

A Brief Introduction Of Five Guys	2
Five Guys Team Health Monitor	4
Persona and Empathy Map of stakeholders	8
Persona of Alex Chen	9
Empathy Map of Alex Chen	11
Persona of Elara Swiftarrow	14
Empathy Map of Elara Swiftarrow	16
Risk Assessment Matrix	19
Meeting notes in space	22
Meeting notes at 07/03/2024	23
Meeting notes at 21/03/2024	24
Meeting notes at 14/03/2024	26
Meeting notes at 18/4/2024	28
Meeting notes at 11/04/2024	30
Meeting notes at 29/03/2024	32
Meeting notes at 04/4/2024	34
Meeting notes at 26/4/2024	36
Meeting notes at 2/5/2024	38
Meeting notes at 9/5/2024	40
Meeting notes at 16/5/2024	42
Meeting notes at 20/05/2024	44
Five Guys Project Plan	45
Roles and Responsibilities	47
Working agreement	49
Team Member Profiles	52
My User Manual: Xuan Tuan Minh Nguyen	53
My User Manual: Phuc Vinh Le	54
My User Manual: Trong Dat Hoang	55
My User Manual: Phuong Doanh Ha	62
My User Manual: Mai An Nguyen	63
Product requirements	64
Entity Relationship Diagram	68
Physical database - create table statements	70
Documentation on data creation and null values	76
Use cases and SQL statements, transactions	79
Major-specific work	82
Team Reflection (4Ls Retrospective)	107
Performance (indexes)	109



A Brief Introduction Of Five Guys

Welcome to Five Guys

"Strength in Five, Success in Unity!"

Team metrics

Key metric #1

Updated 29 Feb 2024

ON TRACK / OFF TRACK

Key metric #2

Updated 29 Feb 2024

ON TRACK / OFF TRACK

Key metric #3

Updated 29 Feb 2024

ON TRACK / OFF TRACK

About Five Guys

Our team is made up of five vibrant people with a range of talents and creative abilities. We all strive towards the same objective - team success, blending individual talents and encouraging and accepting feedback from one another to foster productivity and a positive work environment.

Our objective is to fulfil stakeholder expectations and quality standards while finishing the project in the allotted period. In order to monitor the development, we offer the following metrics:

- Keeping a record of accomplishments for assignments and benchmarks to ensure timely completion.
- Examining defect rates, client satisfaction, and specification compliance in order to assess the quality of outputs.
- Periodical assessments that emphasize teamwork, communication, problem-solving, and role adherence.

Five Guys Blog

Now we got nothing to talk, so do not give any expectations for this, we are so tired of receiving expectations...

Team Members



@XUAN TUAN MINH
NGUYEN
Leader



@VINH LE
Member



@TRONG DAT HOANG
Member



@Phuong Doanh HA
Member



@MAI AN NGUYEN
Member

Featured resources

Our documentations about the project, working agreement, risk matrix, roles and responsibilities for each team member.

[🔗 😊 Five Guys Project Plan](#)

[🔗 📄 Working agreement](#)

[🔗 Team Underperformance Provision](#)

[🔗 😊 Risk Assessment Matrix](#)

[🔗 📄 Roles and Responsibilities](#)

[🔗 📄 Meeting notes in space](#)

[🔗 📄 Product requirements](#)

[🔗 📄 Persona and Empathy Map of stakeholders](#)

Resources

Restrict search to this space's space key.

Search

Search

📅 Major Deadlines

Project Proposal Assessment

Due date: Sunday, 24th of March, 2024 @ 11:59PM (Week 4)

Points: 20 / 100

Specific Details: [Project Proposal \(Week 4\)](#)

📝 Meeting notes

All Updates My Spaces Network

[📝 Meeting notes at 16/5/2024](#) (Five Guys)

TRONG DAT HOANG (May 20, 2024)

[📝 Meeting notes at 2/5/2024](#) (Five Guys)

TRONG DAT HOANG (May 20, 2024)

[📝 Meeting notes at 26/4/2024](#) (Five Guys)

TRONG DAT HOANG (May 20, 2024)

[📝 Meeting notes at 29/03/2024](#) (Five Guys)

TRONG DAT HOANG (May 20, 2024)

Reflection and Peer Assessment

Due date: Unknown (Expect Week 12)

Points: 10 / 100

Specific Details: [Reflection and Peer Assessment \(Week 12\)](#)

Final Report and Product Deliverable

Due date: Unknown (Expect Week 12)

Points: 25 / 100

Specific Details: [Final Report and Product Deliverable \(Week 12\)](#)

Meeting notes at 20/05/2024 (Five Guys)	TRONG DAT HOANG (May 20, 2024)
Meeting notes at 9/5/2024 (Five Guys)	TRONG DAT HOANG (May 20, 2024)
Meeting notes at 04/4/2024 (Five Guys)	Phuong Doanh HA (Apr 19, 2024)
Meeting notes at 11/04/2024 (Five Guys)	XUAN TUAN MINH NGUYEN (Apr 19, 2024)
Meeting notes at 21/03/2024 (Five Guys)	TRONG DAT HOANG (Apr 19, 2024)
Meeting notes at 18/4/2024 (Five Guys)	TRONG DAT HOANG (Apr 19, 2024)
Meeting notes at 14/03/2024 (Five Guys)	XUAN TUAN MINH NGUYEN (Apr 19, 2024)
Meeting notes at 07/03/2024 (Five Guys)	XUAN TUAN MINH NGUYEN (Apr 19, 2024)

Members profile

- Xuan Tuan Minh Nguyen
- Vinh Le
- Trong Dat Hoang
- Phuong Doanh Ha
- Mai An Nguyen

Latest updates

All Updates My Spaces Network

Major-specific work (Five Guys)	XUAN TUAN MINH NGUYEN (about an hour ago)
Documentation on data creation and null values (Five Guys)	MAI AN NGUYEN (about 3 hours ago)
Use cases and SQL statements, transactions (Five Guys)	MAI AN NGUYEN (about 3 hours ago)
(Five Guys)	TRONG DAT HOANG (about 3 hours ago)
Performance (indexes) (Five Guys)	MAI AN NGUYEN (about 3 hours ago)
(Five Guys)	TRONG DAT HOANG (about 3 hours ago)
(Five Guys)	TRONG DAT HOANG (about 3 hours ago)
(Five Guys)	TRONG DAT HOANG (about 3 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 3 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 3 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 3 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 3 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 4 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 4 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 4 hours ago)
(Five Guys)	XUAN TUAN MINH NGUYEN (about 4 hours ago)
Team Reflection (4Ls Retrospective) (Five Guys)	MAI AN NGUYEN (about 7 hours ago)
(Five Guys)	MAI AN NGUYEN (yesterday at 5:31 pm)
(Five Guys)	MAI AN NGUYEN (yesterday at 5:30 pm)
(Five Guys)	MAI AN NGUYEN (yesterday at 5:29 pm)

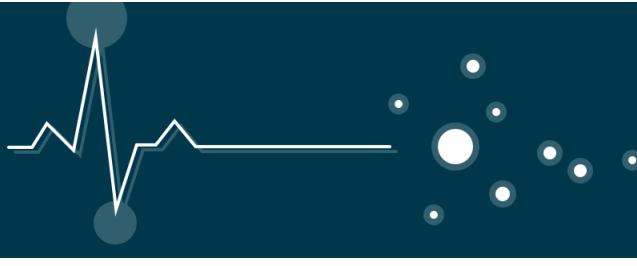
Where to find us

These are our methods of communication, don't hesitate to reach out with us, we love you so much 😊

<https://discord.com/gg>

fiveguysatswinburne@gmail.com

0428492011



Five Guys Team Health Monitor

- 💡 Use the project team health monitor template to keep track of your team's health. Keep this template in your team space and if there are any areas that you're not confident are green, dive into the plays to get back on track. For detailed facilitation instructions go to [health monitor for project teams](#).

Team name	Five Guys
Sponsor	Swinburne University of Technology
Health monitor cadence	Monthly

👉👈 Team health assessment

With your team, read the definition of each attribute of healthy, high-performing teams out loud. On the count of three have each person rate how they feel the team is doing compared to each definition (thumbs-up/green, thumbs-sideways/yellow, thumbs-down/red). Record the results of each attribute rating in the table. Highlight each cell using this color code: **HEALTHY** = "We're strong here", **BIT SICK** = "We're ok... but a little shaky", **SICK** = "We're not healthy".

Area	18 Feb 2024	18 Mar 2024	18 Apr 2024
🚩 Full-time owner There is one lead who is accountable for the result of this project. This needs to be someone whose time is at least 80% dedicated to it, and who can champion the mission inside and outside of the team.	The leader has effectively championed the project's mission while maintaining responsibility.	The leader Xuan Tuan Minh Nguyen is still demonstrating dedication and effective leadership.	The leader's dedication is unwavering, assuring project success.

 Balanced team Roles and responsibilities are clear and agreed upon. The project has people with the right blend of skill set. Acknowledge that team members can change by stage.	Roles and responsibilities are generally clear, but there have been occasional slight variations as a result of team composition changes.	Despite some fluctuations, the team has managed to maintain a balanced skill set.	Team composition has stabilised, but further work is needed to maintain consistency in skills and roles.
 Shared understanding The team has a common understanding of why they're here, the problem/need, are convinced about the idea, confident they have what they need, and trust each other.	The team has repeatedly showed a shared grasp of the project's objectives, as well as trust among members.	Collaboration and production have remained strong due to ongoing mutual understanding and trust.	The team's shared understanding and trust have become stronger, allowing for smooth collaboration.
 Value and metrics It's clear what success means from a business and user's perspective, and there is a unique value proposition in place for the target users and to the business. Success is defined, with a goal, and how it will be measured.	Progress has been made in identifying success criteria, but there is still space for growth in expressing a distinct value proposition.	Efforts have been made to improve metrics and value proposition, but more work is needed.	Metrics and value proposition have improved, and ensure alignment with corporate objectives.
 Proof of concept Some sort of demonstration has been created and tested, that demonstrates why this problem needs to be solved, and demonstrates its value.	There has been no major progress in designing and testing a proof of concept, which puts the project's success at danger.	Despite efforts, the proof of concept remains undeveloped and requires quick attention.	The lack of progress towards establishing a proof of concept remains a problem, needing immediate action.
 One-pager The project is summarized in a one-pager and shared with anyone so that they understand the purpose of the project, and its value.	While a one-pager exists, its distribution and accessibility require improved.	Attempts have been made to improve the clarity and circulation of the one-pager.	The one-pager has been developed and efficiently shared with stakeholders, but additional enhancements are required.

 Managed dependencies Clear understanding of complexity, infrastructure involved, risks, resources, effort, and timeline. Clear understanding of who we depend on, and who depends on us.	A comprehensive grasp of the project's complexity, risks, and dependencies has been maintained, resulting in seamless progress.	Proactive dependency management has helped to minimise disruptions and preserve progress.	Efforts to control dependencies have been successful, allowing for continuous advancement and success.
 Velocity The team is making incremental progress by shipping concrete iterations to stakeholders (and, even better, to production), learning along the way, and implementing lessons learned , resulting in greater success.	Progress has been achieved, there is still an opportunity for growth in both velocity and efficiency.	Efforts to streamline processes and increase velocity have been undertaken, but additional improvements are required.	Velocity has increased, but further efforts are needed to maintain development and success.

Focus areas

Ask your team to collectively come up with one attribute you want to focus on. Then, call out ways to move the **SICK** or **BIT SICK** toward **HEALTHY**. Make sure they are actionable, specific, and measurable.

Date	Focus areas and action items
18 Mar 2024	Focus Area: Velocity <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Conduct a retrospective to discover bottlenecks and inefficiencies in your present workflow. <input checked="" type="checkbox"/> Use agile methods like daily stand-up meetings and sprint planning sessions to improve communication and cooperation. <input checked="" type="checkbox"/> To keep team members motivated and moving forward, provide them with regular feedback and appreciation. <input checked="" type="checkbox"/> Break down work into smaller, more manageable chunks to boost momentum and productivity. <input checked="" type="checkbox"/> To properly track progress, define clear and achievable goals for each iteration.
18 Apr 2024	Focus Area: Value and Metrics <ul style="list-style-type: none"> <input checked="" type="checkbox"/> To properly track progress, define clear and achievable goals for each iteration. <input checked="" type="checkbox"/> Review and refine the existing success criteria from both a business and user perspective. <input checked="" type="checkbox"/> Conduct user surveys or interviews to gather feedback and validate the value proposition. <input checked="" type="checkbox"/> Regularly track and analyze metrics to assess the project's alignment with business goals and user needs.

- | | |
|--|--|
| | <ul style="list-style-type: none"><input checked="" type="checkbox"/> Collaborate with stakeholders to ensure continuous refinement of the value proposition based on market insights and changing requirements.<input checked="" type="checkbox"/> Establish key performance indicators (KPIs) to measure the project's progress and impact. |
|--|--|

👉 Next steps

Consider running the plays in suggested in [Step 2 of the facilitation instructions](#) for improving **BIT SICK** and **SICK** attribute areas. Don't treat these as prescriptions! You know your team better than anyone, so check them out, [explore other plays](#), and do what you think is best.

Five Guys has decided to focus on increasing collaboration. The prior strategy is to use a project management application to improve communication and task management.

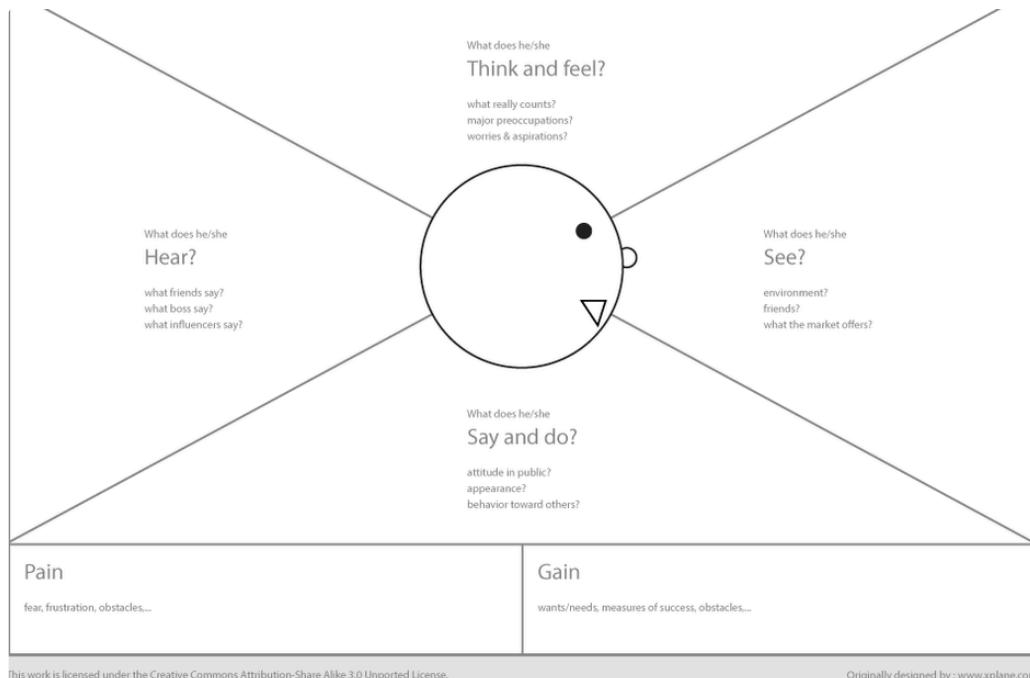
Based on this, we modify our strategy by using a mix of asynchronous stand-ups for daily updates and weekly synchronous video meetings for in-depth talks and team bonding activities. This hybrid method enables you to get the benefits of both asynchronous and synchronous communication while meeting the different needs of our team.

By adapting the communication strategy to our team's specific context and being willing to experiment and adapt, we can find a solution that improves collaboration and supports a healthy team environment.

Persona and Empathy Map of stakeholders

Create Personas

Template of Empathy map



Template of Empathy map

List of Personas

Persona name	Age	Gender	Persona role	Company name	Industry
Alex Chen	30	Male	Software Developer	TechSavvy Solutions, Inc.	Software development and IT services industry
Elara Swiftarrow	25	Female	Professional Archer	Elite Archery Association (EAA)	Sports

Persona of Alex Chen



Alex Chen

Persona name	Alex Chen
Persona role	Software Developer
Job description	Mainly work on backend development for scalable and robust mobile apps and web applications.

[Empathy Map of Alex Chen](#)

🏢 Company

Company name	TechSavvy Solutions, Inc.
Company size	Approx. 200 employees
Industry	Software development and IT services industry

👤 Demographic information

Age	30
------------	----

Gender	Male
Income	around 100,000 AUD per year
Education level	Bachelor's degree in Computer Science at Swinburne University of Technology
Residential environment	Melbourne CBD

✍ Personal quote

"Empowering innovation through collaboration and code."

📋 Biography

Grew up in a family that valued education and technology, he was interested in computers and technology from a small age. After studying Bachelor of Computer Science, he started working at a startup company, which quickly became well-known for a variety of projects. As the company gets larger over time, he gained more trust in the company and in his field. Even though he has 8 years of experience, he is still hardworking, humble, disciplined, grounded, progressive, and contributing more and more to the company. Currently, to balance his work and personal life, he tries several sports, including **archery**.

