Introduction

Residents in Singapore often encounter difficulties when trying to use public sporting facilities such as fields, courts, pools and gyms (ActiveSG, 2025), due to fragmented and insufficient information dissemination. This issue was notably raised in Parliament by Nominated Member of Parliament (NMP) Mr Ong Hua Han, who highlighted the frustration of residents arriving at facilities that are unexpectedly closed 3 weather conditions such as lightning, with closures not communicated in a timely manner (MCCY, 2023). Compounding these issues, high demand for facilities makes securing bookings via the national portal, ActiveSG, challenging (ChannelNewsAsia, 2024). Even when bookings are made, many go underutilised because users are unable to find activity partners (Facebook, 2025), resulting in wasted capacity and reduced user satisfaction.

The Singapore government has actively promoted sports and healthy lifestyles among residents for decades (PMO, 2024), exemplified by initiatives such as free ActiveSG credits. These efforts have seen tangible success, with regular participation in sports rising significantly from 54% in 2015 to 70% in 2022 (SportSingapore, 2024). However, this growth has introduced new operational challenges.

Lightning Cat is a comprehensive cloud-based SaaS platform designed to address these issues. Upon accessing the platform, users are presented with a dynamic map showcasing all sporting facilities across Singapore, with real-time indicators of each facility's operational status (open, closed, maintenance and current capacity). Integrated weather data further enhances this by highlighting temporary closures due to adverse conditions such as lightning. The platform also offers consolidated information on each venue, including supported sports and available booking slots. Lightning Cat enhances user convenience, maximizes facility use, and promotes community participation – all through a single, intuitive interface. Visit our platform at: https://web.chucklenuts.party/

From Problem to Platform: Feature Design Grounded in Real User Needs

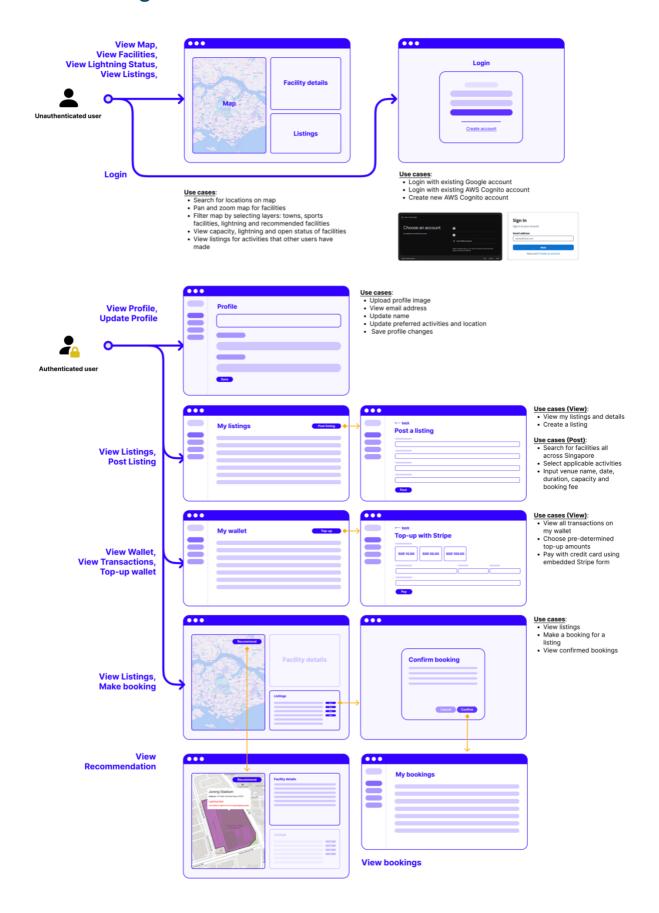
Our analysis finds that the government's ActiveSG offering covers the basic functions of booking and paying for slots, in line with the government's control over the facilities. However, auxiliary needs or quality of life (QoL) features that would improve user experience are evidentially not met from the mushrooming of community initiatives that had sprouted to address the gaps.

