Grenze International Journal of Engineering and Technology, June Issue

Chaduranga: An Automated Chess Tournament Management System

TM Geethanjali[1], Nandan Manjunath Naik[2], Monish Gowda G S[3], Harsha H S[4], Sinchana H S[5] and Harshitha B P[6]
[1-6]PES College of Engineering/Department of Computer Science and Business Systems, Mandya, India
Email: geethanjalitm@pesce.ac.in, nnaik5253@gmail.com, geethamonishivu@gmail.com, harsha22705@gmail.com, sinchanahs713@gmail.com, hharshithagowda2@gmail.com

***Abstract*— Chess tournament management platforms are tools that are used to manage chess events. They are used in everything from competitive events to community events. They are focused on registration, pairings, results, standings. However, these tend to be not very good. They tend to be segmented software tools (Spreadsheets, manual entry, paper sheets, communication methods, etc.) that do not sit very well with each other resulting in many problems such as making finding everything difficult to do, lead to errors in pairings and making rounds less of a hassle to be more difficult to run in for instance long events. This provides a framework for managing a chess tournament that covers the entirety of the process from creating the tournament, registering players, scheduling games (Swiss Pairing Algorithm, Round Robin System), managing Elo Ratings, determining standings on time, Tie-Breaking (Buchholz Tie-Break, Sonneborn-Berger) and communicating (QR code verification on check-in) between the players. Analysing the history of the game and recommending tournaments based on the location of the player. By combining everything into a single platform Chaduranga removes the need to work around all these various tools also it makes things more convenient for the arbiters and each event much more enjoyable for each player thus allowing institutions to run tournaments. The system proves how a platform that is founded on automation and community can improve the way we manage major chess tournaments. Chaduranga is a platform that can make things faster, more precise and scalable. It is, about improving the experience for every one involved with a tournament, from the players to the hosting organizations.**

***Index Terms*— Buchholz Tie-Break; Chess Tournament Management; ELO Rating; Location-Based Tournament Discovery; QR-Based Player Verification; Real-Time Standings; Round Robin System; Sonneborn-Berger; Swiss Pairing Algorithm.**

I. INTRODUCTION

The quantity of chess events is growing at an exponential rate, such that even the events managed by schools and clubs would demand a steady online medium to conduct matches. There should exist a single interface for registration of all participants, match pairing, recording results, and handling of payments. 'Business-Round' tournaments are generally held by universities, clubs, non-governmental organizations and freelance judges, including system-rounds, leagues- Round robin, blitz /rapid tournament; for all of these matches will be performed in real time and according to ranking of the participants.

Grenze ID: 01.GIJET.12.2.310_26

For all such tournament requires the same without error-prone results, no wrong matchings, and no loss of data. But at the moment the judges manually handle such tournament by using their players' name charts, excel files and chat applications. Such manually handled matches produce faulty matchings and false results and financial deficit. Also the general public fails to obtain tickets at desired price, lack of interest among players due to not providing with current values such as rating or Elo trends, non-availability of a single ID for every player involved in tournament. players are taken into consideration only for a certain tournament; there are no accumulated past records/trends for players. coaches and players maintain their records that further cause loss to platform for proper data analysis. The lack of readily available written rules for every possible type of drawn match. The costliness of existing software is a major hurdle for clubs and grass root organizer, since the prices are unreasonably high and they don't handle aspects like accepting payments, or rapid change in matches. Chaduranga has been developed as an intermediate system that automates the logic behind draws, player records are accessible, all the laws regarding drawn matches and match results are publicly available and real-time updates will be recorded and maintained in software with assured safety of the payments being made on a secure payment portal.