Professional goals	Motivators
<ul style="list-style-type: none"> • Deepen his knowledge of emerging technologies (i.e. AI, machine learning, cloud computing) • Lead his own projects, advancing his position 	<ul style="list-style-type: none"> • He is passionate about technologies generally and software development specifically • He wants recognition and achievement in his career
Challenges	Sources of information
<ul style="list-style-type: none"> • He experiences imposter syndrome - despite his achievements, he still feels insecure and pressured • He needs to balance between personal and work-life 	<ul style="list-style-type: none"> • He reads tech blogs and forums (i.e. Stack Overflow, Reddit) daily and follows Tech influencers online via Twitter and LinkedIn • He reads technical publications and news • He listens to tech podcasts and studies online courses

Aside from his work life, as he feels the need to balance and enjoy his life as well as making new relationships, he currently aims to:

- Get into new habits aside from job to relieve stress (he chooses **archery**)
- Maintain a work-life balance

To do so, he:

- Attend archery coaching programs,
- Chit-chatting with new people in the archery classes,
- and so on.

Empathy Map of Alex Chen

 Persona of Alex Chen

➊ What does he think and feel?

What really counts?

- He cares about the quality of his code and always wants to improve it for better efficiency
- He feels a sense of responsibility towards his projects and his team
- He feels the need to better take care about his body
- He wants to lessen stress and make new connections

Major preoccupations?

- This includes ensuring the code's scalability, maintainability, and performance
- He thrives to optimise his code and improve the architecture to meet future needs

Worries and aspirations?

- Alex worries about bugs and other issues within the project scope. He also is bothered by tight deadlines, as well as staying updated on the latest technology trends.
- Alex aspires to become a proficient and respected software developer and work on projects that push the boundaries of technology, as well as contribute positively to society.
- He is bothered that his health is not as good as before.

➋ What does he see?

Environment

- Working environment
 - An environment where he could positively develop his skills, as the company supports their workers with the latest technologies
 - He sees the environment as a structured, professional environment that takes a project-based approach to technical projects using project boards or task management tools displaying the progress of various assignments and deadlines
- Sports related environment
 - A friendly community where people share similar interests
 - Have several competitive individuals who still show respect towards their opponents

Friends

- See opportunities for learning from more experienced developers or people having interest in sports
- See diverse perspectives and experiences that contribute to meaningful discussions related to the technology or sports field
- See different approaches of problem-solving from people with different demographics

What the market offers?

- Working environment
 - Demand for tech talents in various companies in different industries

- Technological advancements and trends shaping the future of software developments like AI and cloud computing
- A lot of online courses, training programs, workshops, and certifications to enhance technical skills for people with different backgrounds
- A lot of conferences, meetups, and forums, where people in the field could connect with others having the same background as them
- Sports related environment
 - Chances to make friends
 - Opportunities to improve health

② What does he say and do?

Attitude in public?

- He maintains a professional and confident image, following social norms and etiquette appropriately. He communicates with others with clarity and with relaxation. He shows enthusiasm towards his projects, technologies, and software development. He also shows interests in sports, namely swimming and archery.

Appearance

- Often wears jeans with a T-shirt, sometimes a jacket thrown over his shoulder
- Looks relaxed with a friendly smile
- Wears a glasses
- Often seen with a laptop

Behaviour toward others?

- He collaborates effectively with his colleagues, acknowledging the value of teamwork. He communicates openly when sharing information, ideas, and resources and respects the contributions of his team.
- He seeks guidance and feedback from his boss, and mentors, generally those who are more experienced, and demonstrates a willingness to learn and develop his skills in both work and sports.
- He communicates professionally regarding the project's progress, challenges, achievements, and others related to his project.
- He provides support and encouragement when working with those having less experience, shows openness to their questions and asks for feedback, is patient with their struggles and gives them recommendations.

② What does he hear?

What friends say?

- Encouragement and support, such as he is a hard worker and shows dedication towards his career
- Alex and colleagues or those in technological field may have technical discussions about the latest technologies, sharing insights and knowledge regarding the topics related
- Alex and people he met in sports centres or via the archery coaching lessons may talk to each other regarding sports techniques and sports news
- Others and Alex would share personal updates

What boss say?

- His boss would give project updates regarding the timelines, milestones, and deliverables, keeping Alex informed and aligned with team goals
- Alex listens to his boss' feedback on performance and encouragement

- His boss offers him opportunities, such as training programs, workshops, or mentorship opportunities to support his growth

What influencers say?

- He hears industry insights and perspectives from people in the tech community, such as experts or prominent figures in software developments
- He updates trends of technologies or sports from his community, like forums, workshops, meetups
- He receives career advice and guidance from people with more experience than him, such as his mentors, colleagues, and other experts who might help him navigate his career path

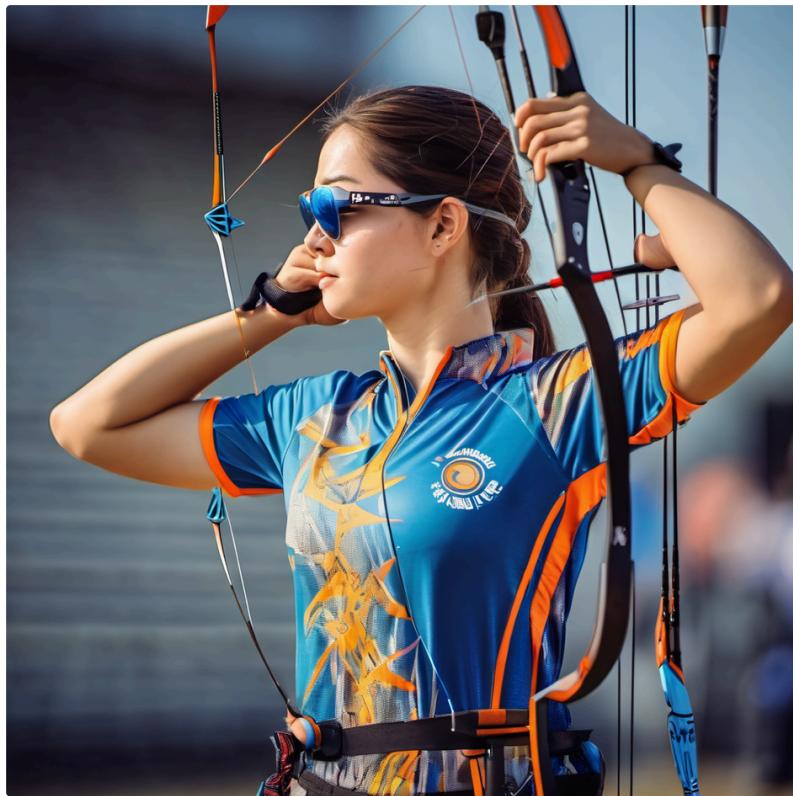
Pain

- Frustration when encountering complex technical challenges or bugs
- He works under tight deadlines while being a perfectionist, which could lead to emotional burnout
- He might experience uncertainty about his career growth opportunities
- He faced physical hindrance as he did not usually exercise

Gain

- He gains important skills, such as problem-solving skills, communication skills, and so on
- He gets new valuable connections and learning opportunities
- He balance his life and enjoy his life more
- He receives recognition and positive feedback from those surrounding him

Persona of Elara Swiftarrow



Elara Swiftarrow

Persona name	Elara Swiftarrow
Persona role	Archer
Job description	<ul style="list-style-type: none">masters the art of precision, focus, and control, responsible for honing their skills in archerymaintain equipment, practice shooting techniques, and participate in competitions in the Professional Archery Circuit

[Empathy Map of Elara Swiftarrow](#)

🏢 Company (Organization/club)

Company name	Elite Archery Association (EAA)
Company size	small to medium-sized organisation
Industry	Sport

👤 Demographic information

Age	25 years old
Gender	Female
Income	AUD 40 000 - 60 000, depending on success in tournaments and sponsorship deals
Education level	High school, while have specialised training in archery through coaching programs and sports academies
Residential environment	Urban area - Melbourne, Australia

✍ Personal quote

"Draw back, aim true, and let determination be your arrow."

📜 Biography

Growing up with her parents, who usually go to the archery club every weekend, she discovered her passion for this sport at a young age. She decided to pursue her dreams of becoming a professional archer in her junior high school years, when she went into puberty, had ups and downs in her emotions, and found out more about herself that she feels truly happy when doing archery. From then on, she started advanced coaching programs and joined competitions, where she became successful. After rising in rankings, she ended up choosing her current club and continues to compete in the archery world.

Professional goals	Motivators
<ul style="list-style-type: none"> • Have achievements in national and international competitions, i.e. National Championships, World Cup Series, World Archery Championships, Olympic Games, and so on • Continue to rise in rankings • Empower other female archers and promote the sport 	<ul style="list-style-type: none"> • Passion for Archery • Desire to succeed and obtain achievements • Be competitive • Inspire others
Challenges	Sources of information
<ul style="list-style-type: none"> • Intense competition as there are various skilled archers from around the world • Injuries or fatigue might occur due to physical and mental demands • Financially demanding as she needs to keep her rankings to obtain sponsorships and achievements 	<ul style="list-style-type: none"> • Coaching and Training programs • Professional archers' communities • Sports mental and physiology training • Sport news

Empathy Map of Elara Swiftarrow

 Persona of Elara Swiftarrow

➊ What does she think and feel?

What really counts?

- Elara is success-driven and cares about her achievements
- She is proud of the achievements she got so far
- She wants to do archery as long as she can
- She wants support from her family and friends

Major preoccupations?

- Emily cares about how to improve her archery skills
- She is focusing on raising her rankings

Worries and aspirations?

- She worries that she could not compete with other archers and be less skilled in her field
- She worries that she might become financially unstable to continue her career
- She wants to contribute more and inspire others, especially female archers

➋ What does she see?

Environment

- She sees the environment around her as friendly and competitive, as there are a lot of ambitious archers in the community who have the same goal and improve together
- She sees the environment around her as diverse, as in her club, there are people with different backgrounds and different levels of archery skills (both professionals and amateurs)

Friends

- She sees diverse perspectives and backgrounds in her social circle, including members, coaches, and staff of the club, professionals and amateurs doing archery, as well as other friends.
- She sees relationships built on trust and respect toward each other, both from her comrades and opponents.
- She sees opportunities for networking and professional development.

What the market offers?

- In sport industry, especially archery, she sees:
 - Opportunities to run business relating to equipment, sport outwears, etc.
 - She receive sponsorships from these brands
 - Training and programs for people to learn and improve skills in archery
 - People from different age groups with different skill levels and different backgrounds attend the archery classes and interact with each other

- Competitions and events for skills development and acquiring achievements
 - Teams work together and encourage each other to achieve the goal
 - People compete with their opponents - their interaction could be friendly or not
 - Individuals make friends as they have the same passion
- Chances to make friends and open connection with people with similar passion

② What does she hear?

What friends say?

- She hears supportive encouragement, as well as praise like she is a hard worker, a perfectionist, and so on.
- She hears feedback to improve her archery skills
- She hears news regarding archery

What boss (mentors) say?

- She hears recommendations to improve her skills
- She hears her mentor's encouragement and support when she has done fine
- Mentors may ask her to assist them in coaching others

③ What does she say and do?

Attitude in public?

- She seems polite, calm, and serious
- She is a quiet person, still, she communicates confidently and straightforwardly

Appearance

- She wears sports clothes (a t-shirt with shorts)
- She is usually seen while wearing hats and sports glasses

Behaviour toward others?

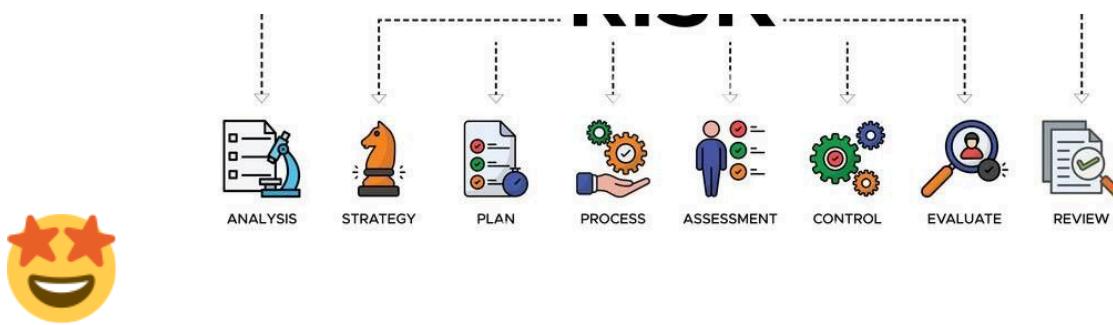
- She treats others around her with respect
- She tends to be a listener
- She is protective towards her teammates and her juniors
- She is all ear to feedback towards her
- She shows professionalism and confidence when stating her opinion to anyone

➊ Pain

- Handle unwanted scenarios while competing, or having a bad match
- Continuously have mental and physical pressure
- Lose other archers
- Resolve conflicts in her team

Gain

- Improving archery skills
- Rising in ranking
- Building relationships with her mentors, seniors, juniors, teammates, opponents, etc. (those with the same goal as her)
- Influence others around her (as a hard-working, calm, ambitious, detail-oriented person)



Risk Assessment Matrix

Background

Five Guys's Target archery groups frequently use manual ways to record and manage results from practice sessions and tournaments. These manual operations can be time-consuming, error-prone, and inefficient. As a result, there is a need for a modernized solution to improve score recording and management in archery groups.

Five Guys aims to centralize a platform for archery clubs to digitally record, manage, and analyze scores from various competitions and practice sessions. The database will include several critical elements designed to increase the efficiency and accuracy of score recording, as well as data administration inside the organization.

Therefore, the objective of Risk Management is to propose and coordinate the roll of plans designed to reduce or change the profile of risks as they arise.

Risks management

Identify and prioritize risks based on their probability and severity. Then define what further actions you need to take to control the risks, and who needs to carry out these actions.

The first stage in risk management is identifying potential sources of uncertainty and problems that may affect the project. These include technical challenges like bugs, compatibility, performance, security, or scalability. And even the company's changing needs.

Risk rating

LOW	MEDIUM	HIGH	EXTREME
<ul style="list-style-type: none"> Acceptable Ok to proceed 	<ul style="list-style-type: none"> As low as reasonably practicable Take mitigation efforts 	<ul style="list-style-type: none"> Generally unacceptable Seek support 	<ul style="list-style-type: none"> Intolerable Place event on hold

SEVERITY			
ACCEPTABLE <i>Little to no effect on event</i>	TOLERABLE <i>Effects are felt, but</i>	UNDESIRABLE <i>Serious impact to</i>	INTOLERABLE <i>Could result in disaster</i>

LIKELIHOOD		<i>not critical to outcome</i>	<i>the course of action and outcome</i>	
IMPROBABLE <i>Risk is unlikely to occur</i>	Minor Budget Adjustments: Small variations from the original budget have no substantial influence.	Technical Challenges: Complexity in implementing certain features, compatibility issues	Performance Issue: Inefficient coding practices, poor server infrastructure, or inadequate testing.	Insecure Development: Coding errors, such as attackers may use coding mistakes, unauthorized access to the website or sensitive data.
POSSIBLE <i>Risk will likely occur</i>	Low Priority Bugs: Minor flaws or issues detected during testing with low impact.	Resource constraints: Temporary shortages of resources, whether human resources or hardware.	Team Member Turnover: Changes in team composition may lead to reassignment or turnover.	Data Breach: A significant breach of sensitive data may result in legal penalties, reputational damage, and financial losses.
PROBABLE <i>Risk will occur</i>	Minor Design Changes: Small adjustments to the website's design or layout.	Quality Control Issues: Deliverables with lower-than-expected quality may require a further examination.	Project scope creep: Uncontrolled expansion of a project's scope, which leads to potential failure.	Cyberattacks: Advanced cyberattacks potentially interrupt website functionality, endanger data integrity, and cause considerable financial loss.

✓ Action items

Scenario: The cost of the project can be adjusted over the expected budget.

Specific case and action items: Budget overruns result from scope creep driven by frequent changes in project outcome requirements.

- Make sure that the budget for the project is considered enough for all the tasks of the project
- The project manager implemented some minor change management processes, assessing suggested changes for budgetary and schedule implications.
- Project team meetings will be conducted to prioritize features that align with project goals, while extensive documentation records modification costs to track the budget situation.
- All suggested solutions will be reviewed and approved by the Project Manager.

Scenario: A significant technical risk arises due to a major error in the database infrastructure change.

Specific case and action items: The project team faces technical risks due to the complexities of migrating legacy systems.

- Database Designer and The project team members will be arranged to conduct extensive reviews of current systems.
- Meetings will be conducted to discuss potential solutions.
- Incremental Migration: Project managers prioritized systems for incremental migration.

- The project team conducts migration performance to ensure optimal performance upon migration. Action will be approved by the Data designer and Project Leader.

Area	Treatment Control Methods	Responsibility
Project Scheduling	Make sure that the schedule for the project (six months) has considered time enough for all the task of the project.	Responsibility for Project team members. Solutions approved by team Leader.
Product Quality	Use static code analysis techniques to detect possible bugs in the codebase, and adhere to coding conventions and best practices to guarantee code maintenance.	Responsibility for Technical manager. Solutions approved by team Leader.
Technical	Implement technical review methods to find and correct coding, and technical errors. Conduct extensive testing, including unit, integration, and system tests, to ensure functional quality and stability.	Responsibility for Technical Manager. Solutions approved by team Leader and Technical Manager.
Project Budget	Make sure that the budget for project has considered time enough for all the tasks of the project.	Project Resources managing and Researcher. Solutions approved by team Leader.