At present, no direct competitor offers a consolidated platform that meets these user needs. Instead, residents are left to navigate a patchwork of grassroots Facebook groups and Telegram channels, each focused on specific sports or venues. This fragmented ecosystem results in inefficiencies, unreliable communication, and a poor user experience.

Our research confirmed that the data and services required to address the above identified pain points are already publicly accessible through APIs —such as weather alerts (data.gov.sg, 2025), facility capacity statuses (ActiveSG, 2025), and venue information (data.gov.sg, 2025). However, these resources are dispersed across various platforms, lack real-time integration, and are not user-friendly for the average resident. The core issue is not the absence of data, but the lack of a centralized, intuitive system that aggregates and presents this information in a meaningful way.

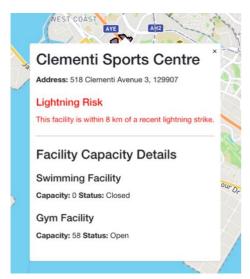
In response, we designed our platform and its features specifically to complement the existing government offering and bridge this identified gaps. Each capability is directly mapped to a user need, ensuring that the solution is targeted, relevant, and effective in addressing the real challenges faced by the community. In addition, here we introduce basic SaaS functions incorporated to support the overall platform.

User Flow Diagram



Platform Features

Real-Time Visibility of Facility Closures



Challenge: Swimming pools and other outdoor facilities may close without prior notice due to weather conditions such as lightning alerts. Residents often arrive unaware of these closures, wasting both time and prepaid pool passes that are valid only on the day. Currently, real-time weather-related closures are only available through APIs or Meteorological Service Singapore's site, which is not user-friendly or well known (MeteorologicalServiceSingapore, 2025).

Solution: Users can view which facilities are currently affected by closures and visually identify nearby alternative facilities that remain open (if any).

Unified Access to Booking Availability

Available Listings

Sport	Sports Center	Venue	Date	Duration (hrs)	Capacity	Fee	Status	Action
Football	Jurong Stadium	Football Stadium	30 May 2025	4	1/10	\$20.00	open	<u>View your</u> <u>booking</u>
Table_tennis	Yishun Sports Centre	Table Tennis	5 May 2025	2	1/3	\$10.00	open	Book

Challenge: Securing facility slots is challenging due to high demand and the lack of a centralized booking platform. Slots are often advertised informally via grassroots Facebook groups and Telegram channels. This makes it difficult for users to find and secure suitable slots for their preferred sport.

Solution: Users can instantly view available listings for each facility and proceed with bookings directly through the platform.

Connecting Users for Group Activities

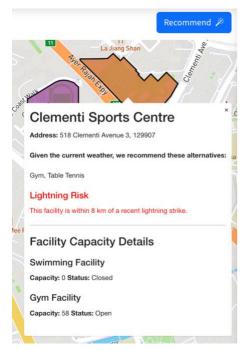
My Listings



Challenge: Many users who successfully book time slots are unable to find others to join them, especially for team-based sports. As a result, bookings go unused despite strong demand.

Solution: Using user preferences and real-time data, our machine learning algorithm can recommend nearby facilities or alternative timeslots. For example, if a lightning alert affects a user's preferred outdoor location, the system can suggest nearby indoor gyms or sports with available capacity.

Live Recommendations Based on Dynamic Conditions



Challenge: When conditions change suddenly – such as a last-minute lightning alert or venue reaching full capacity – there's no system to suggest alternative options tailored to user preferences.

Solution: Using user preferences and real-time data, our machine learning algorithim can recommend nearby facilities or alternative timeslots. For example, if a lightning alert affects a user's preferred outdoor location, the system can suggest nearby indoor gyms or sports with available capacity.

Comprehensive User Account Management

Feature: Users can log in through a secure, single sign-on (SSO) system powered by AWS Cognito, enabling access to personalized features and seamless authentication.