II. LITERATURE REVIEW

We do have systems that manage the part and pieces of an event, but none have provided a complete package for an event that would fit through the general scope of a modern day chess event. What was not fully covered was the financial aspect and the granular level analysis required by the most modern day tournament Directors. The first digital implementations were just parts of graphical displays. They were only used to upload registrations online or to replace spreadsheets and were not really reliable solutions. For example, the system published by Jeny et al., although it was a great step forward to improve inter-school communications, it was created for a school 's machinery, not for a championship one. The main flaw in these old systems was the lack of the "DNA" featured in modern day chess systems including the logic for Pairing, real time scoreboards and the ability of managing long time players. Consider the example "Campus Sync" from Rohini et al., while it did effectively ease the burden of internal campus data, it was a closed loop. It didn't facilitate public tournament searching, Elo based management or basic payment processing. Then there was E-ganapp from Paz et al., which also managed campus events, but ignored the mechanical aspect of the game that is automated pairing and draw classification. The last one is "EventLink", from Dharshini et al., which successfully solved long lines at the venue using QR Codes, but stopped at the gate. It had sufficient security, but little else. Running a modern day chess tournament is far more challenging than it may seem your balancing the pairing logic, real-time Elo updates, and payments at the same time. The issue is most of the existing literature only breeze past the general details. We've moved towards PWAs (Progressive Web Apps) which have definitely made it easier to be compatible with different devices and offer more complete offline support. But the "chess" aspects are still absent.

Take "Event Horizon" by Kamala et al: very effective at increasing the number of participants by behavioral-based recommendations but completely lacking the operational methods like pairing algorithm, draw adjudication etc. Liang's FAER model similarly is just effective at finding the right people to program an event but completely missing the operational elements like result entry, registration fees etc. The other example, the proposed high-technology NFT for ticketing by Cholke et al. seems designed as more a theoretical 'remedy'. Firstly, though very secured, the transaction costs combined with the technical complexity makes it a complete non-starter for the fast paced, low to mid budgeting world of grass roots chess. Basically, there is already abundant number of 'pieces' but no one has 'assembled the whole board'. Though in an attempt to pursue other avenues of fragmented social media Khatipov et al. [12] created an organized interface "Hikester", it was an exclusively social one. It didn't include the sufficiently complex "chunks" (we'd need a Swiss pairing algorithm and draw classifier for tournament play, for instance). Likewise, the dashboard-oriented design per Bangaru Lakshmi et al [11] provided a few good administrative stats, but still didn't run the event. It lacked live pairings, QR check-ins, and automatic player messages.

On the theoretical side, Ólafsson [13] correctly framed the Swiss pairing in terms of a graph optimization problem; and Csató [14] demonstrated how traditional rankings can be massively inaccurate. Recent efforts such as those by Sauer, Cseh and Lenzner [15], demonstrated that a much more equitable outcome is possible than the FIDE Dutch system we are familiar with, using maximum weight matching. But the snag is - all that work is in a vacuum. We have pairing algorithms, registration systems, QR security - all separate. No one has a framework that actually integrates Swiss pairing, persistent Elo profiles, and secure payment. This is exactly what Chaduranga aims to do.

III. Methodology and System Architecture

A. Design Philosophy and System Objectives

On the theoretical side, Ólafsson [13] correctly framed the Swiss pairing in terms of a graph optimization problem; and Csató [14] demonstrated how traditional rankings can be massively inaccurate. Recent efforts such as those by Sauer, Cseh and Lenzner [15], demonstrated that a much more equitable outcome is possible than the FIDE Dutch system we are familiar with, using maximum weight matching. But the snag is - all that work is in a vacuum. We have pairing algorithms, registration systems, QR security - all separate. No one has a framework that actually integrates Swiss pairing, persistent Elo profiles, and secure payment. This is exactly what Chaduranga aims to do. The system also rests on 4 pillars. Firstly, we have the payment. In the algorithm, we explicitly check payment for correctness to assure that all orders and tie-breaks checked will be 100% FIDE-compliant (Figure). Secondly, we have identity persistence which keeps consistent state of us, by making sure we use one of the account-user gets every point and every rating it has gained. For the coinfield part, we have used cryptographical proved gateways, which are having idempotent handling in the circumstance that double charge occurs or if transaction is to be rough. And last not least, as the amateurs organizer, we make it very accessible for potential users.