Meeting notes in space

[Create meeting note](#)

Incomplete tasks from meetings

Task report

Looking good, no incomplete tasks.

Decisions from meetings

Page Title	Decisions
No decisions found	

All meeting notes

Title	Creator	Modified
No content found.		

Meeting notes at 07/03/2024

Date

7 Mar 2024

Participants

- @XUAN TUAN MINH NGUYEN
- @MAI AN NGUYEN
- @VINH LE
- @TRONG DAT HOANG
- @Phuong Doanh HA

Goals

- Planning strategies on how to use the Jira and the Confluence in the peak performance and effortless.
- Game-on some tactics to finish the job exactly ON-TIME and HIGH-PERFORMANCE.

Discussion topics

Time	Item	Presenter	Notes
15:00PM	Confluence / Jira	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">• Give a quick hands on how Jira or Confluence work.• Show every team member how to effectively use Jira and Confluence to manage the work stream.
15:30PM	Confluence / Jira	@MAI AN NGUYEN	<ul style="list-style-type: none">• Distribute the tasks for each team member.Working on Integrating Project plan, Roles & Responsibilities onto the content space

Action items

- @XUAN TUAN MINH NGUYEN Integrate Jira into the general space
- @MAI AN NGUYEN Add project into the general space
- @Phuong Doanh HA Finish designing the introduction page
- @VINH LE @TRONG DAT HOANG Approving fixed content

Decisions

 Prioritize on finishing all of the task on the Action items

 Members must read extra materials for understanding the mechanism of Jira

Meeting notes at 21/03/2024

Date

21 Mar 2024

Participants

- @XUAN TUAN MINH NGUYEN
- @VINH LE
- @TRONG DAT HOANG
- @Phuong Doanh HA
- @MAI AN NGUYEN

Goals

- Review finished tasks
- Set and allocate tasks for the Project Proposal
- Set deadlines for each task

Discussion topics

Time	Item	Presenter	Notes
22:00PM	Listing tasks	@TRONG DAT HOANG	<ul style="list-style-type: none">• Reading the Project Plan's requirements• Need to do according to the rubric: Project Plan, Roles and Responsibilities, Persona and Empathy map, Initial ER diagram• Submission requirements:<ul style="list-style-type: none">◦ Project Proposal:<ul style="list-style-type: none">✓ Team Homepage (including a provision for team member underperformance and Confluence Working Agreement)□ Project Plan□ Roles and Responsibilities✓ Risk Assessment□ Persona<ul style="list-style-type: none">□ Empathy map□ Product requirements□ Initial ER diagram□ Appendix 1: Team member profiles□ Appendix 2: Team meeting notes◦ Team Agreement (export separately to submit under the assessment)• Each member needs to finish their Personal Profile (Personal Profile's diff. to My User Manual)

22:15	Reviewing the finished parts	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none"> • Recheck written pages • Verified accuracy and adherence to requirements.
-------	------------------------------	------------------------------	---

✓ Action items

- @XUAN TUAN MINH NGUYEN Edit "Our Calendar" on the Homepage
- Ensure that all stated tasks are assigned to team members with specific deadlines.
Confirm that the project plan and deliverables are consistent with the criteria and submission requirements. @TRONG DAT HOANG
- @MAI AN NGUYEN Include a provision for team underperformance and a Confluence Working Agreement on the Homepage
- @MAI AN NGUYEN Write down tasks that need to be done (in Discussion topics)
- Complete their respective sections according to the project plan requirements and rubric. @Phuong Doanh HA
- Finalize personal profiles and differentiate them from "My User Manual" formats. @VNH LE
- Provide feedback for revisions if necessary before final submission. @Phuong Doanh HA

⌚ Decisions

- 👉 Finalize personal profiles and differentiate them from "My User Manual" formats.
- 👉 Each team member is in charge of finishing a certain portion of the project plan by the submission guidelines and rubric.
- 👉 Every participant will complete their profiles, making sure they differ from "My User Manual" templates.
- 👉 Before the final submission, quality assurance will confirm that all written pages are accurate and comply with the standards.

Meeting notes at 14/03/2024

📅 Date

14 Mar 2024

👥 Participants

- @XUAN TUAN MINH NGUYEN
- @MAI AN NGUYEN
- @Phuong Doanh HA
- @TRONG DAT HOANG
- @VINH LE

📋 Goals

- Fine-tuning the last two weeks tasks.
- Describing how Entity Relationship Diagram Works.
- Looking for ideas to draw a good Entity Relationship Diagram

🗣 Discussion topics

Time	Item	Presenter	Notes
15:00PM	Database Management Systems (DBMS)	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">• Provide a general view of how DBMS works• Giving the knowledge on Relation Structures and RDBMS options.• Learn database development lifecycle.
15:40PM	Preventing Data Anomalies	@MAI AN NGUYEN	<ul style="list-style-type: none">• Showing the bad aspects of how bad data, such as duplicate or inconsistent data, could affect the whole database system.• Giving solutions regarding prevent data conflicts.
16:15PM	Diagrams.net	Five Guys Group Chat (@XUAN TUAN MINH NGUYEN @MAI AN NGUYEN @Phuong Doanh HA @TRONG DAT HOANG @VINH LE)	<ul style="list-style-type: none">• Asking questions by team member regarding on the Database Management Systems (DBMS) and Preventing Data Anomalies that is shown by @XUAN TUAN MINH NGUYEN and @MAI AN NGUYEN .• Teaching member how to use Diagrams.net.• Thinking optimal strategies to optimize drawing diagrams.

✓ Action items

- Five Guys Group Chat (@XUAN TUAN MINH NGUYEN @MAI AN NGUYEN @Phuong Doanh HA @TRONG DAT HOANG @VINH LE) Self-learning Database Management Systems (DBMS) and Preventing Data Anomalies.
- Five Guys Group Chat (@XUAN TUAN MINH NGUYEN @MAI AN NGUYEN @Phuong Doanh HA @TRONG DAT HOANG @VINH LE) Brainstorming to find which strategies is optimize for drawing diagrams.
- @XUAN TUAN MINH NGUYEN @MAI AN NGUYEN Initialize the diagram and send it to Confluence.

⌚ Decisions

👉 Focusing on learning new knowledges

👉 Finding out solutions and start implementing the diagram

👉 Integrate it to Confluence

Meeting notes at 18/4/2024

Date

18 Apr 2024

Participants

- @TRONG DAT HOANG
- @XUAN TUAN MINH NGUYEN
- @Phuong Doanh HA
- @MAI AN NGUYEN
- @VINH LE

Goals

- Updating, reviewing, and evaluating the Project Progress.
- Updating Team Health Monitor for April.
- Review the current state of Confluence Work Space.

Discussion topics

Time	Item	Presenter	Notes
9.30 am	Updating, reviewing, and evaluating the Project Progress.	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">Discussed completed milestones, ongoing work, and anticipated bottlenecks.Agreed to reallocate resources to ensure timely completion.
10.00 am	Updating Team Health Monitor for April.	@TRONG DAT HOANG	<ul style="list-style-type: none">Reviewed team satisfaction survey outcomes; noticed improvements in communication but worries about task balance.Follow-up discussions have been scheduled with relevant team members.
11.00 am	Review the current state of Confluence Work Space.	@MAI AN NGUYEN	<ul style="list-style-type: none">Checked document updates, resolved access issues, and updated workspace structure for better navigation and collaboration.

Action items

- Set up regular check-ins to track work progress and address any issues that arise as soon as possible. @XUAN TUAN MINH NGUYEN
- Follow up on identified bottlenecks by assigning tasks to specific team members or departments. @XUAN TUAN MINH NGUYEN
- Schedule individual meetings with team members to look deeper into the issues mentioned during the Team Health Monitor session. @TRONG DAT HOANG @MAI AN NGUYEN

- Create action plans to alleviate workload imbalances and increase overall team satisfaction. @TRONG DAT HOANG
- Communicate any needed updates or modifications to the Confluence Work Space structure to the team. @MAI AN NGUYEN
- Help team members maximize their usage of Confluence for collaboration and document management. @MAI AN NGUYEN
@Phuong Doanh HA
- Provide feedback or suggestions to improve project progress, team dynamics, or workplace efficiency. @Phuong Doanh HA @VINH LE
- Respond promptly to any action items or assignments assigned during the meeting. @VINH LE

⌚ Decisions

👉 @XUAN TUAN MINH NGUYEN will reallocate resources to ensure timely completion.

👉 @TRONG DAT HOANG will consider workload balance based on team comments.

👉 @MAI AN NGUYEN IT has rectified Confluence difficulties and will provide training for optimal use.

Meeting notes at 11/04/2024

📅 Date

11 Apr 2024

👥 Participants

- @XUAN TUAN MINH NGUYEN
- @TRONG DAT HOANG
- @MAI AN NGUYEN
- @Phuong Doanh HA
- @VINH LE

📋 Goals

- Revise on how to use phpMyAdmin for interacting with MySQL.
- Access the MySQL database on Swinburne's Mercury Server.
- Revise the syntax of SQL query language and essentials commands for MySQL.
- Convert the Logical Model to a Relational Model

🗣 Discussion topics

Time	Item	Presenter	Notes
15:00PM	Harness the power of MySQL A brief overview of SQL Finding databases, tables and columns	@MAI AN NGUYEN	<ul style="list-style-type: none">• Introduce and give an overview on SQL, give an overview on few useful syntax of MySQL.• Give a short demonstration on how to use MySQL and phpMyAdmin to integrate with MySQL.
15:20PM	What are data types? The CREATE TABLE statement	@Phuong Doanh HA @TRONG DAT HOANG	<ul style="list-style-type: none">• Give a closer look on the data types that is mostly used in MySQL (Numeric, Date-time, String).• Introduce the CREATE TABLE statement with foreign keys, foreign key constraints and how to edit the foreign keys (Deleting and changing).• Demonstrate how to use CREATE TABLE to create a table.
15:50PM	Confluence, Jira	@XUAN TUAN MINH	<ul style="list-style-type: none">• Updating on the works regarding changing the current database into the highest level of normalization (3NF).

		NGUYEN @VINH LE	<ul style="list-style-type: none"> • Proposing an update on the progress of the group based on the works on Jira.
16:00PM		@XUAN TUAN MINH NGUYEN @VINH LE @MAI AN NGUYEN @TRONG DAT HOANG @Phuong Doanh HA	<ul style="list-style-type: none"> • Discussing the plan to create the database. • Initialize the database and submit it to GitHub for group collaboration.

✓ Action items

- @XUAN TUAN MINH NGUYEN @MAI AN NGUYEN @TRONG DAT HOANG @VINH LE @Phuong Doanh HA Revise on the knowledges regarding MySQL and phpMyAdmin.
- @TRONG DAT HOANG Convert the current Model into Relational Model.
- @Phuong Doanh HA @MAI AN NGUYEN Design the database and append it into the Feenix server.
- @VINH LE @XUAN TUAN MINH NGUYEN Add the designed database to GitHub.

⌚ Decisions

👉 Implementing based on the action items.

👉 Fast-forward to next week task for pushing up the progress.

Meeting notes at 29/03/2024

📅 Date

29 Mar 2024

👥 Participants

- @XUAN TUAN MINH NGUYEN
- @VINH LE
- @TRONG DAT HOANG
- @Phuong Doanh HA
- @MAI AN NGUYEN

📋 Goals

- Continue doing Requirements Documentation
- Start doing Personas and Empathy Mapping
- Start doing Product Requirements and ER Diagram
- Review the pages for Project Proposal

🗣 Discussion topics

Time	Item	Presenter	Notes
10:30	Review the Project Proposal, listing tasks to be finished off	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">• Tasks: Personas and EM, Product Requirements, ER diagram, complete Project Proposal (team) and Requirement Documentations (individual)
10:45	Allocating tasks	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">• Mentioned below
10:45	Requirements Documentation (Group discussion)	Five Guys Group Chat (@MAI AN NGUYEN @Phuong Doanh HA @TRONG DAT HOANG @VINH LE @XUAN TUAN MINH NGUYEN)	<ul style="list-style-type: none">• Discuss the potential answers to the activities

Action items

- @everyone Continue doing the Requirements Documentation, move My Usual Manual to the group's workspace
- Do ER Diagram [@XUAN TUAN MINH NGUYEN](#)
- Do Product Requirements [@Phuong Doanh HA](#)
- Indicate Team Health Monitor [@TRONG DAT HOANG](#)
- Do Personas and Empathy Map [@MAI AN NGUYEN](#)
- Complete Project Plan [@VINH LE](#) [@Phuong Doanh HA](#)
- Complete Risk Assessment Matrix [@TRONG DAT HOANG](#)
- recheck every page [@XUAN TUAN MINH NGUYEN](#)

Decisions

-  Each member will review and complete their assigned sections by the next meeting.
-  Shared workspace will be used for collaborative editing.
-  The team will study and provide input on the project proposal draft by the next meeting.
-  Final draft must be finished and submitted before the end of the month.

Meeting notes at 04/4/2024

Date

4 Apr 2024

Participants

- @Phuong Doanh HA
- @XUAN TUAN MINH NGUYEN
- @TRONG DAT HOANG
- @VINH LE
- @MAI AN NGUYEN

Goals

- Create and Put into Practice a Database Naming Convention
- Limitations: A Way to Verify and Improve Data Integrity
- Control Many-to-Many Connections in the Database
- Assess and Make Sense of It Normalisation of Databases

Discussion topics

Time	Item	Presenter	Notes
19:00	Database Naming Rules	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">• Establish a standard naming scheme for all database elements in order to foster consistency and comprehension within the development team.
19:20	Resolve Many-to-Many	@MAI AN NGUYEN	<ul style="list-style-type: none">• Assess if the database structure contains many-to-many relationships and discuss the effective management of these relationships with junction tables.
20:00	Assess Normalisation	@VINH LE	<ul style="list-style-type: none">• Analyse database architecture for 3NF compliance and talk about the advantages of denormalization for ease of use or speed.
20:20	Product Requirement	@TRONG DAT HOANG @Phuong Doanh HA	<ul style="list-style-type: none">• Review core product functionalities and discuss new features based on stakeholder feedback.• Confirm priorities, finalize requirement decisions, and outline next steps for development integration.

Action items

- @Phuong Doanh HA @TRONG DAT HOANG Determine which database constraints—such as NOT NULL, UNIQUE, CHECK, or FOREIGN KEY (FK)—are required for each user narrative by going over each one. After adding these limitations, update the database schema and update Confluence to reflect the changes.

- @XUAN TUAN MINH NGUYEN Create a prefix- or suffix-based database name convention, follow it consistently, and record it in Confluence for team reference.
- @MAI AN NGUYEN Update the ERD, report the modifications in Confluence, and use junction tables to find and fix many-to-many relationships in the database model.
- @VINH LE Determine whether denormalization is necessary based on needs, evaluate the database for 3NF compliance, and record changes in Confluence.

⌚ Decisions

- 👉 Examine user stories and add any database restrictions that are required.
- 👉 In Confluence, create and record a uniform naming scheme.
- 👉 Junction tables for many-to-many relationships should be updated in the ERD.
- 👉 Check the database for 3NF compliance, then use Confluence to document the denormalization process.

Meeting notes at 26/4/2024

Date

26 Apr 2024

Participants

- @XUAN TUAN MINH NGUYEN
- @VINH LE
- @TRONG DAT HOANG
- @Phuong Doanh HA
- @MAI AN NGUYEN

Goals

- Discuss about Use Cases (Individual assignment)
- Work on use cases for the project database
- Document them on Confluence

Discussion topics

Time	Item	Presenter	Notes
10:30	Use Cases (tutorial week 8)	Five Guys Group Chat (@MAI AN NGUYEN @Phuong Doanh HA @TRONG DAT HOANG @VINH LE @XUAN TUAN MINH NGUYEN)	<ul style="list-style-type: none">• Discuss each questions and their potential answers
11:00	Use cases of the project database	@XUAN TUAN MINH NGUYEN @TRONG DAT HOANG @Phuong Doanh HA @VINH LE	<ul style="list-style-type: none">• For each use case:<ul style="list-style-type: none">◦ Describe the scenario◦ Decide what SQL statements are needed to achieve this◦ Discuss the need for a transaction• Document them all in Confluence
11:30	Update the backlog in Jira	@MAI AN NGUYEN	<ul style="list-style-type: none">• lists the tasks on Jira• check if tasks have been done

Action items

- @XUAN TUAN MINH NGUYEN List out the use cases, lead the discussion
- @XUAN TUAN MINH NGUYEN @VINH LE @Phuong Doanh HA @TRONG DAT HOANG @MAI AN NGUYEN For each use case: describe the scenario, decide what SQL statements used to achieve the use case, discuss the need for a transaction
- @MAI AN NGUYEN Update the backlog in Jira

Decisions

 Each participant will work on developing comprehensive scenarios for their assigned use cases.

 SQL statements specific to each use case will be identified and documented.

 The Confluence documentation will cover the specifics of each transaction.

