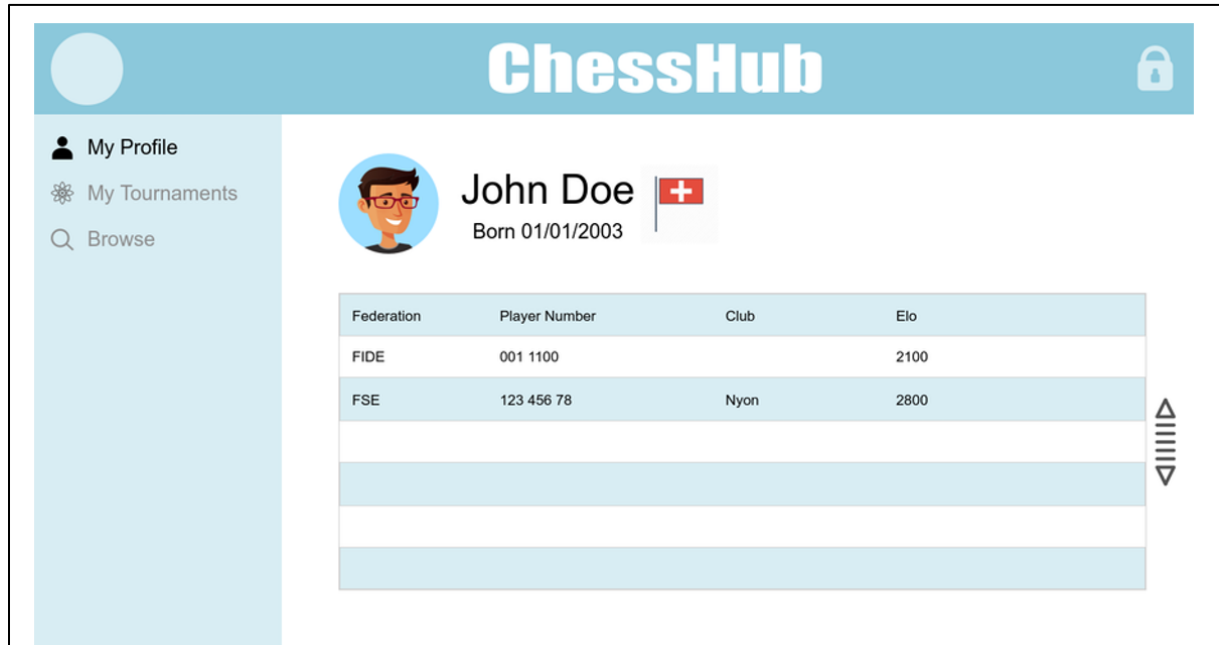


Figure 15: Prototype: Profile Page



Figure 16: Prototype: "My Tournaments" Page



Figure 17: Prototype: "All Tournament" Page



Figure 18: Prototype : "Create Tournament" Page

Conclusion

Achieved Results

Three alternative solutions can be extracted from the work done for this project. In decreasing order of market opportunities:

- Initial idea: developing a brand-new pairing software, with good user interface and easy to get into and develop proficiency with. Idea dropped because of lack of opportunities and too strong competition, with barely any gaps being filled.
- Close partnership with the FIDE to re-design user accounts and database: a strong idea, setting a foot into the monopoly of the FIDE to provide new features and completely overhaul an existing but outdated system. Solution pending, mainly due to lack of responsiveness of FIDE officials, but to be kept in mind for agile focus strategy when developing final idea.
- External application with strong partnership with pairing programs: the selected solution, profusely detailed in this report, taking advantage of a completely unexploited market opportunity. Core strength being new technologies allowing linking of vast pre-existing systems and direct penetration into the market thanks to the partners and their userbases.

This third solution is ready to be implemented and be somewhat easily made compatible with the second idea for future-proofing. A prototype has been made, to be seen in the appendices.

Future Development

Development of the software, based on knowledge from implementation of the prototype and prior experience with software engineering, is a medium-difficulty task which should cost around 100 to 150 development hours. No technical difficulties should be encountered by the developer, seeing how basic the concepts are and how profusely most frameworks described in the 'Implementation' section fill technical requirements. Three ideas are: investing man-hours ourselves into the development, obviously the cheaper alternative in terms of direct finances, delegating development for a semester project in software engineering thanks to a partnership with an EPFL laboratory, or hiring an external junior developer at an attractive rate. These last two solutions would probably offer a good quality/price factor but would also require efforts of well-defining the requirements and a close follow-up of the work done by the developer to insure agile implementation. Developing ourselves still seems to be the best solution all-around.

Insights

This project was extremely interesting as it was a direct application of the Enterprise and Service-Oriented Architecture course. We encountered in real situation many examples and concepts from the course and were able to tackle them with enough prior knowledge to get the most profit out of each, thanks to solutions studied during the course. We learned from practice how contextual inquiries and SEAM models are strong tools to put into perspective systems, and how to review our ideas. We had to completely change the project direction in the middle of the semester, because our initial idea was completely off from what was needed by the system and could potentially hold a place in the market.

Obviously, we learned a lot about the chess world, how federations work, how small and big of a system it is at the same time: we were able to contact the developers of applications with thousands of users and to get feedback from organizers from a large portion of Switzerland, while at the same time learning of geo-political implications of federation officials. This was a very eye-opening scenario and a good introduction to what could be the range of a system which seems small at first: we realized that when getting into the project, we could only see the very tip of the iceberg.

Finally, we had to develop from start a business idea, which is something new and not often taught during theoretical courses. We had to explore new branches of econometry, marketing, business and market analysis... those are important tools that will benefit us in the upcoming years and for future projects. We also had to develop our communication and negotiation skills, getting in touch with people with very different experiences and points of view, and having to understand the goals and needs of each under various conditions.

Research Question

One of the major difficulties that we went through during this project was modeling temporal evolution of systems. We believe that systems go through not only large phases during their lifetime, but also might follow periodic (maybe even cyclic) changes in behavior. Although we studied a single system, we often wanted to show different states of it (previous states, different alternative solutions, evolution of a system when going through the different stages of the adoption process...) and often had to go through a tedious process of redefining actors and concepts, as well as geometrical sanity of the models. We believe that, in a few words, an interesting topic for future research into systemic modelling could be temporal evolution of systems.

References

- Chess Federations:
 - FIDE: www.fide.com
 - FSE: www.swisschess.ch
 - FSI: www.italychess.com
- Chess Pairing Programs:
 - VEGA: www.vegachess.com
 - SWISS MANAGER: www.swiss-manager.at
 - bbpPairings: <https://github.com/BieremaBoyzProgramming/bbpPairings>
 - JaVaFo: <http://www.rweb.org/javafo/JaVaFo.htm>
- Regulations:
 - FIDE: <http://pairings.fide.com/>
 - FSE: we were orally asked to refer to the FIDE regulation
- Business:
 - Where to Play, Marc Gruber and Sharon Tal (2017)
 - Crossing the Chasm, Geoffrey A. Moore (1991, revised 1999 and 2014)
- Core information on how to target the right market share and how to position the product in the system. Concepts extracted: “Market Opportunity Set, Attractiveness Map, Agile Focus Strategy”
 - [LAMS Research Papers](#)

Appendices

All of the implemented codebase is available on our [Github Repository](#)

Appendices are documents that were shown during meetings with M. Regev during the semester and other miscellaneous docs.