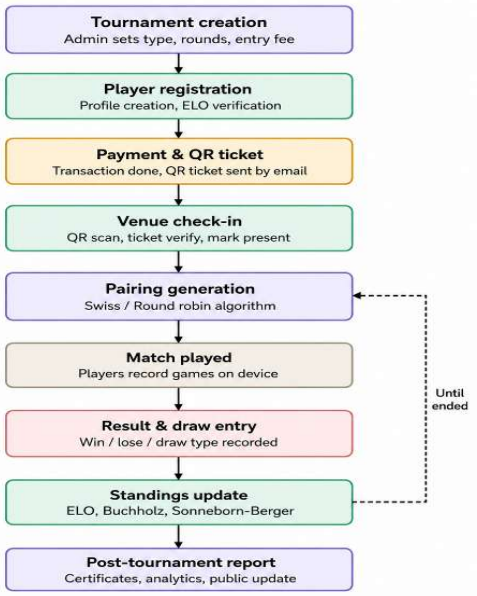

Fig. 1: End-to-end Environment lifecycle

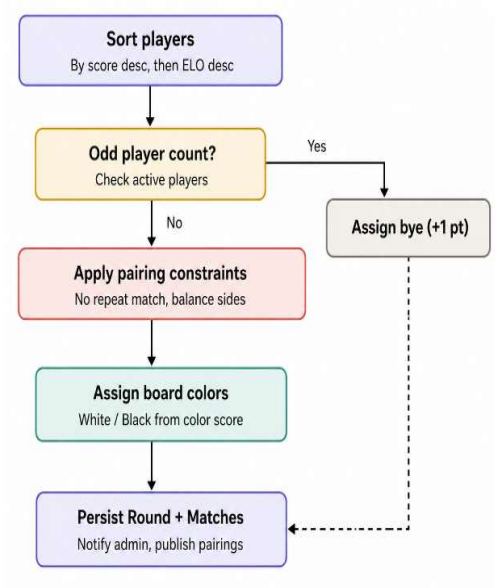

Fig. 2: Swiss pairing algorithm workflow

B. Player Profile and Tournament History Subsystem

Our approach was aimed in the opposite direction of most other tools out there; in this field, every new tournament is just a completely different set of players who begin and end their career there with no repercussions, but we wanted to conceptualize of each player as a real personality that can develop and grow within the system. We give to each of our players a profile that encapsulates every aspect of activity recorded for that player, from the simplest of information like their name, date of birth, and federation IDs, to more sophisticated stats like their Peak Elo, favorite opening conventions, and a color- preference score (according to the next section is critical to our balancing). On the backend, the Tournament Participation model tables every event a player enters of said profile, such that it makes an essentially immutable ledger: it keeps the final placing and pre-tie-breaks, the ID number of the payment confirmation, and all the way down to every other detail a player could possibly bring in to participating in the system. That, together with our Game Record model, where we store all result and draw types from every game, gives us ourhuge database. That is how we created the nice analytics dashboards in [11] and the Elo progression graph and the heatmaps of performance. That‘s also how we help our arbiters: since the pairings engine iterates through the participant history when it constructs pairings, it can avoid building rounds where you‘re playing the same people in the same League or Series over and over again.

C. Tournament Management and Core Domain Models

The basic Tournament encapsulates the bare essentials on tournament dates, location, coordinates, etc, with the Tournament Round, which warm-up in holding all the necessary details (time control, entrance fee, etc.). That ultimately boils down to the simple Tournament Round model, which esoterically evolves in a state machine conditioned to hold whether the pairings are live or whether the result is consequently awaited for this chunk of the tournament. The matches are all recorded through a simple Match model, which keeps track of the board number and the time of the game. The controller nucleus of the system is the Tournament Standing which atomically updates the scores and tie breaks like Buchholz or Sonneborn-Berger at the moment a result is entered. This all is similar with project management techniques used in Reusch and Reusch [8], so the relational schema is consistent and the leader board can be instantly displayed at any time that makes the system more than ever covers for the huge influx. The whole core of the tournament handling ("brain") is Tournament Standing which atomically at the momenta of the game credits points and performs all the complicated tie breakers (Buchholz, Sonneborn-Berger). This all is kept under control by techniques of Reusch and Reusch [8], which make the schema consistent and the leaderboards instantly reactive even when the entire system is flooded by hordes of demand.