Meeting notes at 2/5/2024

📅 Date

2 May 2024

👥 Participants

- @XUAN TUAN MINH NGUYEN
- @VINH LE
- @TRONG DAT HOANG
- @Phuong Doanh HA
- @MAI AN NGUYEN

📋 Goals

- Discuss about tutorial week 9 Indexes
- Review the use cases and add more information
- Discuss the need for indexing
- Document the use case and the need for indexing in Confluence

🗣 Discussion topics

Time	Item	Presenter	Notes
20:00	Discuss about Indexes (tutorial week 9)	Five Guys Group Chat (@MAI AN NGUYEN @Phuong Doanh HA @TRONG DAT HOANG @VINH LE @XUAN TUAN MINH NGUYEN)	<ul style="list-style-type: none">• Discuss about the activities for tutorial 9
20:30	Review and edit the use cases	@XUAN TUAN MINH NGUYEN @VINH LE	<ul style="list-style-type: none">• Review the use cases regarding speed• Try to use the index to speed up the search
21:00	Discuss the need for indexing	@TRONG DAT HOANG @Phuong Doanh HA	<ul style="list-style-type: none">• Index where there could be potential to speed up the search for data in the database• Investigate how indexes are used in the database• Remove the index if it is not useful

21:15	Update the Jira backlog	@MAI AN NGUYEN	<ul style="list-style-type: none"> • List the tasks in detail and check if they have been done
-------	-------------------------	----------------	---

✓ Action items

- Five Guys Group Chat (@MAI AN NGUYEN @Phuong Doanh HA @TRONG DAT HOANG @VINH LE @XUAN TUAN MINH NGUYEN) Discuss about the tutorial week 9, the use cases and the need for indexing
- Document discussed topics in Confluence
 - @MAI AN NGUYEN Review the use cases from the POV of speed
 - @Phuong Doanh HA @XUAN TUAN MINH NGUYEN Potential attributes that might need indexing and explanation
- @VINH LE @TRONG DAT HOANG Run the SQL statements and investigate whether the DBMS chooses to use the index and document them in Confluence
- @MAI AN NGUYEN Update Jira backlog

⌚ Decisions

- 👉 All conversations about tutorial activities, use cases, and indexing requirements will be published in Confluence.
- 👉 Indexing requirements were addressed and defined for qualities that could potentially accelerate data retrieval.
- 👉 Redundant or unnecessary indices will be eliminated following performance testing.

Meeting notes at 9/5/2024

Date

9 May 2024

Participants

- @TRONG DAT HOANG
- @MAI AN NGUYEN
- @XUAN TUAN MINH NGUYEN
- @Phuong Doanh HA
- @VINH LE

Goals

- Specialized enhancements of database solutions;
- Project/team management.

Discussion topics

Time	Item	Presenter	Notes
11:00	Review of current database enhancement s	@XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">• Overview of ongoing enhancements, including performance improvements, security updates, and user interface refinements.
11:30	New feature plan for database solutions	@TRONG DAT HOANG	Presentation of proposed new features such as advanced search capabilities.
12:30	Team performance and feedback	@MAI AN NGUYEN @Phuong Doanh HA	Review of team performance metrics, feedback from team members, and strategies for improving collaboration and productivity.
1:30	Future project roadmap	@VINH LE	Outline of upcoming projects, key milestones, and timelines. Discussion on resource allocation and potential challenges.

✓ Action items

- ✓ @XUAN TUAN MINH NGUYEN Finalize performance improvements by next week.
- ✓ @TRONG DAT HOANG Create a detailed proposal for new features, including cost estimates and implementation timelines.
- ✓ @MAI AN NGUYEN Compile input from the team and offer an action plan to address problems.
- ✓ @Phuong Doanh HA Set up demos of the new tools and collect feedback from team members.

- @VINH LE Create a thorough roadmap for the upcoming quarter and share it with the team for review.

⌚ Decisions

- 👉 Approved the continuation of current database enhancement efforts.
- 👉 Agreed to proceed with the suggested new features pending cost analysis.
- 👉 Finalised the project roadmap for the coming quarter, with a focus on achieving important milestones and addressing potential problems.

Meeting notes at 16/5/2024

📅 Date

16 May 2024

👥 Participants

- @TRONG DAT HOANG
- @Phuong Doanh HA
- @XUAN TUAN MINH NGUYEN
- @VINH LE
- @MAI AN NGUYEN

📋 Goals

- Finalizing work on the project, documentation, and presentation video.
- Major-specific additions in project repository.

🗣 Discussion topics

Time	Item	Presenter	Notes
10:30	Finalizing project work	@XUAN TUAN MINH NGUYEN	Review the remaining project completion activities, which include coding, testing, and debugging.
11:30	Presentation video planning	@MAI AN NGUYEN	Planning the presentation video's content and structure, such as essential points to highlight, images to incorporate, and script authoring.
12:00	Major-specific additions in project repository	@TRONG DAT HOANG	An overview of new features and improvements introduced to the project repository, focusing on changes related to the project's major.
12:30	Review and feedback on additions	@VINH LE	Collecting feedback on current additions and discussing potential future improvements.
13:00	Next steps and deadlines	@Phuong Doanh HA	Setting timeframes for final tasks, documentation, and video creation.

✓ Action items

- @XUAN TUAN MINH NGUYEN Make sure all remaining project duties are finished by the end of the week.
- @MAI AN NGUYEN Draft the first edition of the project documentation and distribute it for review.
- @TRONG DAT HOANG Make a storyboard for the presentation video and assign scriptwriting and production responsibilities.

- @VINH LE Compile feedback on the additions and propose further enhancements based on the feedback.
- @Phuong Doanh HA Create a timeline for the final tasks, documentation, and video creation, and track progress.

⌚ Decisions

👉 We agreed to focus on completing the remaining project duties by the end of the week.

👉 The structure and content of the project documentation have been decided, and drafting duties have been assigned.

Meeting notes at 20/05/2024

Date

20 May 2024

Participants

- @XUAN TUAN MINH NGUYEN
- @Phuong Doanh HA
- @MAI AN NGUYEN
- @VINH LE
- @TRONG DAT HOANG

Goals

- Finalising work on project, documentation and presentation video.

Discussion topics

Time	Item	Presenter	Notes
			•

Action items

Decisions





Five Guys Project Plan

ADriver	@Phuong Doanh HA
Approver	@XUAN TUAN MINH NGUYEN
Contributors	@TRONG DAT HOANG @MAI AN NGUYEN @VINH LE
Informed	Archers and the archery club recorder
Objective	Create a scoring database system that allows accurate tracking of scores, looking up definitions, informations, results, etc.
Due date	Expect: 20/05/2024 (Week12)
Key outcomes	<ul style="list-style-type: none"> Accurate tracking of scores for archers and club recorder. Ability to look up the scores with restrictions and filters (date range, type of round, etc.); to look up competition, championship results, ranking, etc. Ability to enter scores with all the relevant data (date, time, round, equipment). Ability to add new archers, rounds, competition, scores, identify archer's division.
Status	IN PROGRESS

🧐 Problem Statement

Five Guys aims to provide a user-friendly, effective, comprehensive database system to track archery scores for both the individual archers and for club competitions/championships, with multiple functionalities such as looking up data records, definitions, requirements, and appending records.

🎯 Scope

Must have:	<ul style="list-style-type: none"> High-accuracy data recording. Effective lookup functionalities. User-friendly interface for end users (i.e. archers and club recorder).
-------------------	---

▶ Milestones and deadlines

Milestone	Owner	Deadline	Status
Project Report	@XUAN TUAN MINH NGUYEN @TRONG DAT HOANG @Phuong Doanh HA @MAI AN NGUYEN @VNH LE	27 May 2024	IN PROGRESS
Database design	@XUAN TUAN MINH NGUYEN @TRONG DAT HOANG @VNH LE	24 Mar 2024	DRAFT COMPLETE
UI Design	@Phuong Doanh HA @MAI AN NGUYEN	27 Apr 2024	IN PROGRESS
Backend design	@XUAN TUAN MINH NGUYEN @TRONG DAT HOANG @VNH LE	27 Apr 2024	IN PROGRESS

🔗 Reference materials

No reference materials is used at this point

Roles and Responsibilities

📋 Overview

Identify and discuss team responsibilities by following the instructions for the **Roles and Responsibilities Play**.

Team	Five Guys
Team members	@XUAN TUAN MINH NGUYEN @VINH LE @TRONG DAT HOANG @Phuong Doanh HA @MAI AN NGUYEN
Date	7 Mar 2024
Team mission	<ul style="list-style-type: none">• Develop a scoring database for the Archery club• Take a project-based approach to the project• Apply design thinking to the data-focused project

📘 Roles and responsibilities

Roles	Responsibilities (what others think)	Responsibilities (what I think)	Comments
Team leader @XUAN TUAN MINH NGUYEN	<ul style="list-style-type: none">• Set goals and objectives• Plan, organize, allocate tasks• Finish assigned tasks before deadlines• Monitor the progress• Lead team meetings• Support other members• Give feedbacks	<ul style="list-style-type: none">• Organize works, including dividing and planning the works, assigning tasks to team members, setting deadlines for the tasks• Manage project progress• Communicate with team members to clarify objectives, assign responsibilities, settle disputes, provide coaching comments, and inspire the group to reach its targets• Decider for main decisions in problem-solving• Report progress to stakeholders, fight for what the team needs	
Team members @VINH LE @TRONG DAT HOANG @Phuong Doanh HA @MAI AN NGUYEN	<ul style="list-style-type: none">• Contribute to the team's goal, ideas• Finish assigned tasks before deadlines• Communicate with the team, work together to solve bugs• Support other members	<ul style="list-style-type: none">• Execute all tasks assigned on schedule to the highest standard• Collaborate with other team members to achieve team goals• Participating in meetings	

	<ul style="list-style-type: none"> • Give feedbacks 		
Technical manager @TRONG DAT HOANG	<ul style="list-style-type: none"> • Design, code, manage the website • Fix bugs • Inform team leader of any problems found and work together with the team to find the solution 	<ul style="list-style-type: none"> • Main builders of the website: front-end and back-end • Monitor the webpages • Look for and fix bugs • Report issues regarding the websites to the Team leader, communicate with the team to solve the issue 	sub-role of Team members
Data designer @VINH LE	<ul style="list-style-type: none"> • Design the database • Make sure the database fits the stakeholder's requirements and needs 	<ul style="list-style-type: none"> • Build and design database for the project • Control and manage the database, looking for bugs • Improve the database • Report Team Leader, communicate with the team to resolve issues regarding the database 	sub-role of Team members
UI designer @Phuong Doanh HA	<ul style="list-style-type: none"> • Design the overall visual of the webpages • Design interface elements, focusing on style 	<ul style="list-style-type: none"> • Create visual designs for overall and each elements in webpages • Make sure webpages are consistent in design, including color, style, typography, and so on • Take in consider of the company's values to incorporate styles into the website • Work with other programmers to ensure the designs are implemented correctly 	sub-role of Team members
Resources managing and Researcher @MAI AN NGUYEN	<ul style="list-style-type: none"> • Do research on the project and related projects for references • Analyse the project to determine best practices 	<ul style="list-style-type: none"> • Gather information and data for problem-solving • Analyse information based on what found • Report team leader and communicate the work with the team for decision making 	sub-role of Team members

Working agreement

Team Preferences

Team Member					
@XUAN TUAN MINH NGUYEN Leader	@VINH LE Member	@TRONG DAT HOANG Member	@Phuong Doanh HA Member	@MAI AN NGUYEN Member	
Working location and timezone	Melbourne, AEDT (GMT +11)	Melbourne, AEDT (GMT +11)	Melbourne, AEDT (GMT +11)	Melbourne, AEDT (GMT +11)	Melbourne, AEDT (GMT +11)
Working hours and commitments	Everyday except Tuesday, Thursday and Saturday.	Evening and Tuesday, Wednesday, and the weekends	Morning and Midnight hours, starting 9am and 9pm	9:00 - 12:00; 19:00 - 00:00	14:00 - 18:00; 20:00 - 00:00
Working environment and preferences	An energetic, chill and relax environment. Especially I would really love to work in a stress-free and zero-arguments environment.	Chill, stress-free environment with room for creativity.	Flexibility and Collaborative workspace, where I can better balance professional and home life.	Flexible environment: my workspace is customized to my requirements, allowing me to effortlessly transition between quiet spaces for focused work and communal spaces for ideation sessions.	Quiet spaces to study alone and public study spaces for group studying; block out Sundays for other activities
How I like receiving feedback	I would prefer getting feedback from every individuals and on-premise since I want to know and fix the problem on the feedback as	Direct feedback from individuals or from the whole team. I'm capable of receiving feedback in the purest way.	Regular feedback sessions, such as one-on-one or team meetings.	One-in-one discussion Written feedback via message Feedback during performance reviews	Written feedback via Canvas and straightforward, face to face (or via messaging apps) discussion

	soon as possible.				
Context about me	9 hours per day still not enough for sleeping loh.	I'm lactose intolerant. I sleep early from 11.	Kind of introvert but anyway I am happy when you come and talk to me ^^.	Slow and steady wins the race.	I'm a night owl, but sometimes not.

💬 Communication Channels

Channel	Purpose	Audience	Standards
Confluence	Documentation and Knowledge Management, Project Management Provide centralised resources for essential information	Project team	Regular responses
Jira	Project Management: <ul style="list-style-type: none">Issue tracking and managementWorkflow customisationTracking performanceTask allocation	Project team	Expectations on responses
Messenger chat	Daily work-related conversations, announcements	Project team	Open on default
Discord	Online team meetings	Project team	Open on default
Email	Formal communication with the team Communicating with stakeholders/clients	Stakeholders, clients, project team	Expectations on responses

📅 Meetings

Objective	Project team makes decision on the project	Project team solves Project bugs/do reflection
Outcomes	Decision is made	Bugs are solved

		Reflection is made for later improvements
Format	Face-to-face meetings, Online meetings	Face-to-face meetings, Online meetings
	Whole team	Whole team
Who	Project team	Project team
Resources	Laptop, Confluence, Jira, (Discord if meeting online)	Laptop, Confluence, Jira, (Discord if meeting online)
How will we show up?	Respect, with concentration, collaborative, agendas	Respect, with concentration, agendas
How will we manage follow up?	Record on meeting notes, allocating tasks on Jira, take actions	Record on meeting notes, take actions, resolve tasks on Jira

⬆ Escalation Process

Decider	How	Transparency	Feedback Loop
@XUAN TUAN MINH NGUYEN	Discuss with team members via team meetings	The decision process will be shared with the whole team	💡 Share the decisions with clients/stakeholders via meetings and Jira for progress checking

💡 Continuous Improvement

Purpose	How	Standards
Reflection	<ul style="list-style-type: none"> • Face-to-face/online meetings • Messaging • Self-reflect 	<ul style="list-style-type: none"> • Documented on meeting notes
Celebrating success	Face-to-face	Rarely/at the end of the project

Team Member Profiles

Member	Profiles
@XUAN TUAN MINH NGUYEN	 My User Manual: Xuan Tuan Minh Nguyen
@VINH LE	 My User Manual: Phuc Vinh Le
@TRONG DAT HOANG	 My User Manual: Trong Dat Hoang
@Phuong Doanh HA	 My User Manual: Phuong Doanh Ha
@MAI AN NGUYEN	 My User Manual: Mai An Nguyen

My User Manual: Xuan Tuan Minh Nguyen

Environments I like to work in	<ul style="list-style-type: none"> An energetic, chill and relax environment. Especially I would really love to work in a stress-free and zero-arguments environment. 	
Preferred working hours	<ul style="list-style-type: none"> Everyday except Tuesday, Thursday and Saturday. 	
Communication preferences	<ul style="list-style-type: none"> Online: Facebook, Instagram, Discord, Google Meets (Microsoft Teams), Whatsapp and any other methods. Oh I do accept offline meeting as well !!!! 	
Preferred ways to receive feedback	<ul style="list-style-type: none"> I would prefer getting feedback from every individuals and on-premise since I want to know and fix the problem on the feedback as soon as possible. 	
Things I need	A nice and friendly environment just like I have mentioned before in the first section, that's it !	
How I learn best	Self-learning and learning through failure.	
Things I struggle with	Easily to get panic when dealing with the problems.	
Things I love	Billiards and Coding, ofc !	
If I were an animated gif/meme/animal/song, I would be...	Rick Roll xDDDD.	
My favorite saying	"Death is nothing, but to live defeated and inglorious is to die daily" - Napoleon Bonaparte.	
Other things I want you to know about me	Just nothing if we are not close enough 😊	

My User Manual: Phuc Vinh Le

Environments I like to work in	<ul style="list-style-type: none">An environment that allows me to be creative, while also stress-free and chill.	
Preferred working hours	<ul style="list-style-type: none">Full day: Tuesday and Wednesday.Evening: Every other day.	
Communication preferences	<ul style="list-style-type: none">Discord, Messenger, Zoom, Teams, Meets, etc.	
Preferred ways to receive feedback	<ul style="list-style-type: none">Direct feedback from the team, or from separate individuals should be great for me, as I'm open for feedbacks.	
Things I need	<ul style="list-style-type: none">A nice environment as mentioned.Good music, good laptop.Great sleep.	
How I learn best	<ul style="list-style-type: none">Self-learning, with some occasional support from friends.	
Things I struggle with	<ul style="list-style-type: none">Time management.	
Things I love	<ul style="list-style-type: none">Music, F1, Gym, Basketball.	
If I were an animated gif/meme/animal/song, I would be...	<ul style="list-style-type: none">That One Sweaty Guy GIF	
My favorite saying	<ul style="list-style-type: none">It's hammer time - Bono	
Other things I want you to know about me	<ul style="list-style-type: none">I'm lactose intolerant.	