Feature: Users can top up their wallet with our payment system integrated and withdraw the money accumulated as well

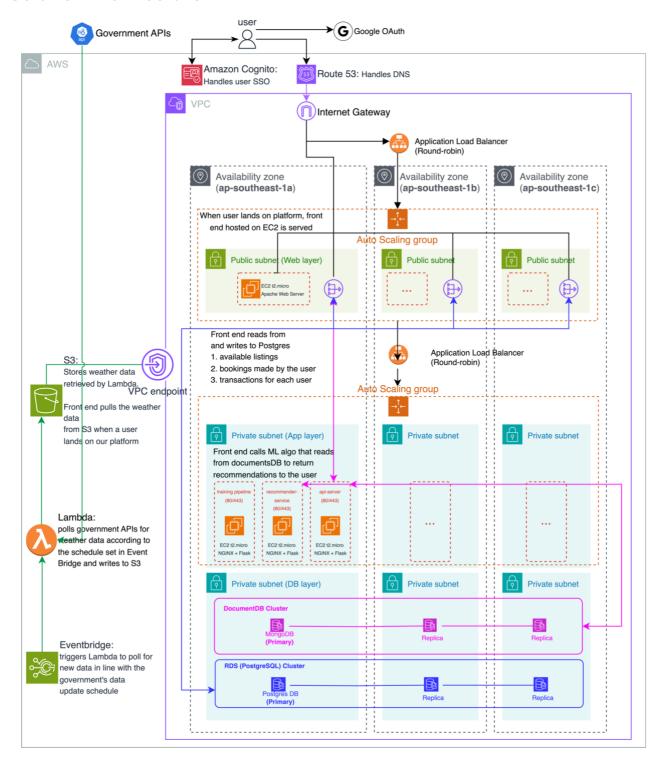


My Wallet Credits Your current wallet balance is \$80.00.

Transaction history

ID	Amount	Туре	Status	
2	\$20.00	payment	completed	
1	\$100.00	deposit	completed	

Solution Architecture



Cloud Services Underpinning the Platform

Although the platform's interface is intentionally simple, its core functionality relies on critical backend components powered by cloud infrastructure. Our decision to adopt cloud services was guided by the need to balance rapid time-to-market with robust support for mission-critical capabilities, without the burden of maintaining infrastructure ourselves.

Essential backend needs – such as secure user management, performance monitoring, and high availability – are not part of our core value proposition. Given their complexity and our team's limited expertise, it was prudent to adopt mature, cloud-native solutions. These services reduce development overhead by offering scalable, reliable, and well-documented functionality that simplifies concerns over reliability and scaling.

We unanimously selected AWS due to time constraints and team familiarity. While alternatives offer similar features, AWS allowed us to work efficiently with minimal ramp-up. Each team member could confidently adopt AWS tools suited to their domain: front-end developers with EC2, our data engineer with Lambda – resulting in a form of soft vendor lock-in shaped by existing knowledge rather than external constraints.

Function as a Service (FaaS) Utilised

AWS Lambda with AWS Event Bridge Schedular

We used AWS Lambda together with EventBridge Scheduler to periodically poll government APIs for facility (gym/pool) capacity status and weather data. This approach was significantly easier than managing a custom cron job with retry logic, monitoring, and uptime concerns. As Lambda offers one million free invocations per month (AWS, AWS, 2025), and EventBridge Scheduler provides up to 14 million free invocations monthly (AWS, 2025), our utilisation from our predictable and low-frequency workload of approximately 7000 API calls per month, keeps us well within the free tier. Compared to running scheduled tasks on a fixed server, this approach removes the burden of server provisioning and improves reliability.

Platform as a Service (PaaS) Utilised

AWS Cognito User Management

We used AWS Cognito as a fully managed authentication service to handle user sign-up, sign-in, and session management. As a SaaS solution, Cognito allowed us to offload complex identity management tasks while integrating secure, standards-compliant single sign-on (SSO) with minimal configuration. This was especially important given the sensitivity of user data and the team's limited experience with building secure authentication flows from scratch.