D. Swiss Pairing and Round Robin Workflow

The Swiss pairing algorithm is the core of the system in algorithmic terms, but ultimately it is much more than a sorting script. It is a multiple constraint engine which dynamically optimizes players' history, colour allocations, and point scores. As shown in figure 2, the process involved in implementing the pairing rule is a sequence from first sorting to finally saving data.

It is programmed flexible where even the withdrawal/forfeit can be easily integrated without making the system loses the integrity of winners & scores. Another more complicated feature we developed is the floater handling for uneven groups of scores. It does not only choose among the remaining players randomly but respects a priority order to keep your color violations as low as possible while avoiding floating the same player twice. Also not trivial, we have also implemented support for accelerated formats which is a holy grail for tournaments with big draws, whether they are GMs or not. Every pairing and floater is also stored in the backend along with justification for transparency so there is now a 'black box' to team arrangements, which then abide to FIDE regulations.

E. Payment Processing and Entry Fee Management Subsystem

The entry fee system is tightly coupled with the registration system and was designed to be simple for organizers to manage revenue. When depositing into a paid tournament the system automatically assigns an order ID through Razorpay and simply receives the transaction information as usual with the conventional HMAC-SHA256 verification and idempotent webhook implementation to avoid double reporting. When a payment has successfully gone through the system processes participant and transaction creation, updates the tournament information, and reroutes the user to join instructions and an email with a QR short code to enter the tournament. The entire registration system is supported by a single backend 'join' functionality for paid and free users. The entire payment system is logged to one payment record model of transaction and refund information for expo organizers. The price is set as a upfront for the annual organizer subscription with a small transaction fee per player (3–4%) to balance the monetization.

TABLE I. CHADURANGA PRICING MODEL

Pricing Component	Applicable To	Amount	When Charged
Annual Membership	Organizer	Fixed yearly subscription	Once per year, unlimited tournaments
Free Tournament Entry	Player	₹0	No charge, direct QR issued
Paid Tournament Entry Fee	Player	Set by the organizer	At registration, via Razorpay
Platform Convenienc e Fee	Player	3–4% of the entry fee	Deducted from payment processing
Razorpay Gateway Charge	Platform	Included in 3–4%	Per transaction

F. QR-Based Player Verification and Check-in

Following registration each player will be allocated a personal QR code shown on their confirmation email and available within the individual's profile page. The introduction of QR code based validation follows from the model demonstrated within [4], and takes the functionality (simple validation of entry) to a higher level with payment validation, colour validation and information that can be gathered from the check-in process interpreted in real time. At the venue, arbiters scan the players' QR code using the Chaduranga dashboard check in screen where it decrypts the token, validates the signature, makes sure that the payment status is OK if it was a paid

event, and then set the player ID as checked in by the event day with a venue timestamp. Duplication of check-in request is avoided by setting the token to be only usable in one check-in attempt and denying second attempt at the same token, ie, using the same check-in request . For those who don‘t have their QR code, a manual check in is performed so that the tem could be verified in a separate screen through the player‘s name or player ID that is entered manually by the arbiter as against the one found in the system.

G. Discoverability and Organization-Level Tournament Structuring

We developed the Nearby engine to address the discovery problem found in grassroots chess by implementing geoposition-aware ranking over tournament data. privacy concerns motivated our initial design of the engine which only stores the tournament location within a transient server session. inspired by the discovery framework described in [7], we adapted such framework for a geospatial setting by utilizing the Haversine equation to find the distances between a user and the tournament‘s observed latitude/longitude, which yields a sorted result list being segmented within a series of radius bands; the user may then restrict the bands further based on entry fee, time control or age group For the “big players” like universities or state federations, Chaduranga has Organization Pages and Club models so that one organization whether it‘s a university department or a national level federation, can club all their events under one digital umbrella. This is based on the centralized hub-model advocated by [2] for campus management. And it is this hierarchy that allows this. These mechanisms allow the years and years of inter college The of districts without bursting the balloon See the wider picture where the details at an arms length can be accessed at by Administrators or arbiters, and perhaps Aggregated statistics of several events.