My User Manual: Trong Dat Hoang

Environments I like to work in	<ul style="list-style-type: none"> Flexibility and Collaborative workspace, where i can better balance professional and home life. 	
Preferred working hours	<ul style="list-style-type: none"> Morning and Midnight hours Starting 9am and 9pm 	
Communication preferences	<p>Clear communication: open and transparent communication.</p> <ul style="list-style-type: none"> Face-to-face meeting. Online meeting. 	
Preferred ways to receive feedback	<ul style="list-style-type: none"> Regular feedback sessions, such as one-on-one or team meetings. 	 <p>Copyright 2004 by Randy Glasbergen. www.glasbergen.com GLASBERGEN</p> <p>"It's not a great mission statement, but we'll revise it if things get better."</p>
Things I need	<ul style="list-style-type: none"> Competitive workspace where have opportunities for personal growth and development. Quiet place to focus on study, work ,work and work 	 <p>TOONSTOCK CartoonStock.com</p> <p>Damn you Smith, you competitive bastard...</p>
How I learn best	<ul style="list-style-type: none"> I learn best through active listening methods, in fact i often listen to recordings with headphones to immerse myself in the audio, or use background music to enhance focus 	 <p>Education corner www.edcorner.com</p> <p>Active Listening</p> <ul style="list-style-type: none"> Hearing Content Listening for Feeling Observing Body Language Neutral Technique Paraphrasing Self-Awareness Reflection Questioning Clarifying Technique Summarizing

Things I struggle with

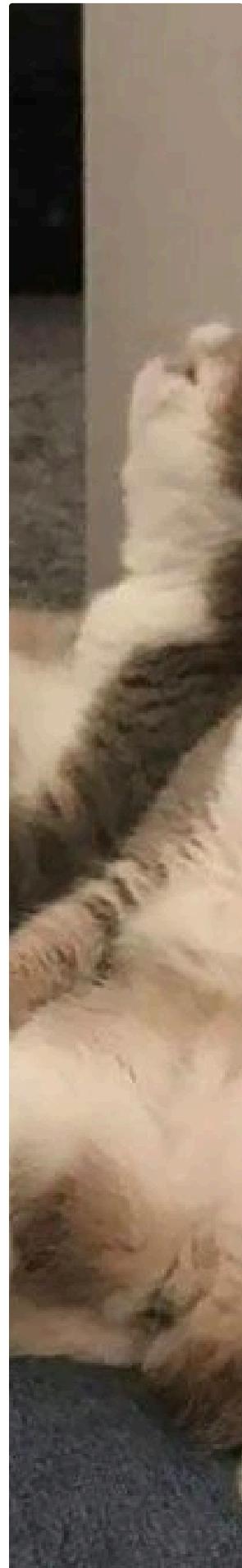
- Managing stress and maintaining work-life balance.
- I wish i could stop thinking to much and go to bed earlier.





Things I love

- SLEEP
- SLEEEP
- SLEEPPE
- Zzzzz... Its better raining outside ❤️

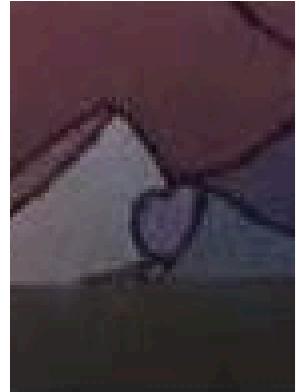




**If I were an animated
gif/meme/animal/song, I would
be...**

I would be the song "Birds" by Imagine Dragon that a sense of overcoming obstacles and spreading my wings to explore new horizons.



		
My favorite saying	<ul style="list-style-type: none"> The only thing we have to fear is fear itself. -Franklin D. Roosevelt 	
Other things I want you to know about me	<ul style="list-style-type: none"> I am kind of an introvert person but if you just come to me and say Hi!, we will have a nice conversation !!! 	

i For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

My User Manual: Phuong Doanh Ha

Environments I like to work in	<ul style="list-style-type: none">Flexible environment : my workspace is customized to my requirements, allowing me to effortlessly transition between quiet spaces for focused work and communal spaces for ideation sessions.
Preferred working hours	9:00 - 12:00; 19:00 - 00:00
Communication preferences	<ul style="list-style-type: none">Video callsFace-to-face meetingsDiscordConfluence
Preferred ways to receive feedback	<ul style="list-style-type: none">One-on-one discussionWritten feedback via messageFeedback during performance reviews
Things I need	<ul style="list-style-type: none">Professional growth.Supportive environment.Recognition & appreciation.Work-life balance & wellness.
How I learn best	Receiving enough input has made a huge contribution to my professional development and improvement at work.
Things I struggle with	<ul style="list-style-type: none">Maintaining focus and avoiding distractions.Overcoming perfectionism & failure.Balancing responsibilities.

i For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

My User Manual: Mai An Nguyen

Environments I like to work in	<ul style="list-style-type: none">• Quiet spaces to study alone• Public study spaces for group studying
Preferred working hours	<ul style="list-style-type: none">• 14:00 - 18:00; 20:00 - 00:00
Communication preferences	<ul style="list-style-type: none">• Verbal: face-to-face communication, online meetings• Written: chats via messaging apps
Preferred ways to receive feedback	<ul style="list-style-type: none">• Written feedback via Canvas• Any type of discussions (as long as they are direct and honest)
Things I need	<ul style="list-style-type: none">• Clear objectives• Defined roles and responsibilities• Regular Check-ins and Meetings• Flexibility and Adaptability
How I learn best	<ul style="list-style-type: none">• Association• Rewrite things down• Test myself• Mnemonics
Things I struggle with	<ul style="list-style-type: none">• Memorise info

 For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

Product requirements

Target release	Version 0.1 (22 May 2024)																																			
Epic	<table border="1"> <thead> <tr> <th>Type</th><th>Key</th><th>Summary</th><th>Assigned to</th></tr> </thead> <tbody> <tr> <td></td><td>FGPMS-44</td><td>Week 10/11 Workshop</td><td> XU</td></tr> <tr> <td></td><td>FGPMS-43</td><td>Week 9 Workshop</td><td> XU</td></tr> <tr> <td></td><td>FGPMS-42</td><td>Week 8 Workshop</td><td> XU</td></tr> <tr> <td></td><td>FGPMS-41</td><td>Week 7 Workshop</td><td> XU</td></tr> <tr> <td></td><td>FGPMS-40</td><td>Week 6 Workshop</td><td> XU</td></tr> <tr> <td></td><td>FGPMS-39</td><td>Week 5 Workshop</td><td> XU</td></tr> <tr> <td></td><td>FGPMS-22</td><td>Reflection and Peer Assessment</td><td> XU</td></tr> </tbody> </table>				Type	Key	Summary	Assigned to		FGPMS-44	Week 10/11 Workshop	XU		FGPMS-43	Week 9 Workshop	XU		FGPMS-42	Week 8 Workshop	XU		FGPMS-41	Week 7 Workshop	XU		FGPMS-40	Week 6 Workshop	XU		FGPMS-39	Week 5 Workshop	XU		FGPMS-22	Reflection and Peer Assessment	XU
Type	Key	Summary	Assigned to																																	
	FGPMS-44	Week 10/11 Workshop	XU																																	
	FGPMS-43	Week 9 Workshop	XU																																	
	FGPMS-42	Week 8 Workshop	XU																																	
	FGPMS-41	Week 7 Workshop	XU																																	
	FGPMS-40	Week 6 Workshop	XU																																	
	FGPMS-39	Week 5 Workshop	XU																																	
	FGPMS-22	Reflection and Peer Assessment	XU																																	
	<p>15 items</p>																																			
	<p>Synced just now </p>																																			
Document status	DONE																																			
Document owner	@XUAN TUAN MINH NGUYEN																																			
Designer	@MAI AN NGUYEN																																			
Tech lead	@XUAN TUAN MINH NGUYEN																																			
Technical writers	@TRONG DAT HOANG @VINH LE @Phuong Doanh HA																																			
QA																																				

🎯 Objective

- Provide an advanced and user-friendly UI system for the complete administration and on-the-spot tracking of archery scores and competition information. With the ability to see and update archers' scores and competition-related variables in detail, this system will improve user experience by managing and delivering information efficiently.

📊 Success metrics

Goal	Metric
User satisfaction <ul style="list-style-type: none"> Achieve 90% or more in the number of visits from archers (visitors) Have the scoreboard updated regularly from the recorder due to ease of use. 	<ul style="list-style-type: none"> Access rate (commitment rate) Survey index Latest update table completion time
Accuracy, Data Accessibility and Management <ul style="list-style-type: none"> 100% accurate score table, updated over time (From when archers started participating to the present) + sorted by time New competitions are updated immediately after the competition ends. Ability to store and display the highest score of each archer, per round, from the competition, based on search requests. 	<ul style="list-style-type: none"> Based on system logs and reports Track update announcement dates, start and completion dates of system modifications, and deployment dates. Track how often input errors are modified (edited by users)

💡 Assumptions

- Assumption of **mobile compatibility**:

We anticipated that the system would perform admirably on portable devices with a usage rate of at least 95%. Its perfect performance across several mobile devices ensures that its user interface is both inviting and responsive. They also held the opinion that all of the archers' scoring devices are always connected to the internet, enabling data entry and synchronization in real time.

- Assumption regarding **Database and Server Infrastructure**:

To avoid data intrusions, the server and database are housed in a safe environment with robust security mechanisms. Their robust processing power and ample storage allow them to easily handle complex calculations and enormous volumes of updated data. We'll make an effort to keep things running smoothly even during times of high usage.

📋 Requirements

Stakeholder	Requirement	User Story	Importance	Issue	Notes
Archer	Score Entry and Update	The archer wants to enter scores via a handheld device immediately after the rounds just end to ensure accuracy and timeliness.	HIGH	Delays in data synchronization and problems with mobile compatibility	Make sure the mobile platform is well-supported.
Archer	Retrieve and Sort Past Results	To monitor progress in performance, an archer must filter and sort previous results according to round type and date.	HIGH	Problems with massive dataset performance and intricate filter logic	Optimize for user-friendliness and rapid access
Archer	Round Definitions and	The requirement for thorough definitions	HIGH	Frequent updates are	Regular changes to

	Details (Including Equivalent Rounds)	of each round, including equivalent rounds based on current standards.		required owing to regulatory changes and to ensure data accuracy. (Business rules)	ensure correctness.
Recorder	Validation of historical data and past competition.	The recorder necessitates that the system keep track of previous scores and competitions, particularly when round definitions change.	HIGH	Challenges in maintaining historical integrity and data consistency notwithstanding rule modifications.	The system should facilitate historical data auditing.
Recorder	Recorders have detailed score recording capabilities.	To ensure proper records, the recorder must precisely submit detailed scores, including per-arrow and end statistics.	HIGH	Input form complexity leads to user mistake when entering scores.	User interface simplification and training are required.

🎨 User interaction and design

1. **Entire Competition Management:**

Combine notifications and scheduling for events, automatically declare winners and participant rankings, and exhibit competition formats and round specifics with clarity.

2. **Real-time Analysis and Scoring:**

Provide a real-time scoring system for archery that includes features for in-depth score breakdowns by round, equipment, and range in addition to access to previous data for trend analysis.

3. **Data Visualisation:**

Charts and Graphs are a must to help both users / managers tracking their metrics regarding their performance evaluations, score distributions, and statistical examinations.

❓ Open Questions

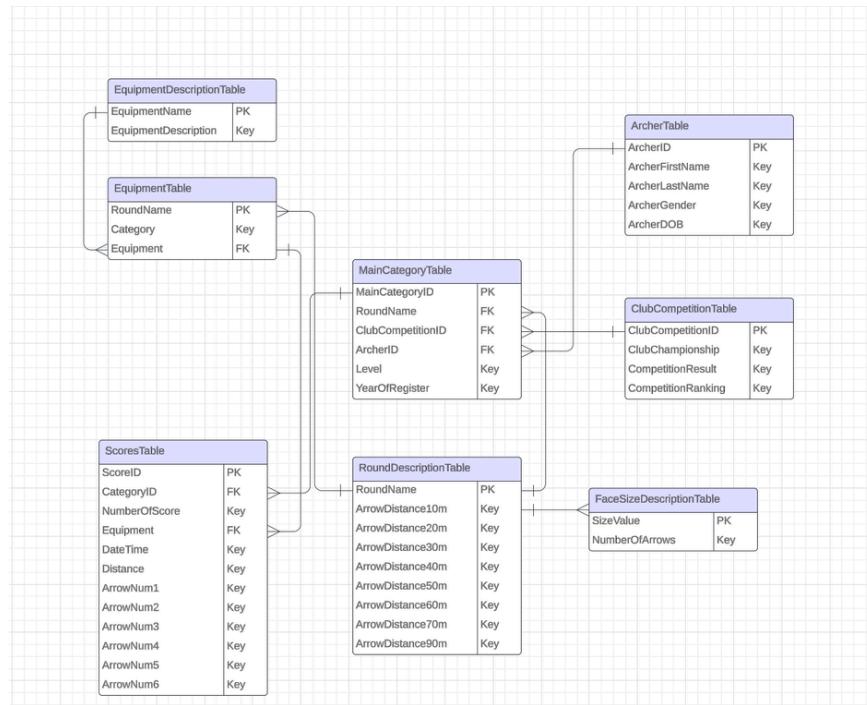
Question	Answer	Date Answered

⚠️ Out of Scope

- **Multi-Sport Management:** Sports management solutions that handle several sports will not be compatible with the system.
- **Social Media Updates:** There is no automatic sharing of scores on social media.
- **Live streaming of video and connection:** to streaming services are not supported by the system.
- **Selling products:** Using the system to directly sell goods or tickets is not possible.

Entity Relationship Diagram

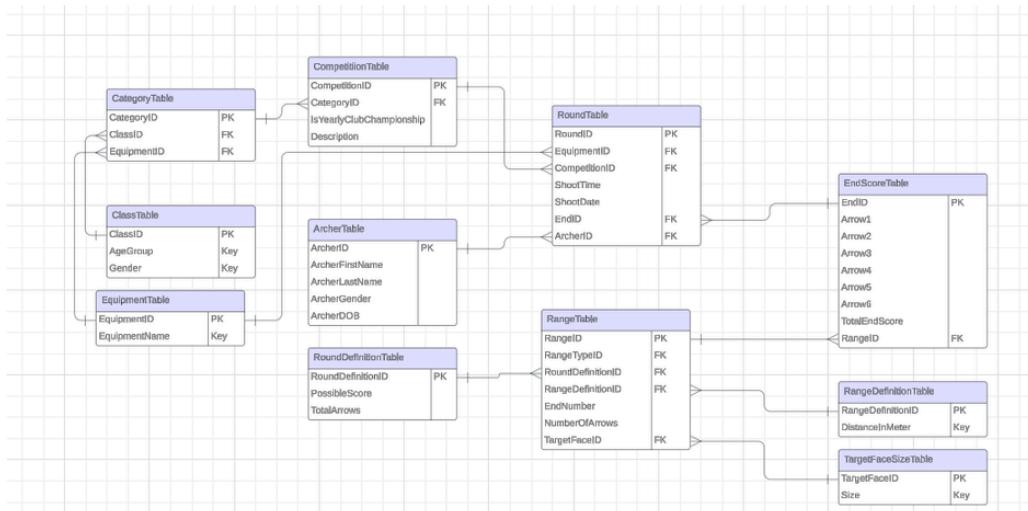
Initial



Update

- 1st draft ERD: Most of the tables are not in 3NF yet. Some are still in 1NF.
- For the final ERD:
 - Check for Naming Convention: choose Pascalcase
 - Check for relationships between fields in tables: all relationships are one-to-many
 - Normalise the database