AWS Database Solutions - S3, RDS and DocumentDB

We used a combination of Amazon S3, RDS, and DocumentDB to manage different types of data in our application, selected based data characteristics and access patterns.

Amazon S3 stores the JSON output from our above AWS Lambda polling functions that gather the government's API data. This data is static once written, read frequently by the frontend, and does not require complex querying. Moreover, for our expected access volume, the usage falls within the AWS Free Tier, making it ideal for this type of append-only, read-heavy workload.

Amazon RDS (PostgreSQL) was chosen for managing user account information, booking records, and facility listings – data that is relational, frequently updated, and requires transactional integrity. Unlike S3, RDS provides a managed SQL engine with support for ACID transactions, indexing, and complex joins. These features are essential for maintaining business logic and ensuring data consistency across interrelated entities such as users and bookings.

Amazon DocumentDB was chosen to store data related to our ML model primarily because our JSON schema evolved constantly, making DocumentDB's flexible NoSQL structure fit well here. Additionally, as our team is already familiar with MongoDB, DocumentDB's compatibility let us work efficiently with minimal onboarding.

AWS's flexibility allowed us to match each dataset to its best-fit technology, tap on our team's individual database expertise and made it easy to reassess choices as the project evolved.

Infrastructure as a Service (IaaS) Utilised

AWS EC2

We selected EC2 to host our web and API servers, prioritising control over the software environment, deployment process, and update cadence. This allowed us to directly push updates via scp, which was faster and more flexible than going through a managed deployment service like Elastic Beanstalk. While Elastic Beanstalk offers automation, its abstraction layer would have added unnecessary complexity during our rapid prototyping phase. EC2 let us fine-tune configurations and iterate quickly.

EC2 also hosts our ML model that is our recommendation engine. Our recommendation system transforms facility data (locations, activities, capacity) into vector embeddings through a structured ML pipeline. When users request recommendations, their preferences are similarly converted to embeddings, allowing our vector search engine (Faiss) to identify initial matches. The system then applies real-time filters for weather conditions and capacity to refine results, dynamically suggesting indoor alternatives when outdoor activities are compromised by weather. All of this is accessible through a simple POST API for frontend integration. EC2 is ideal for hosting this ML system because it offers the necessary GPU options for scaling and accelerating vector operations and model inference needed to handle variable user traffic and complex vector searches.

AWS Auto Scaling and Load Balancer

To handle unpredictable user demand, we implemented Auto Scaling alongside a Load Balancer. Auto Scaling automatically adjusts the number of EC2 instances in response to real-time traffic patterns, eliminating the need for manual intervention. The Load Balancer works in tandem by efficiently distributing traffic across available instances, ensuring high availability and fault tolerance. This setup not only protects us from service degradation during traffic spikes but also keeps infrastructure costs efficient during low-load periods. More importantly, it gives us operational breathing room—removing the burden of traffic management so we can concentrate on refining core features and learning user demand dynamics over time.

Software as a Service (SaaS) Utilised

AWS IAM

Outside of the explicit offerings, AWS also provides us a comprehensive and free IAM solution. Instead of maintaining our own multi-user authentication and authorisation workflow, IAM provided better security through their comprehensive offering, allowing us to address access control concerns to keep accounts safe and allowed our team to focus on application development.

Google OAuth

We added Google OAuth as an alternative login method to lower the friction for potential users who may not want to create a new account. It also provides some redundancy for our authentication mechanism,

such as when AWS Cognito might be unavailable. Lastly, because our authentication system is architected around multiple OIDC providers, it is easier to tack on more providers in future as and when needed.

Stripe

We integrated Stripe's payment infrastructure for our wallet functionality instead of building our own solution. This saved significant development time while providing strong security compliance (PCI DSS Level 1), fraud detection, and automated verification processes. Stripe securely handles payment data storage and transactions, reducing our security risks. Their ready-made components and clear documentation helped us implement quickly, while giving users confidence through a trusted payment processor.