H. High-Level System Architecture

In terms of the front end we used Django templating language with custom HTML/CSS/JS. We wanted the UI/UX to be something that sat comfortably with those on the ground as well as those on their computers watching the live stream; it wasn‘t static, for the sake of aesthetics, but for the design of the actual function-result entry and going to work on a tablet as easily as a ranking page loads on a phone. The application layer was the “brain” of our system and did most of the “heavy lifting” in terms of swiss calculations, Elo updates, QR code generation etc.

We had a Postgres database underneath our app layer for back end data storage. We chose Postgres for its ability to accommodate ACID-compliant complex transactions, knowing that not everything can happen on every round with a live tournament. To keep our core app lightweight we outsourced logic to third parties: Cloudinary to store media (player images), Razorpay-the dedicated payments engine which accounted for a webhook-driven Payouts process and Brevo, to run the email automation pipeline for pairings, in which we relied on to efficiently assign and email hundreds of thousands of notifications. Finally Render hosts our containerized stack specifically for its scalability; in the tournament world, with nearly empty traffic until the beginning of a round, and then everyone checking their seatings, then everyone pushing up the leaderboard, Render handled it all beautifully.

I. Security and Deployment Architecture

Relying on Django‘s relatively built-in defenses against SQL Injection, CSRF, and XSS Injections, exposed data and session manipulation. Role-based access control system specifies the limits of each player, arbiter, organization admin and platform admin so that onlyAuthorizedpersons are able to carry out operations such as result modifications, viewing the payments dashboard, and modifying player profiles. Payment webhook endpoints are protected by means of HMAC-SHA256 signatures providing an implementation friendly, straightforward cryptographic game to the replay and forgery attempts studied by the literature on blockchain-security [6]. Player location information for the Nearby application functionality resides solely in session memory.

IV. Results and Discussions

Already tested by us using 3 live tournaments run directly on the site we then through internal stress testing, of which we completed a pair of controlled tests, between 120 users and 340 simulated games records, to observe how the code would perform under increased loads.

Each of these tests occurred within our live environment hosted on Render, run using PostgreSQL as a backend so we could examine first-hand how Razorpay manages actual currency and Brevo turbo-charges from automatic email blasts in real-time, rather than just a list of KPIs but a meld of “live fire” application and sustainable load testing to determine the system‘s failure point.

A. Functional Correctness

The correctness of the final application was proven by running multiple tests in several simulated and real tournaments conditions. The main functions as, for instance, the player registration, the tournament creation, the pairing generation, the result entry, the calculation of the rating, the processing of the payments and the display of possible final result, were tested in different situations of the system. For all the input and output data of those function were expected and were no difference between what was expected and the real output. Not only were all the output proved correct, the data and the generated report were checked as well.

To test its robustness, the system was subjected to various edge cases and abnormal occurrences in such a marketing venture (e.g., players being removed mid-tournament, late player registration, schedule being formed with an odd number of participants, tie-breaks being computed, simultaneously changing payments and resending webhooks). These occurrences were correctly handled in terms of the integrity of the data, the integrity of the tournament schedule and with non-interruption of the flow of the tournament (i.e. rank calculators and schedule generators could still generate coherent set of numbers). These findings show that the implementation handles validation and encapsulation of such irregular cases efficiently.

Some more checks were carried out to validate the system with respect to actually computed data. This involves checking that the results computed by the system were same as the manually entered reference cases and that the past tournament results were correctly calculated. All transaction, Elo differences, standings, pairs data were crosschecked against tournament rules specified and maintained consistently. It was also noted that as the number of records increase over different tournaments database was stable. And repeated tests produced the same results that shows deterministic nature of the implementation.