Reviews



Physical database - create table statements

```
1 -- MySQL dump 10.13 Distrib 8.0.37, for macos14 (arm64)
2 --
3 -- Host: localhost      Database: cos20031_5_db
4 --
5 -- Server version     8.0.37
6
7 /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8 /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9 /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!50503 SET NAMES utf8mb4 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17
18 --
19 -- Table structure for table `ArcherTable`
20 --
21
22 DROP TABLE IF EXISTS `ArcherTable`;
23 /*!40101 SET @saved_cs_client      = @@character_set_client */;
24 /*!50503 SET character_set_client = utf8mb4 */;
25 CREATE TABLE `ArcherTable` (
26   `ArcherID` int NOT NULL AUTO_INCREMENT,
27   `ArcherFirstName` varchar(255) NOT NULL,
28   `ArcherLastName` varchar(255) NOT NULL,
29   `ArcherGender` enum('Male','Female') NOT NULL,
30   `ArcherDOB` date NOT NULL,
31   PRIMARY KEY (`ArcherID`)
32 ) ENGINE=InnoDB AUTO_INCREMENT=546 DEFAULT CHARSET=latin1;
33 /*!40101 SET character_set_client = @saved_cs_client */;
34
35 --
36 -- Dumping data for table `ArcherTable`
37 --
38
39 /*!40000 ALTER TABLE `ArcherTable` DISABLE KEYS */;
40 INSERT INTO `ArcherTable` VALUES (1,'Kevin','James','Male','1980-05-14'),(2,'Gary','Piastri','Male','1983-07-21');
41 /*!40000 ALTER TABLE `ArcherTable` ENABLE KEYS */;
42
43 --
44 -- Table structure for table `CategoryTable`
45 --
46
47 DROP TABLE IF EXISTS `CategoryTable`;
48 /*!40101 SET @saved_cs_client      = @@character_set_client */;
49 /*!50503 SET character_set_client = utf8mb4 */;
50 CREATE TABLE `CategoryTable` (
51   `CategoryID` int NOT NULL,
52   `ClassID` int NOT NULL,
53   `EquipmentID` int NOT NULL,
```

```

54  PRIMARY KEY (`CategoryID`),
55  KEY `ClassID` (`ClassID`),
56  KEY `EquipmentID` (`EquipmentID`),
57  CONSTRAINT `CategoryTable_ibfk_1` FOREIGN KEY (`ClassID`) REFERENCES `ClassTable` (`ClassID`),
58  CONSTRAINT `CategoryTable_ibfk_2` FOREIGN KEY (`EquipmentID`) REFERENCES `EquipmentTable` (`EquipmentID`)
59 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
60 /*!40101 SET character_set_client = @saved_cs_client */;
61
62 --
63 -- Dumping data for table `CategoryTable`
64 --
65
66 /*!40000 ALTER TABLE `CategoryTable` DISABLE KEYS */;
67 INSERT INTO `CategoryTable` VALUES (1,2,5),(2,2,3),(3,2,4),(4,1,5),(5,1,3),(6,1,4),(7,4,5),(8,4,3),(9,4,4),(10,
68 /*!40000 ALTER TABLE `CategoryTable` ENABLE KEYS */;
69
70 --
71 -- Table structure for table `ClassTable`
72 --
73
74 DROP TABLE IF EXISTS `ClassTable`;
75 /*!40101 SET @saved_cs_client      = @@character_set_client */;
76 /*!50503 SET character_set_client = utf8mb4 */;
77 CREATE TABLE `ClassTable` (
78   `ClassID` int NOT NULL,
79   `AgeGroup` enum('Open','50+','60+','70+','Under 21','Under 18','Under 16','Under 14') NOT NULL,
80   `Gender` enum('Male','Female') NOT NULL,
81   PRIMARY KEY (`ClassID`)
82 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
83 /*!40101 SET character_set_client = @saved_cs_client */;
84
85 --
86 -- Dumping data for table `ClassTable`
87 --
88
89 /*!40000 ALTER TABLE `ClassTable` DISABLE KEYS */;
90 INSERT INTO `ClassTable` VALUES (1,'Open','Female'),(2,'Open','Male'),(3,'50+','Female'),(4,'50+','Male'),(5,'6
91 /*!40000 ALTER TABLE `ClassTable` ENABLE KEYS */;
92
93 --
94 -- Table structure for table `CompetitionTable`
95 --
96
97 DROP TABLE IF EXISTS `CompetitionTable`;
98 /*!40101 SET @saved_cs_client      = @@character_set_client */;
99 /*!50503 SET character_set_client = utf8mb4 */;
100 CREATE TABLE `CompetitionTable` (
101   `CompetitionID` int NOT NULL AUTO_INCREMENT,
102   `CategoryID` int DEFAULT NULL,
103   `YearlyClubChampionship` tinyint(1) NOT NULL,
104   `Description` varchar(255) NOT NULL,
105   PRIMARY KEY (`CompetitionID`),
106   KEY `CategoryID` (`CategoryID`),
107   CONSTRAINT `CompetitionTable_ibfk_1` FOREIGN KEY (`CategoryID`) REFERENCES `CategoryTable` (`CategoryID`)
108 ) ENGINE=InnoDB AUTO_INCREMENT=72 DEFAULT CHARSET=latin1;
109 /*!40101 SET character_set_client = @saved_cs_client */;
110
111 --

```

```

112 -- Dumping data for table `CompetitionTable`  

113 --  

114  

115 /*!40000 ALTER TABLE `CompetitionTable` DISABLE KEYS */;  

116 INSERT INTO `CompetitionTable` VALUES (37,NULL,0,'Practice'),(38,1,1,'Male Open Recurve Compound'),(39,2,0,'Male  

117 /*!40000 ALTER TABLE `CompetitionTable` ENABLE KEYS */;  

118  

119 --  

120 -- Table structure for table `EndScoreTable`  

121 --  

122  

123 DROP TABLE IF EXISTS `EndScoreTable`;  

124 /*!40101 SET @saved_cs_client      = @@character_set_client */;  

125 /*!50503 SET character_set_client = utf8mb4 */;  

126 CREATE TABLE `EndScoreTable` (  

127   `EndID` int NOT NULL AUTO_INCREMENT,  

128   `Arrow1` tinyint(1) DEFAULT NULL,  

129   `Arrow2` tinyint(1) DEFAULT NULL,  

130   `Arrow3` tinyint(1) DEFAULT NULL,  

131   `Arrow4` tinyint(1) DEFAULT NULL,  

132   `Arrow5` tinyint(1) DEFAULT NULL,  

133   `Arrow6` tinyint(1) DEFAULT NULL,  

134   `TotalEndScore` int NOT NULL,  

135   `RangeID` int NOT NULL,  

136   PRIMARY KEY (`EndID`),  

137   KEY `EndScoreTable_ibfk_1` (`RangeID`),  

138   CONSTRAINT `EndScoreTable_ibfk_1` FOREIGN KEY (`RangeID`) REFERENCES `RangeTable` (`RangeID`)  

139 ) ENGINE=InnoDB AUTO_INCREMENT=271 DEFAULT CHARSET=latin1;  

140 /*!40101 SET character_set_client = @saved_cs_client */;  

141  

142 --  

143 -- Dumping data for table `EndScoreTable`  

144 --  

145  

146 /*!40000 ALTER TABLE `EndScoreTable` DISABLE KEYS */;  

147 INSERT INTO `EndScoreTable` VALUES (1,4,3,2,2,2,5,18,1),(2,10,8,1,4,10,3,36,1),(3,7,8,6,5,2,2,30,1),(4,1,7,7,7,  

148 /*!40000 ALTER TABLE `EndScoreTable` ENABLE KEYS */;  

149  

150 --  

151 -- Table structure for table `EquipmentTable`  

152 --  

153  

154 DROP TABLE IF EXISTS `EquipmentTable`;  

155 /*!40101 SET @saved_cs_client      = @@character_set_client */;  

156 /*!50503 SET character_set_client = utf8mb4 */;  

157 CREATE TABLE `EquipmentTable` (  

158   `EquipmentID` int NOT NULL,  

159   `EquipmentName` enum('R','C','B','L','RC','RCB','RCBL','BL') NOT NULL,  

160   PRIMARY KEY (`EquipmentID`)  

161 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  

162 /*!40101 SET character_set_client = @saved_cs_client */;  

163  

164 --  

165 -- Dumping data for table `EquipmentTable`  

166 --  

167  

168 /*!40000 ALTER TABLE `EquipmentTable` DISABLE KEYS */;  

169 INSERT INTO `EquipmentTable` VALUES (1,'R'),(2,'C'),(3,'B'),(4,'L'),(5,'RC'),(6,'RCB'),(7,'RCBL'),(8,'BL');

```

```

170 /*!40000 ALTER TABLE `EquipmentTable` ENABLE KEYS */;
171
172 --
173 -- Table structure for table `RangeDefinitionTable` -
174 --
175
176 DROP TABLE IF EXISTS `RangeDefinitionTable`;
177 /*!40101 SET @saved_cs_client      = @@character_set_client */;
178 /*!50503 SET character_set_client = utf8mb4 */;
179 CREATE TABLE `RangeDefinitionTable` (
180   `RangeDefinitionID` int NOT NULL,
181   `DistanceInMeter` int NOT NULL,
182   PRIMARY KEY (`RangeDefinitionID`)
183 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
184 /*!40101 SET character_set_client = @saved_cs_client */;
185
186 --
187 -- Dumping data for table `RangeDefinitionTable` -
188 --
189
190 /*!40000 ALTER TABLE `RangeDefinitionTable` DISABLE KEYS */;
191 INSERT INTO `RangeDefinitionTable` VALUES (1,90),(2,70),(3,60),(4,50),(5,40),(6,30),(7,20),(8,10);
192 /*!40000 ALTER TABLE `RangeDefinitionTable` ENABLE KEYS */;
193
194 --
195 -- Table structure for table `RangeTable` -
196 --
197
198 DROP TABLE IF EXISTS `RangeTable`;
199 /*!40101 SET @saved_cs_client      = @@character_set_client */;
200 /*!50503 SET character_set_client = utf8mb4 */;
201 CREATE TABLE `RangeTable` (
202   `RangeID` int NOT NULL AUTO_INCREMENT,
203   `RangeDefinitionID` int NOT NULL,
204   `RoundDefinitionID` varchar(255) NOT NULL,
205   `EndNumber` int NOT NULL,
206   `NumberOfArrows` int NOT NULL,
207   `TargetFaceID` int NOT NULL,
208   `RangeTypeID` int NOT NULL,
209   PRIMARY KEY (`RangeID`),
210   KEY `RangeDefinitionID` (`RangeDefinitionID`),
211   KEY `RoundDefinitionID` (`RoundDefinitionID`),
212   KEY `fk_target_face_id` (`TargetFaceID`),
213   KEY `idx_RangeTypeID` (`RangeTypeID`),
214   CONSTRAINT `fk_target_face_id` FOREIGN KEY (`TargetFaceID`) REFERENCES `TargetFaceSizeTable` (`TargetFaceID`),
215   CONSTRAINT `RangeTable_ibfk_1` FOREIGN KEY (`RangeDefinitionID`) REFERENCES `RangeDefinitionTable` (`RangeDefinitionID`),
216   CONSTRAINT `RangeTable_ibfk_2` FOREIGN KEY (`RoundDefinitionID`) REFERENCES `RoundDefinitionTable` (`RoundDefinitionID`)
217 ) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=latin1;
218 /*!40101 SET character_set_client = @saved_cs_client */;
219
220 --
221 -- Dumping data for table `RangeTable` -
222 --
223
224 /*!40000 ALTER TABLE `RangeTable` DISABLE KEYS */;
225 INSERT INTO `RangeTable` VALUES (1,1,'WA90/1440',6,36,2,1),(2,2,'WA90/1440',6,36,2,1),(3,4,'WA90/1440',6,36,1,1
226 /*!40000 ALTER TABLE `RangeTable` ENABLE KEYS */;
227

```

```

228 --
229 -- Table structure for table `RoundDefinitionTable` 
230 --
231
232 DROP TABLE IF EXISTS `RoundDefinitionTable`;
233 /*!40101 SET @saved_cs_client      = @@character_set_client */;
234 /*!50503 SET character_set_client = utf8mb4 */;
235 CREATE TABLE `RoundDefinitionTable` (
236   `RoundDefinitionID` varchar(255) NOT NULL,
237   `PossibleScore` int NOT NULL,
238   `TotalArrows` int NOT NULL,
239   PRIMARY KEY (`RoundDefinitionID`)
240 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
241 /*!40101 SET character_set_client = @saved_cs_client */;
242 
243 --
244 -- Dumping data for table `RoundDefinitionTable`
245 --
246
247 /*!40000 ALTER TABLE `RoundDefinitionTable` DISABLE KEYS */;
248 INSERT INTO `RoundDefinitionTable` VALUES ('AA40/1440',1440,144),('AA50/1440',1440,144),('Adelaide',1200,120),(
249 /*!40000 ALTER TABLE `RoundDefinitionTable` ENABLE KEYS */;
250 
251 --
252 -- Table structure for table `RoundTable` 
253 --
254
255 DROP TABLE IF EXISTS `RoundTable`;
256 /*!40101 SET @saved_cs_client      = @@character_set_client */;
257 /*!50503 SET character_set_client = utf8mb4 */;
258 CREATE TABLE `RoundTable` (
259   `RoundID` int NOT NULL AUTO_INCREMENT,
260   `EquipmentID` int DEFAULT NULL,
261   `CompetitionID` int NOT NULL,
262   `EndID` int NOT NULL,
263   `ArcherID` int NOT NULL,
264   `ShootTime` time DEFAULT NULL,
265   `ShootDate` date DEFAULT NULL,
266   PRIMARY KEY (`RoundID`),
267   KEY `EquipmentID` (`EquipmentID`),
268   KEY `CompetitionID` (`CompetitionID`),
269   KEY `EndID` (`EndID`),
270   KEY `ArcherID` (`ArcherID`),
271   CONSTRAINT `RoundTable_ibfk_1` FOREIGN KEY (`EquipmentID`) REFERENCES `EquipmentTable`(`EquipmentID`),
272   CONSTRAINT `RoundTable_ibfk_2` FOREIGN KEY (`CompetitionID`) REFERENCES `CompetitionTable`(`CompetitionID`),
273   CONSTRAINT `RoundTable_ibfk_3` FOREIGN KEY (`EndID`) REFERENCES `EndScoreTable`(`EndID`),
274   CONSTRAINT `RoundTable_ibfk_4` FOREIGN KEY (`ArcherID`) REFERENCES `ArcherTable`(`ArcherID`)
275 ) ENGINE=InnoDB AUTO_INCREMENT=73 DEFAULT CHARSET=latin1;
276 /*!40101 SET character_set_client = @saved_cs_client */;
277 
278 --
279 -- Dumping data for table `RoundTable`
280 --
281
282 /*!40000 ALTER TABLE `RoundTable` DISABLE KEYS */;
283 INSERT INTO `RoundTable` VALUES (1,1,37,1,35,'14:39:00','2024-04-16'),(2,1,37,2,35,'14:47:00','2024-04-16'),(3,
284 /*!40000 ALTER TABLE `RoundTable` ENABLE KEYS */;
285

```

```

286 --
287 -- Table structure for table `TargetFaceSizeTable`
288 --
289
290 DROP TABLE IF EXISTS `TargetFaceSizeTable`;
291 /*!40101 SET @saved_cs_client      = @@character_set_client */;
292 /*!50503 SET character_set_client = utf8mb4 */;
293 CREATE TABLE `TargetFaceSizeTable` (
294     `TargetFaceID` int NOT NULL,
295     `TargetFaceSize` enum('80cm','122cm') NOT NULL,
296     PRIMARY KEY (`TargetFaceID`)
297 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
298 /*!40101 SET character_set_client = @saved_cs_client */;
299
300 --
301 -- Dumping data for table `TargetFaceSizeTable`
302 --
303
304 /*!40000 ALTER TABLE `TargetFaceSizeTable` DISABLE KEYS */;
305 INSERT INTO `TargetFaceSizeTable` VALUES (1,'80cm'),(2,'122cm');
306 /*!40000 ALTER TABLE `TargetFaceSizeTable` ENABLE KEYS */;
307 /*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
308
309 /*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
310 /*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
311 /*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
312 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
313 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
314 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
315 /*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
316
317 -- Dump completed on 2024-05-24 11:44:54
318

```

Documentation on data creation and null values

Nullable attributes

For activity 1, within the team, discuss which attributes in the project database should be nullable. Most of the attributes are nullable since those fields could lack information without affecting the project database. Here are a few nullable attributes the team has discussed about:

Attributes	Table	Data type	Explanation
CategoryID	CompetitionTable	Integer	if the value in this field is NULL, this indicates practice rounds, not any type of competition.

Data creation

For activities 2 and 3, the team needs to create and insert data into each table. We have inserted them using SQL code files and uploaded them into the project database in our team's phpMyAdmin account.

For tables like CategoryTable, ClassTable, EquipmentTable, RangeDefinitionTable, RangeTable, RoundDefinitionTable, and TargetFaceSizeTable, which are tables with definite data depending on the archery's rule (could be used as a look-up table), our team manually input the INSERT queries into SQL code file.

For other tables, we generate INSERT queries using Python, where each line takes responsibility for inserting the data generated into the real table in the database. The data for each query is randomised based on the tables' requirements, which are coded within the Python files while generating data.