On-premise vs Cloud Native Implementation

Our cost analysis compares on-premises and cloud-native deployment options, using projected load estimates based on the government's reported ActiveSG usage (339,000 bookings and 307,000 users in Q4 2024 (OpenGovernmentProducts, 2025)). While we do not expect to reach such levels in the first three months, these figures help us define a performance envelope. With added buffer, we assume 200 peak hourly bookings and 200 concurrent users. Our infrastructure choices are benchmarked against these metrics.

Our on-premises setup follows our past startup experiences, where a single server is deployed from a home environment to minimise initial costs. We have chosen Dell Poweredge R240, an entry level server. Approximate costs are taken from BuyServer.SG (BuyServer.SG, 2025) and rounded up for ease of calculation. Hardware upgrades to support our max concurrency estimates were also included. Only major costs are included in our cost estimation.

On-premises costs for one quarter, at 200 max concurrent users, rounded to next hundred:

Туре	Item	Quantity/Usage	Cost (SGD)
Capex	Dell PowerEdge R240 Server	1	2,000
Capex	DDR4 8GB ECC Memory Upgrade	3	600
Capex	Enterprise SATA SSD Upgrade	2	1,600
Opex	Electricity + Internet	100 * 3 months	300
		Total Cost	4,500

Our cloud-based setup however, heavily utilises AWS's free-tier offerings where applicable and is in line with standard best practices such as splitting front end and back end servers onto separate EC2 instances and setting up load balancing and auto scaling. Our cloud costs are taken from AWS Pricing (AWS, 2025).

Cloud costs for one quarter, at 200 max concurrent users, rounded to next hundred:

Туре	Item	Quantity/Usage	Cost (SGD)
Opex	EC2 t2.micro Instances	3 * 24h * 31d * 3 months	0 (Free Tier)
Opex	Application Load Balancer	1 month free + 30 * 2 months	100
Opex	Route 53 DNS Service	100 * 3 months	300
Opex	DocumentDB	1 month free + 100 * 2 months	200
Opex	Relational Database Service	< 20 GB / month * 3 months	0 (Free Tier)

Туре	Item	Quantity/Usage	Cost (SGD)
Opex	S3 (PUT + GET Requests) 6,944 PUT, 148,800 GET * 3 months		0 (Free Tier)
Opex	Lambda	6,944 * 3 months	0 (Free Tier)
Opex	EventBridge	6,944 * 3 months	0 (Free Tier)
		Total Cost	600

Business Sustainability

Our business will generate cash flow by taking a \$1 cut from every booking transaction on our platform, charged to the party who listed the slot. User testing confirmed that users are willing to accept this platform fee for the benefit of accessing more players, with most intending to simply pass the cost to the booking party. Our platform provides value over sport specific community pages by allowing listings to reach users outside their specific sport community. This creates a two-way benefit: people listing slots can find more players, while users interested in trying different sports can discover new opportunities. For example, someone from the badminton community would be able to see basketball slots as well.

From the estimation, using cloud development saves us about 86.6% in initial development costs. Further, assuming a 5% conversion rate of users to bookings facilitated, it would take 60 days (600 / (200/hour * 0.05)) to break even on our cloud solution and 450 days to break even for our on-premises solution. Should our assumptions prove to be incorrect, utilising the cloud route would allow us to recoup our investment must faster. Hence, our team strongly endorses utilising cloud.

User Testing Results and Future Work

Our project was inspired by the parliamentary question (PQ) submitted by NMP Mr Ong, in which he highlighted the information dissemination gap.

Throughout the development of our prototype, we consulted with Mr Ong who provided valuable feedback that shaped our feature design and we thank him for the valuable feedback. When we shared the government's API with the capacity and lightning statuses, Mr Ong immediately adopted and integrated checking the APIs as part of his routine, demonstrating the practical value of our solution.