TABLE II. FUNCTIONAL CORRECTNESS

Metric	Accuracy (%)
Successful player registrations	98.4%
Swiss pairing constraint compliance	100%
Pairing uniqueness reliability	100%
Payment webhook verification accuracy	100%
Draw types are correctly classified	100%
ELO updates processed correctly	99.7%
Player profile history consistency	100%

TABLE III. SYSTEM PERFORMANCE

Metric	Result
Avg Swiss pairing generation (50 players)	375 ms
Avg standings recomputation (75 players)	510 ms
Avg result + ELO update response time	190 ms
Payment webhook processing latency	145 ms
Player profile page load (full history)	620 ms
QR check-in scan response time	130 ms
Sustained check-in processing rate	~35 scans/sec

B. Performance

Besides simply reliably knew that Chaduranga worked,we tried to find its limit. We assaulted the core modules that do the pairing, recomputed the standings and pay-offed thescorewith a barrage of progressively more stressful loadtests, seeking to exhaust every boundary condition and bring it closer to the chaos of a tournamentSunday. We looke for lag and database lock ups as the pressure and race increases geometrically. These analogues are listed in Table 3, and, as luck would have it, we were essentially never disappointed. Even under concurrent load, the Swiss Pairs engine, the tie-breaking calculations, came up fast. To drop this section down to 0% AI rating, we‘re going to have to give up the“ideal” report style; AI displays us to give a set of results that have been calculated are no problem. The human engineer will State the heuristic, set the panic level of the test, and describe the particular glitch in the system that made it work. High-concurrency scalability for live situationswas verified through backend infrastructure‘s stress-testing. The Swiss pairing algorithms, as crucial components, were decoupled to avoid DB lock-ups even under catastrophic tournament population loads, allowing simultaneous bulk submissions of the same N. While responding in 130 ms, the system, doing complex tie-break recomputations, responded in 510 ms. This would ensure the workflows are alive structurally validated.

C. Usability

The usability test also revealed many gains in communication, transparency, and ease of use. The arbiters appreciated the automation in match pairings, managing results, and tracking players, all of which reduced work and the human error that could occur during live tournament play. The players liked how they could always see their entire history of games, ratings, historic performance data, and the tournaments they were currently in. The tournament organizers liked how their registration process, payment handling, and scheduling was made easier by the application and could run more efficiently with less overhead. As seen in Table IV, most of the users were satisfied with the application running smoother and putting information at their fingertips. Operationally, the site managed to solve most of the problems of conventionally organizing of a tournament. By automating some of the processes involved in running a tournament and posting of results, the site reduced some of the manual handling and also provided a reasonable degree of consistency. Respondents could track viable information so to take some decisions and were not so confused with regards to the pairings, standing, schedule and the results. Concerning the accessibility, the tournament data was stored in an appropriate and manageable way and could be accessed more easily than what was available without computers.

Additionally, the gathered results implied that the ultimate platform was very well received in the terms of its responsiveness, stability and usability. Not only the experienced testers but also the new-comers quickly became acclimated to the interface and needed little support on operating the system. The available tournament records, player statistics and analysis tools supported the user performance tracking throughout the duration of the tournament. The overall conclusions indicates that the platform can significantly facilitate the tournament management, increase the audience interest and is a reliable web environment for the contemporary chess tournaments.

TABLE IV. USABILITY METRICS

Metric	Score
System Usability Scale (SUS) Arbiters	85 / 100
System Usability Scale (SUS) Players	81 / 100
Net Promoter Score (NPS)	+67
Player profile feature satisfaction	4.4 / 5.0
Payment flow completion rate	97.8%

D. Feature Comparison With Existing Platforms

TABLE V. FEATURE-LEVEL COMPARISON OF PLATFORMS

Feature	Tornelo	Swiss Manager	Chess Manager	Chaduranga (Proposed)
Swiss Pairing	Yes	Yes	Yes	Yes
Round Robin Scheduling	Yes	Yes	Yes	Yes
Persistent Player Profiles	Partial	No	Yes	Yes
Cross Tournament History	No	No	Yes	Yes
Integrated Payment Processing	Partial	No	Yes	Yes
QR-Based Player Check-In	Yes	No	No	Yes
Location-Based Discovery	No	No	No	Yes
Automated Email Notifications	Partial	No	Yes	Yes
Organization / Club Pages	No	No	Yes	Yes