- Example of Python files: for EndScoreTable

Mutating EndScoreTable:

```
14 def generate_sql_insert_statements():
15     sql_statements = []
16     for range_id in range(1, 14):
17         print(range_id)
18         if range_id <= 5:
19             for _ in range(6):
20                 arrows = generate_random_arrows()
21                 total_end_score = calculate_total_end_score(arrows)
22                 sql = f"""
23 --- For RangeID: {range_id}
24 INSERT INTO EndScoreTable (Arrow1, Arrow2, Arrow3, Arrow4, Arrow5, Arrow6, TotalEndScore, RangeID)
25 VALUES ({arrows[0]}, {arrows[1]}, {arrows[2]}, {arrows[3]}, {arrows[4]}, {arrows[5]}, {total_end_score}, {range_id});
26 """
27                 sql_statements.append(sql)
28         else:
29             for _ in range(5): # 4 arrow tuples per RangeID
30                 arrows = generate_random_arrows()
31                 total_end_score = calculate_total_end_score(arrows)
32                 sql = f"""
33 --- For RangeID: {range_id}
34 INSERT INTO EndScoreTable (Arrow1, Arrow2, Arrow3, Arrow4, Arrow5, Arrow6, TotalEndScore, RangeID)
35 VALUES ({arrows[0]}, {arrows[1]}, {arrows[2]}, {arrows[3]}, {arrows[4]}, {arrows[5]}, {total_end_score}, {range_id});
36 """
37                 sql_statements.append(sql)
38
39     return sql_statements
```

Function generating queries

```
def write_sql_to_file(filename, sql_statements):
    with open(filename, 'w') as file:
        for statement in sql_statements:
            file.write(statement + "\n")
    print(f"SQL statements written to {filename}")
```

Function writing the queries into SQL code file

Example of data inserted

Table	Data insert example	Explanation
ArcherTable	<pre> 1 INSERT INTO ArcherTable (ArcherFirstName, ArcherLastName, ArcherGender, ArcherDateOfBirth) 2 ('Kevin', 'James', 'Male', '1980-05-14'); </pre>	<p>Insert 'Kevin', 'James', 'Male', and '1980-05-14' as the archer's first and last name, gender, and date of birth respectively into ArcherTable.</p> <p>ID is auto-incremented, therefore, not required to be mentioned when inserting into the table.</p>
CategoryTable	<pre> 1 INSERT INTO CategoryTable(CategoryID, ClassID, EquipmentID) 2 (1, 2, 5); </pre>	<p>Insert 1, 2, and 5 as CategoryID (primary key), ClassID, and EquipmentID. Values whose attributes are ClassID and EquipmentID would be further used to look up ClassTable and EquipmentTable.</p>
ClassTable	<pre> 1 INSERT INTO ClassTable (ClassID, AgeGroup, Gender) 2 (1, 'Open', 'Female'); </pre>	<p>Insert 1, 'Open', and 'Female' as ClassID, AgeGroup, and Gender of the table. Age group and gender determine the category the archer competes in according to the archery rules.</p>
CompetitionTable	<pre> 1 INSERT INTO CompetitionTable(CategoryID, Yearly, Description) 2 (1, 1, "Male Open Recurve Compound"); </pre>	<p>Insert 1, 1, and string "Male Open Recurve Compound" as the category in the competition, whether the club championship is yearly, and the description of each item.</p> <p>CompetitionID is auto-incremented, therefore, not required to be mentioned when inserting into the table.</p>
EndScoreTable	<pre> 1 INSERT INTO EndScoreTable (Arrow1, Arrow2, EndScore, RangeType) 2 (8, 3, 3, 9, 9, 9, 41, 1); </pre>	<p>Insert the shooting score of each arrow, the end's total score computed by the sum of each arrow, and the range type taken from RangeTable which indicates the type of round an archer is shooting.</p> <p>EndID is auto-incremented, therefore, not required to be mentioned when inserting into the table.</p>
EquipmentTable	<pre> 1 INSERT INTO EquipmentTable(EquipmentID, EquipmentName) 2 (1, 'R'), 3 (2, 'C'), 4 (3, 'B'), 5 (4, 'L'), 6 (5, 'RC'), 7 (6, 'RCB'), 8 (7, 'RCBL'), 9 (8, 'BL'); </pre>	<p>Insert the ID and equipment name used in archery. There are 8 types of equipment used, so 8 items are inserted into this table.</p>
RangeDefinitionTable	<pre> 1 INSERT INTO RangeDefinitionTable(RangeID, RangeDistance) 2 (1, 90), 3 (2, 70), 4 (3, 60), 5 (4, 50), 6 (5, 40), 7 (6, 30), 8 (7, 20), 9 (8, 10); </pre>	<p>Insert the ID and the distance to the target when shooting. For example, where ID is 1, the shooting distance is 90 meters. Each item has a different distance.</p>
RangeTable	<pre> 1 INSERT INTO RangeTable(RangeTypeID, RangeName, RangeDistance, NumberEnds, NumberArrows) 2 -- WA90/1440 3 (1, 'WA90/1440', 1, 6, 36, 2) 4 (1, 'WA90/1440', 2, 6, 36, 2), 5 (1, 'WA90/1440', 4, 6, 36, 1), 6 (1, 'WA90/1440', 6, 6, 36, 1); </pre>	<p>Insert the range type, rounds according to the official target archery rounds, range (distance), number of ends at that distance, number of arrows to be shot at that distance, and target face type into the table. For example, regarding the WA90/1440 round, at a distance with RangeID is 1 (90 meters), archers shoot 6 ends (36 arrows) to the target face ID is 2 (sized 122 cm).</p>

		PK is auto-incremented, therefore, not required to be mentioned when inserting into the table.
RoundDefinitionTable	<pre> 1 INSERT INTO RoundDefinitionTable (RoundDefinitionID, CompetitionID, Score, TotalArrows) 2 ('WA90/1440', 1440, 144); </pre>	<p>Insert 'WA90/1440', 1440, 144 as official target archery rounds (according to the rule), possible score and total arrows to be shot in each round respectively.</p> <p>RoundDefinitionID is auto-incremented, therefore, not required to be mentioned when inserting into the table.</p>
RoundTable	<pre> 1 INSERT INTO RoundTable(EquipmentID, CompetitionID, ArcherID, Date, ShootTime, EndID) 2 (1, 37, 35, '2023-09-20', '14:00:00', 1); </pre>	Insert User number 35 at ArcherTable with Recurve as equipment, CompetitionID set to 37 as practice session, date and shoot time, and EndID set to 1 as the first end value.
TargetFaceSizeTable	<pre> 1 INSERT INTO TargetFaceSizeTable(TargetFaceSize) 2 (1, '80cm'), 3 (2, '122cm'); </pre>	Each item indicates the size of the target (e.g. TargetFaceSize 80cm, 122cm).

Use cases and SQL statements, transactions

Stakeholders	Use Case	Scenario	SQL statements required	Transaction
Archer	Score Entry and Update	The archer wants to enter scores via a handheld device immediately after the rounds just end to ensure accuracy and timeliness. Author: @TRONG DAT HOANG	<pre> 1 -- Use Case: Archer Score Entry and Update 2 INSERT INTO RoundTable 3 VALUES (5, 41, '2022-09-15')</pre>	No. A single insert query does not require any transactions.
	Retrieve and Sort Past Results	They have to be able to record a date and time, round, equipment (some archers shoot several types of equipment on different days)		
	Round Definitions and Details (Including Equivalent Rounds)	To monitor progress in performance, an archer must be able to look up, filter and sort previous scores according to round type and date. Archer's best score for a round is determined if required. Author: @MAI AN NGUYEN	<pre> 1 -- Use Case: Retrieve and Sort Past Results 2 SELECT r.ShootDate, r.Score, r.RoundDefinitionID 3 FROM RoundTable r 4 JOIN EndScoreTable e ON r.EndScoreID = e.ID 5 WHERE r.ArcherID = 87 6 ORDER BY r.ShootDate DESC</pre>	No. Read-only queries does not required any transactions.
	Competition lookup	Archers need to be able to look up the requirement for thorough definitions of each round, including equivalent rounds based on current standards. Author: @VINH LE	<pre> 1 -- Use Case: Round Definitions and Details (Including Equivalent Rounds) 2 SELECT r.RoundDefinitionID, r.Definition 3 FROM RoundDefinitionTable r 4 WHERE r.RoundDefinitionID = 1</pre>	No. Read-only queries does not required any transactions.
		Archers could look up competition results and see how everyone has placed with their scores. Club's best score for a round and the archer who shot the round is determined if called. Author: @XUAN TUAN MINH NGUYEN	<pre> 1 -- Use Case: Competition lookup 2 SELECT a.ArcherFirstName, a.ArcherLastName, a.ArcherID, r.ShootDate, r.Score, c.CompetitionName, c.CompetitionID 3 FROM RoundTable r 4 JOIN ArcherTable a ON r.ArcherID = a.ArcherID 5 JOIN EndScoreTable e ON r.EndScoreID = e.ID 6 JOIN CompetitionTable c ON e.CompetitionID = c.CompetitionID 7 WHERE c.CompetitionID = 1 8 ORDER BY e.TotalEndScore DESC</pre>	No. Read-only queries does not required any transactions.

Recorder	Archers, Rounds, Competition, Entry Author: @XUAN TUAN MINH NGUYEN	<p>The recorder could enter new archers, new rounds, and new competition.</p> <p>Some competitions have to be able to be identified as part of a club championship.</p>	<pre> 1 -- Use Case: Recorder 2 -- Insert a new archer 3 INSERT INTO ArcherTable 4 VALUES ('John', 'Doe', 5 -- Insert a new round 6 INSERT INTO RoundDefinitionTable 7 VALUES ('NewRound/1440' 8 -- Insert a new competition 9 INSERT INTO CompetitionTable 10 VALUES (1, 0, 'New Competition') </pre>	<p>Yes. These operations are related and should be executed as a single transaction to ensure data integrity</p> <p>Transaction formula:</p> <pre> 1 START TRANSACTION 2 3 INSERT INTO ArcherTable 4 VALUES ('John', 'Doe') 5 6 INSERT INTO RoundDefinitionTable 7 VALUES ('NewRound/1440') 8 9 INSERT INTO CompetitionTable 10 VALUES (1, 0, 'New Competition') 11 12 COMMIT; 13 </pre>
Recorders have detailed score-recording capabilities	Author: @Phuong Doanh HA	<p>To ensure proper records, the recorder must precisely submit detailed scores, including per-arrow and end statistics.</p> <p>Each end has to be identified as to its position in the round score (the range and the end order in that range). Arrows are recorded highest to lowest arrow score within an end.</p> <p>Some of the scores have to be able to be linked to a competition.</p>	<pre> 1 -- Use Case: Detailed Score 2 INSERT INTO EndScoreTable 3 VALUES (10, 9, 8, 10, 9) </pre>	<p>No. A single insert query does not require any transactions.</p>
Class (divisions) and equipment definition lookup	Author: @MAI AN NGUYEN	<p>The category can be identified in the absence of user input as there is data needed to identify the archer's division (age, gender) and a definition of the default equipment to link to.</p>	<pre> 1 -- Use Case: Class (Division) 2 SELECT c.ClassID, c.AgeGroup 3 FROM ClassTable c 4 JOIN CategoryTable cat ON c.CategoryID = cat.CategoryID 5 JOIN EquipmentTable e ON c.EquipmentID = e.EquipmentID 6 WHERE c.AgeGroup = 'Under 18' </pre>	<p>No. Read-only queries does not required any transactions.</p>
Validation of historical data and past competition.	Author: @TRONG	<p>The recorder necessitates that the system keep track of previous scores and competitions, particularly when round definitions change.</p>	<pre> 1 -- Use Case: Validation 2 SELECT r.ShootDate, r.Score, r.CompetitionID 3 FROM RoundTable r 4 JOIN EndScoreTable e ON r.RoundID = e.RoundID 5 JOIN RoundDefinitionTable rd ON r.RoundID = rd.RoundID 6 WHERE r.CompetitionID = 'C1' 7 ORDER BY r.ShootDate DESC </pre>	<p>No. Read-only queries does not required any transactions.</p>

DAT
HOANG

Major-specific work

AI/Data Science

Mai An Nguyen

I (Mai An) did an AI database and specifically analyzed the archers, one of the main stakeholders, especially their gender and age.

I have used Workbook to create a Project AI Database. After getting all the information on the database result in our project, I have inserted a few SELECT queries to show tables of information. As a result, tables and visualisations (provided by Kinetica) were created, which from there, were used for my analysis.

▼ SQL queries

SQL queries applied in AI database project for analysis

Create a new table to divide the age range of Archers

```
1 drop table IF exist AgeGroupTable;
2 create table AgeGroupTable (
3     AgeGroupName varchar(255) not null,
4     StartDate date,
5     EndDate date,
6     PRIMARY KEY (AgeGroupName)
7 );
8
9 INSERT INTO AgeGroupTable (AgeGroupName, StartDate, EndDate) VALUES
10 ('Below 14 (U14)', '2010-05-26', '2024-05-26'),
11 ('From 14 to 15 (U16)', '2008-05-26', '2010-05-26'),
12 ('From 16 to 17 (U18)', '2006-05-26', '2008-05-26'),
13 ('From 18 to 20 (U21)', '2003-05-26', '2006-05-26'),
14 ('From 21 to 29 (Open)', '1994-05-26', '2003-05-26'),
15 ('From 30 to 39 (Open)', '1984-05-26', '1994-05-26'),
16 ('From 40 to 49 (Open)', '1974-05-26', '1984-05-26'),
17 ('Over 50 (50+)', '1964-05-26', '1974-05-26'),
18 ('Over 60 (60+)', '1954-05-26', '1964-05-26'),
19 ('Over 70 (70+)', '1800-05-26', '1954-05-26');
```

Show the distribution of gender of archers

```
1 SELECT
2     ArcherGender as Gender,
3     COUNT (*) as Count
4 FROM ArcherTable
5 GROUP BY ArcherGender
6 ORDER BY Gender;
```

Show the distribution of the age range of archers

```
1 SELECT
2     GroupAge.AgeGroupName as AgeRange,
3     count(ArcherInfo.ArcherDOB) as Count
4 FROM
5     ArcherTable ArcherInfo
```

```

6 INNER JOIN
7     AgeGroupTable GroupAge
8 on ((ArcherInfo.ArcherDOB >= GroupAge.StartDate or GroupAge.StartDate = NULL)
9 and (ArcherInfo.ArcherDOB < GroupAge.EndDate or GroupAge.EndDate = NULL))
10 GROUP BY
11     AgeRange
12 ORDER BY AgeRange

```

Show the distribution of the age range of archers according to their gender

For both gender

```

1 SELECT
2     ArcherInfo.ArcherGender as Gender,
3     GroupAge.AgeGroupName as AgeRange,
4     count(ArcherInfo.ArcherGender) as HeadCount
5 FROM
6     ArcherTable ArcherInfo
7 INNER JOIN
8     AgeGroupTable GroupAge
9 on ((ArcherInfo.ArcherDOB >= GroupAge.StartDate or GroupAge.StartDate = NULL)
10 and (ArcherInfo.ArcherDOB < GroupAge.EndDate or GroupAge.EndDate = NULL))
11 GROUP BY
12     AgeRange, Gender
13 ORDER BY
14     AgeRange, Gender

```

According to female

```

1 SELECT
2     GroupAge.AgeGroupName as AgeRange,
3     count(ArcherInfo.ArcherGender) as Count
4 FROM
5     ArcherTable ArcherInfo
6 INNER JOIN
7     AgeGroupTable GroupAge
8 on ((ArcherInfo.ArcherDOB >= GroupAge.StartDate or GroupAge.StartDate = NULL)
9 and (ArcherInfo.ArcherDOB < GroupAge.EndDate or GroupAge.EndDate = NULL))
10 WHERE
11     ArcherInfo.ArcherGender = 'Female'
12 GROUP BY
13     AgeRange
14 ORDER BY AgeRange

```

According to male

```

1 SELECT
2     GroupAge.AgeGroupName as AgeRange,
3     count(ArcherInfo.ArcherGender) as Count
4 FROM
5     ArcherTable ArcherInfo
6 INNER JOIN
7     AgeGroupTable GroupAge
8 on ((ArcherInfo.ArcherDOB >= GroupAge.StartDate or GroupAge.StartDate = NULL)
9 and (ArcherInfo.ArcherDOB < GroupAge.EndDate or GroupAge.EndDate = NULL))
10 WHERE
11     ArcherInfo.ArcherGender = 'Male'
12 GROUP BY
13     AgeRange

```

Here are the key findings recorded:

- The number of females and males using this is equal (which is the cause of inserting fake data).
- The archers using this are within the age range of 18 to 50. The number of archers under 21 disregard of gender is the lowest, while archers joined the Open division take the highest proportion. For the age range 21 to 30, 30 to 40, and 40 to 50, which all belong to the open division, the number of people is evenly distributed.
- Regarding each age range
 - Only within the age range 21 to 30, the number of females is more than the number of males (by 10 people). For other ranges, the number of males is more than the number of females (by below 5 people).
- Regarding female archers
 - Archers aged below 21 make up around 7% of all female archers using the site. Archers aged from 21 to 30, 30 to 40, and 40 to 50 make up around 30% each.
 - Archers aged from 21 to 30 are the largest in number among all female archers.
- Regarding male archers
 - Archers aged below 21 make up around 9% of all male archers using the site. Archers aged from 21 to 30, 30 to 40, and 40 to 50 make up around 30% each.
 - Archers aged from 40 to 50 are the largest in number among all male archers.

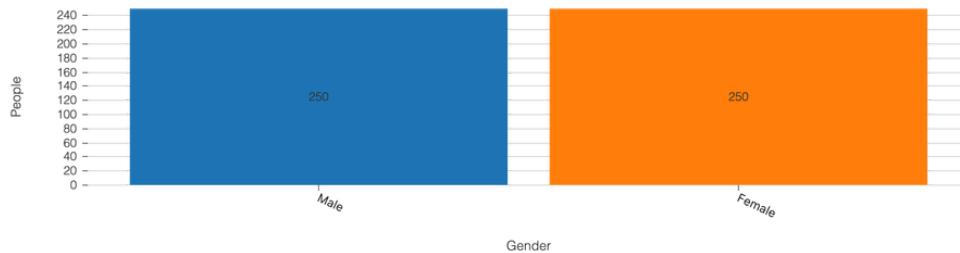
Dummy data were used, which might cause inaccurate analysis.

The tables and visualisations supporting the analysis (the result of using Kinetica to analyse the database) are displayed within the expansion below.