We recognise and acknowledge that our prototype did not address all the pain points and QoL asks given by Mr Ong and users, such as having our ML suggest when the user can next engage in their preferred sports, push notifications and having a mobile friendly webapp. These are in the roadmap for execution.

Mr Ong asked if we plan to share our solution with ActiveSG/MCCY. This question aligns with our analysis of our position in relation to existing government initiatives. Given that the Singapore government has a team of nine dedicated to developing the ActiveSG platform, we believe our work could become obsolete if our features are adopted into their roadmap. This reality presents an opportunity rather than a challenge – our best path forward may be to exit by selling our solution and knowledge to the government. This approach would allow our innovations to reach the widest possible audience while leveraging the government's established infrastructure and user base. Our solution offers significant social impact by encouraging users to make new friends, strengthening community through sports activities, making it easier to engage in sports, increasing workout frequency, and ultimately improving health outcomes - objectives shared by government health and community initiatives.

Team Contribution

Chee Han Quan, Edison: Cloud architecture diagram, api-server development (Flask + uWSGI + Nginx), session management, database schema and API design for Users, Bookings, Listings, Wallet, Transactions, Authentication, Payments. Frontend development: integration with OAuth, backend API integration, frontend routing, user account-related pages, empty states, logo design

Le Tan Dang Khoa: Backend service development (API design, integration with other components, ML inference), ML model development & deployment, database scheme design, project report writing.

Lai Foong Ming: Solution design: cloud solution design, UI design and flow, data flow logic. Data engineering: Data research and processing. AWS: Implemented IAM, VPC, EC2, S3, Lambda, EventBridge Schedular FE: Architecture and base code, map, side panel implementation. Report writing: Business case concept and development, component analysis and cost analysis.

Rifqi Rafifandi: Frontend development (pulling lightning data from public API to be displayed on the frontend, calculating whether a sports center is within a certain radius from a recent lightning strike), project report writing.

Bibliography

SportSingapore. (2024, April). Sport Singapore. Retrieved from Sport Singapore:

https://www.sportsingapore.gov.sg/activesg-celebrates-10th-anniversary-with-islandwide-celebrations-free-entry-to-pools-gyms/

MCCY. (2023, Nov). *MCCY*. Retrieved from MCCY: https://www.mccy.gov.sg/about-us/news-and-resources/parliamentary-matters/2023/Nov/Provision-real-time-updates-on-closures-swimming-pools-in-ActiveSG-app

ChannelNewsAsia. (2024, Nov 22). Retrieved from https://www.channelnewsasia.com/today/ground-up/badminton-court-booking-hosts-profit-4759586

Facebook. (2025). Retrieved from https://www.facebook.com/groups/147985058667079/about

PMO. (2024, August 18). *Prime Minister's Office Singapore*. Retrieved from Prime Minister's Office Singapore: https://www.pmo.gov.sg/Newsroom/National-Day-Rally-2024

MeteorologicalServiceSingapore. (2025, April). *Meteorological Service Singapore*. Retrieved from Meteorological Service Singapore: https://www.weather.gov.sg/lightning/lightning/lightningalertinformationsystem.jsp

ActiveSG. (2025, April). Retrieved from https://activesg.gov.sg/api/trpc/pass.getFacilityCapacities

data.gov.sg. (2025, April). Retrieved from https://api-open.data.gov.sg/v2/real-time/api/weather?api=lightning

data.gov.sg. (2025, April). Retrieved from https://api-open.data.gov.sg/v2/real-time/api/two-hr-forecast

ActiveSG. (2025, April). Retrieved from https://activesg.gov.sg/

OpenGovernmentProducts. (2025). Retrieved from OpenGovernmentProducts: https://products.open.gov.sg/activesg

BuyServer.SG. (2025). Retrieved from BuyServer.SG:

https://www.buyserver.sg/product_info.php?cPath=29_31&products_id=137&osCsid=d81f3mdk2ce1otno5c6d8ifm85

AWS. (2025). Retrieved from AWS: https://aws.amazon.com/pricing/