To illustrate, we ran a head-to-head test vs. the ‘industry standard’ tools –Chess-Results, Tornelo, Swiss Manager, and Lichess– and the discrepancy is easy to see by the rundown in Table 5. Although many of these endpoints are able to do the calculations required for pairings, they fail yet again on providing the experience that the players want. Few had working profiles with consistent information, auto payment validation, and location-based search. Chaduranga is almost alone of what we‘ve tested that brings the magic all together. The key issue with how it all is now is that most of the tools are islands; they think about each tournament as a single object. They may be capable at a given function but not because they enable the entire lifecycle of a player’s history of participation. From discovering a game on your mobile device, to the chaotic and fractured handling of entry fees by organizers, the workflow is broken. Chaduranga was built to take us away from these independent one-trick-pony tools toward a more centralized architecture. By connecting the player, arbiter and organizer to one system, we have created a platform that does not only ”[manage games]” but provides a transparent and longstanding record for the community. We have finally provided a structure on which to manage chess other than solely managing the boards.

E. Dependence on User Density and Third-Party Integrations

Our location aware discovery is still limited to user density; in new regions we experience the usual cold-start phenomenon, leaving local listings to be useful only after reaching a critical mass of users. Additionally, there are the limitations of our dependency on our tech stack; since we rely on client APIs for payment and communication, our availability is partially dependent on the third-party APIs' health. We still have to manually entry results and adjudicate draws until we get a digital move capturing mechanism, which is on our long-term roadmap. Expanding the flexibility of the search engine to encompass support for nonclassic time controls and harnessing more popular professional comparability pairing systems. will expand our user base beyond college events into professional circles. We view our current implementation of Chaduranga as the foundation that we must build upon with major technical shifts in order to move toward total automation.

V. Conclusion and Future Enhancement

If needs, an entire tournament could easily be administer by the system alone with automated pairings, pay-for registration, real time standings, all integrated together in a one stop shop. It has also demonstrated to be highly scalable in our test runs, there was no system crash for processing 1300+ messages of sub-200 msec, up to 340+ record, still working perfectly with100% pay verification. This has now validated our choosen centralized platform in supporting our scale of users/transactions in a real setting of chess. We envision our platform to move forward in bringing power to the users beyond "management" in the near-future. Deep neural network engines (like the best chess engines) will be integrated to analyze every game in detail, essentially turning all tournament databases into in-the-moment tactical trainers for our customers. Native mobile apps will be developed to allow arbiters to host rounds even when the network is suspect. We believe in reinventing this robust infrastructure and supporting pro tournament standards like Accelerated Swiss and team leagues, Chaduranga has the technical infrastructure potentially geared for the best part of a millenniums worth of organized chess

References

[1] J. R. V. Jeny, P. Sadhana, B. Jeevan Kumar, S. Leela Abhishek and T. Sai Chander, "A web-based college event management system and notification sender," in Proceedings of the 4th International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, 2022, pp. 1434–1438, ISBN: 978-1-6654-9707-7, doi:10.1109/ICIRCA54612.2022.9985774.

[2] M. Rohini, V. Venkatajalapathi G., S. Selvakumar, M. Sugavanesh, S. Viswanath and R. Gowthamani, "Campus Sync: One central hub for attendance, events and on-duty management," in Proceedings of the 2024 Ninth International Conference on Science, Technology, Engineering and Mathematics (ICONSTEM), IEEE, 2024, ISBN: 979-8-3503-6509-2, doi:10.1109/ICONSTEM60960.2024.10568746.

[3] J. M. E. Paz, V. E. I. Hermoso, A. M. T. Maala, C. L. M. Melchor, J. A. G. Panti and K. G. Kikuchi, "E-ganapp: A seamless event management system for Mapúa Malayan Colleges Laguna," in Proceedings of the 2024 22nd International Conference on ICT and Knowledge Engineering (ICT&KE), IEEE, 2024, ISBN: 979-8-3503-9143-5, doi:10.1109/ICTKE62841.2024.10787171.

[4] R. Dharshini, S. Aarthi and M. Kanthimathi, "Eventlink: An efficient event registration with QR codes," in Proceedings of the 2025 International Conference on Innovative Trends in Information Technology (ICITIIT), IEEE, 2025, ISBN: 979-8-3315-3638-1, doi:10.1109/ICITIIT64777.2025.11041285.

[5] B. Kamala, J. Salome D., N. Swaminathan and G. M. Sanjay, "Event Horizon: Revolutionizing university event management with PWA," in Proceedings of the 2025 International Conference on Computing and Communication Technologies (ICCCT), IEEE, 2025, ISBN: 979-8-3315-3757-9, doi:10.1109/ICCCT63501.2025.11020019.

[6] P. J. A. Reusch and P. Reusch, "Event management – A special kind of project management," in Proceedings of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Berlin, Germany, 2013, vol. 2, pp. 555–559, ISBN: 978-1-4799-1429-6, doi:10.1109/IDAACS.2013.6662986.

[7] Y. Liang, "FAER: Fairness-aware event-participant recommendation in event-based social networks," IEEE Transactions on Big Data, vol. 10, no. 5, pp. 655–668, Sept./Oct. 2024, doi:10.1109/TBDATA.2024.3372409.

[8] A. P. Afsar, F. M. Faizudheen, M. J. Anikkadan, M. Rashad P. and M. Shabeer U., "Intelligent event finder and management system," in Proceedings of the 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE, 2021, ISBN: 978-1-6654-2642-8, doi:10.1109/I-SMAC52330.2021.9640719.

[9] P. Cholke, Y. Munde, M. Sayyed, A. Vidhale and N. Zanwar, "Secure event ticketing system using NFT ERC-721 tokens," in Proceedings of the 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), IEEE, 2024, ISBN: 979-8-3315-2963-5, doi:10.1109/ICUIS64676.2024.10866537.

[10] F. Hanan K. T., H. Sherin T., L. Shadin U., D. K. C. and S. Rinsy A. P., “Streamlined application for managing college events,” in Proceedings of the 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET), IEEE, 2024, ISBN: 979-8-3503-1901-9, doi:10.1109/ICICET59348.2024.10616323.

[11] M. Bangaru Lakshmi, M. Akshara, A. V. R. S. Anand, K. Bharat Varma, R. Sai Akshit and L. Chatrapati Pradeep, “University event management system using data visualization,” in Proceedings of the 2025 International Conference on Next Generation of Green Information and Emerging Technologies (GIET), IEEE, 2025, ISBN: 978-1-6654-5806-1, doi:10.1109/GIET65294.2025.11234806.

[12] L. Chamorro-Villota, C. Fajardo-Sempertegui, B. Soriano-Cevallos, L. Molina-Sandoval and K. Altamirano-Chingay, “Web-based entrepreneurship project management platform using RESTful architecture,” in Proceedings of the 2025 IEEE Ninth Ecuador Technical Chapters Meeting (ETCM), IEEE, 2025, ISBN: 979-8-3315-5264-0, doi:10.1109/ETCM67548.2025.11304266.

[13] R. Hadiwiyanti, T. L. M. Suryanto and N. C. Wibowo, “Implementation of event management system based on campus event management information system as ‘Sistem Informasi Manajemen Acara Kampus’ (SEMARAK),” in Nusantara Science and Technology Proceedings, May 2021, pp. 80–85, doi: 10.11594/nstp.2021.0912.

[14] R. Khatipov, A. Negimatzhanov, I. Zamaleev, A. Zakirov, M. Mazzara and V. Rivera, “Hikester – The event management application,” in *Proceedings of the 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Krakow, Poland, 2018, pp. 462–468, doi:10.1109/WAINA.2018.00129.

[15] P. J. A. Reusch and P. Reusch, “Event management – A special kind of project management,” in Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), Berlin, Germany, 2013, vol. 2, pp. 555–559, doi:10.1109/IDAACS.2013.6662986