▼ Tables and Visualisations supporting the analysis

Tables and Visualisations supporting the analysis

Number of Archers vs. Gender

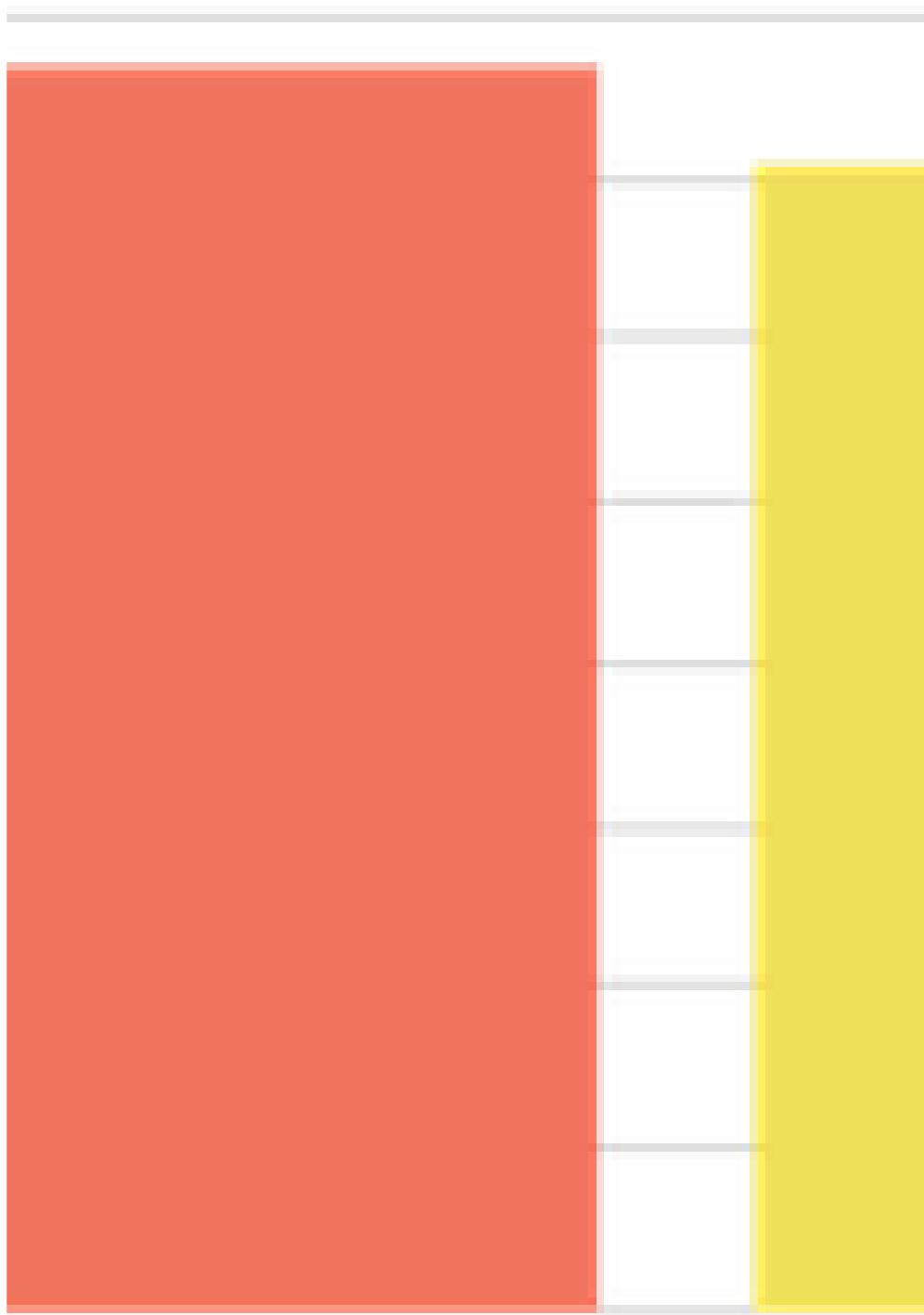


Bar Chart of Number of Archers Regarding Gender

Table copied:

-----.
Gender Count
----- -----
Male 250
Female 250
'-----'

Number of Archers vs. Age Groups



Age Groups

Bar Chart of Number of Archers Regarding Age Groups

Table copied:

AgeRange	Count
From 18 to 20 (U21)	42
From 21 to 29 (Open)	154
From 30 to 39 (Open)	142
From 40 to 49 (Open)	162

Number of Archers vs. Age Groups Regarding Gender

Gender	AgeRange	HeadCount
Female	From 18 to 20 (U21)	19
Male	From 18 to 20 (U21)	23
Female	From 21 to 29 (Open)	82
Male	From 21 to 29 (Open)	72
Female	From 30 to 39 (Open)	69
Male	From 30 to 39 (Open)	73
Female	From 40 to 49 (Open)	80
Male	From 40 to 49 (Open)	82

Table of Number of Archers in Terms of
Gender and Age Groups

Table copied:

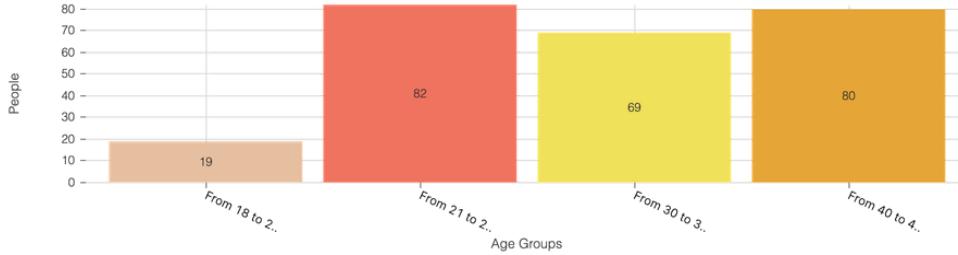
Gender	AgeRange	HeadCount
Female	From 18 to 20 (U21)	19
Male	From 18 to 20 (U21)	23
Female	From 21 to 29 (Open)	82
Male	From 21 to 29 (Open)	72
Female	From 30 to 39 (Open)	69

```
| Male | From 30 to 39 (Open) | 73 |
```

```
| Female | From 40 to 49 (Open) | 80 |
```

```
| Male | From 40 to 49 (Open) | 82 |
```

Number of Female Archers vs. Age Groups



Bar Chart of Number of Female Archers Regarding Age Groups

Table copied:

```
| AgeRange | Count |
```

```
| ----- | ----- |
```

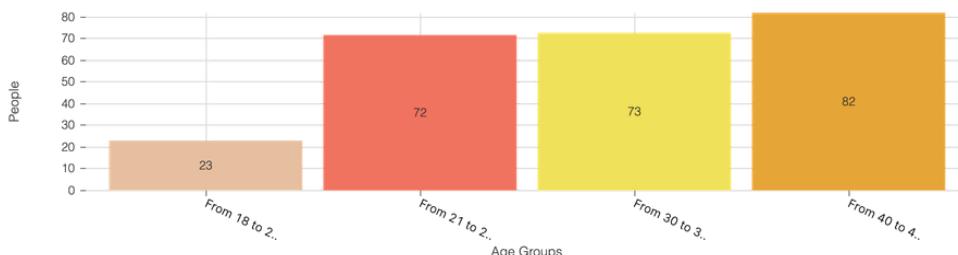
```
| From 18 to 20 (U21) | 19 |
```

```
| From 21 to 29 (Open) | 82 |
```

```
| From 30 to 39 (Open) | 69 |
```

```
| From 40 to 49 (Open) | 80 |
```

Number of Male Archers vs. Age Groups



Bar Chart of Number of Male Archers Regarding Age Groups

Table copied:

```
| AgeRange | Count |
```

```
| ----- | ----- |
```

```
| From 18 to 20 (U21) | 23 |
```

```
| From 21 to 29 (Open) | 72 |
```

```
| From 30 to 39 (Open) | 73 |
```

'-----'

Cybersecurity

Xuan Tuan Minh Nguyen

For the cybersecurity major, I (Xuan Tuan Minh Nguyen) have implemented the SQL injection prevention using the two powerful tools that are created by PHP, which are: SQL sanitization function and SQL prepared statements function. Here is the main code that is has logical implementation used for validating if the statement has SQL Injection or not.

Validator

```

1 <?php
2 interface IValidatorStrategy
3 {
4     public function sanitize(?string $input): string;
5     public function isSanitized(?string $input): bool;
6 }
7
8 final class ValidatorStrategy implements IValidatorStrategy
9 {
10    public function sanitize(?string $input): string
11    {
12        if ($input === null) {
13            throw new Exception("Input can not be empty");
14        } else if ($input === '') {
15            throw new Exception("Input can not be empty");
16        } else {
17            return filter_var($input, FILTER_SANITIZE_FULL_SPECIAL_CHARS);
18        }
19    }
20
21    public function isSanitized(?string $input): bool
22    {
23        if ($input === null || $input === '')
24            return false;
25        return $input === $this->sanitize($input);
26    }
27 }
28 ?>
```

Detailed Explanations:

- In the given code, I have created the sanitize function that will check if the input is null or an empty string, if it is then the code will return throw an exception and let user know that they have not inputted anything.
- Otherwise, it will check that if the input has any special characters that could cause an SQL Injection, parsing it to prevent any special characters and return the parsed string.
- For example, if the query is: "Robert"); DROP TABLE Students;--", then the query should return "Robert"); DROP TABLE Students;--".
- In addition, I have written a function called `isSanitized()` to ensure that the input must be matched with the parsed string, this function will be used later on for the rest of the code.

For more information, here is the full code that I have implemented to construct a Representational State Transfer (REST) Application Programming Interface (API) using PHP as the main language, OOP as the main design and NestJS Architecture and the main architecture.

▼ Service

- Interface

```
1 <?php
2 include_once "api/DTO/archer/archer.dto.php";
3 interface IArcherService
4 {
5     public function read(int $limit, int $offset): mysqli_result | bool;
6     public function readBy(ReadArcherDTO $dto): mysqli_result | bool;
7     public function create(CreateArcherDTO $dto): mysqli_result | bool;
8     public function update(UpdateArcherDTO $dto): mysqli_result | bool;
9     public function delete>DeleteArcherDTO $dto): mysqli_result | bool;
10 }
11 ?>
```

- Implementation

```

40         $params[] = $dto->ArcherID;
41         $types .= "i"; // integer
42     }
43     if (!empty($dto->ArcherFirstName)) {
44         $conditions[] = "ArcherFirstName = ?";
45         $params[] = $dto->ArcherFirstName;
46         $types .= "s"; // string
47     }
48     if (!empty($dto->ArcherLastName)) {
49         $conditions[] = "ArcherLastName = ?";
50         $params[] = $dto->ArcherLastName;
51         $types .= "s"; // string
52     }
53     if (!empty($dto->ArcherGender)) {
54         $conditions[] = "ArcherGender = ?";
55         $params[] = $dto->ArcherGender;
56         $types .= "s"; // string
57     }
58     if (!empty($dto->ArcherDOB)) {
59         $conditions[] = "ArcherDOB = ?";
60         $params[] = $dto->ArcherDOB;
61         $types .= "s"; // string
62     }
63
64     $query = "SELECT ArcherID, ArcherFirstName, ArcherLastName, ArcherGender, ArcherDOB FROM $this->tableName";
65     if (count($conditions) > 0) {
66         $query .= " WHERE " . implode(" AND ", $conditions);
67     }
68
69     $stmt = $this->conn->prepare($query);
70     $stmt->bind_param($types, ...$params);
71     $stmt->execute();
72     return $stmt->get_result();
73 }
74
75 public function create(CreateArcherDTO $dto): mysqli_result | bool
76 {
77     $dto->validate();
78
79     $stmt = $this->conn->prepare("INSERT INTO $this->tableName (ArcherFirstName, ArcherLastName, ArcherGender, ArcherDOB) VALUES (?, ?, ?, ?)");
80     $stmt->bind_param("ssss", $dto->ArcherFirstName, $dto->ArcherLastName, $dto->ArcherGender, $dto->ArcherDOB);
81     $stmt->execute();
82     return $stmt->get_result();
83 }
84
85 public function update(UpdateArcherDTO $dto): mysqli_result | bool
86 {
87     $dto->validate();
88
89     $stmt = $this->conn->prepare("UPDATE $this->tableName SET ArcherFirstName = ?, ArcherLastName = ?, ArcherGender = ?, ArcherDOB = ? WHERE ArcherID = ?");
90     $stmt->bind_param("ssssi", $dto->ArcherFirstName, $dto->ArcherLastName, $dto->ArcherGender, $dto->ArcherDOB, $dto->ArcherID);
91     $stmt->execute();
92     return $stmt->get_result();
93 }
94
95 public function delete>DeleteArcherDTO $dto): mysqli_result | bool
96 {
97     $dto->validate();

```

```

98
99     $stmt = $this->conn->prepare("DELETE FROM $this->tableName WHERE ArcherID = ?");
100    $stmt->bind_param("i", $dto->ArcherID);
101    $stmt->execute();
102    return $stmt->get_result();
103 }
104 }
105 ?>

```

▼ Data Transfer Object (DTO)

- Interface

```

1 <?php
2 interface IDataTransferObject
3 {
4     public function validate(): bool;
5 }
6 ?>

```

- DTO

```

1 <?php
2 include_once "api/DTO/DTO.interface.php";
3 include_once "api/Validator/Validator.strategy.php";
4 final class ReadArcherDTO implements IDataTransferObject
5 {
6     public ?string $ArcherID;
7     public ?string $ArcherFirstName;
8     public ?string $ArcherLastName;
9     public ?string $ArcherGender;
10    public ?string $ArcherDOB;
11    private int $TotalRecords;
12    private IValidatorStrategy $Validator;
13
14    public function __construct(?array $data, int $totalRecords)
15    {
16        $this->Validator = new ValidatorStrategy();
17        $this->ArcherID = $this->Validator->sanitize($data['ArcherID']);
18        $this->ArcherFirstName = $this->Validator->sanitize($data['ArcherFirstName']);
19        $this->ArcherLastName = $this->Validator->sanitize($data['ArcherLastName']);
20        $this->ArcherGender = $this->Validator->sanitize($data['ArcherGender']);
21        $this->ArcherDOB = $this->Validator->sanitize($data['ArcherDOB']);
22        $this->TotalRecords = $totalRecords;
23    }
24
25    public function validate(): bool
26    {
27        if (empty($this->ArcherID) && empty($this->ArcherFirstName) && empty($this->ArcherLastName) && empty($this->ArcherDOB)) {
28            throw new Exception("Invalid data. All fields are required.");
29        } else if ($this->ArcherID > $this->TotalRecords) {
30            throw new Exception("Unable to read archer. Archer does not exist.");
31        } else if ($this->ArcherGender != "Male" && $this->ArcherGender != "Female") {
32            throw new Exception("Unable to read archer, Archer Gender must be Male or Female");
33        } else if (!preg_match("/^[-0-9]{4}-[-0-9]{2}-[-0-9]{2}$/", $this->ArcherDOB)) {
34            throw new Exception("Unable to read archer, Archer DOB must be in the format YYYY-MM-DD");
35        } else if (!$this->Validator->isSanitized($this->ArcherID) && !$this->Validator->isSanitized($this->ArcherFirstName) && !$this->Validator->isSanitized($this->ArcherLastName)) {
36            throw new Exception("Invalid data. Data is not sanitized.");
37        } else {
38            return true;
39        }
40    }
41
42    public function getArcherID(): string
43    {
44        return $this->ArcherID;
45    }
46
47    public function getArcherFirstName(): string
48    {
49        return $this->ArcherFirstName;
50    }
51
52    public function getArcherLastName(): string
53    {
54        return $this->ArcherLastName;
55    }
56
57    public function getArcherGender(): string
58    {
59        return $this->ArcherGender;
60    }
61
62    public function getArcherDOB(): string
63    {
64        return $this->ArcherDOB;
65    }
66
67    public function getTotalRecords(): int
68    {
69        return $this->TotalRecords;
70    }
71
72    public function setArcherID(?string $ArcherID): void
73    {
74        $this->ArcherID = $ArcherID;
75    }
76
77    public function setArcherFirstName(?string $ArcherFirstName): void
78    {
79        $this->ArcherFirstName = $ArcherFirstName;
80    }
81
82    public function setArcherLastName(?string $ArcherLastName): void
83    {
84        $this->ArcherLastName = $ArcherLastName;
85    }
86
87    public function setArcherGender(?string $ArcherGender): void
88    {
89        $this->ArcherGender = $ArcherGender;
90    }
91
92    public function setArcherDOB(?string $ArcherDOB): void
93    {
94        $this->ArcherDOB = $ArcherDOB;
95    }
96
97    public function setTotalRecords(int $TotalRecords): void
98    {
99        $this->TotalRecords = $TotalRecords;
100    }
101
102    public function setValidator(IValidatorStrategy $Validator): void
103    {
104        $this->Validator = $Validator;
105    }
106
107    public function getValidator(): IValidatorStrategy
108    {
109        return $this->Validator;
110    }
111
112    public function __toString(): string
113    {
114        return "ReadArcherDTO [ArcherID=" . $this->ArcherID . ", ArcherFirstName=" . $this->ArcherFirstName . ", ArcherLastName=" . $this->ArcherLastName . ", ArcherGender=" . $this->ArcherGender . ", ArcherDOB=" . $this->ArcherDOB . ", TotalRecords=" . $this->TotalRecords . "]";
115    }
116
117    public function __clone()
118    {
119        $this->Validator = null;
120    }
121
122    public function __wakeup()
123    {
124        $this->Validator = null;
125    }
126
127    public function __sleep()
128    {
129        $this->Validator = null;
130    }
131
132    public function __destruct()
133    {
134        $this->Validator = null;
135    }
136
137    public function __get(string $name)
138    {
139        if ($name === "ArcherID") {
140            return $this->ArcherID;
141        }
142        if ($name === "ArcherFirstName") {
143            return $this->ArcherFirstName;
144        }
145        if ($name === "ArcherLastName") {
146            return $this->ArcherLastName;
147        }
148        if ($name === "ArcherGender") {
149            return $this->ArcherGender;
150        }
151        if ($name === "ArcherDOB") {
152            return $this->ArcherDOB;
153        }
154        if ($name === "TotalRecords") {
155            return $this->TotalRecords;
156        }
157        if ($name === "Validator") {
158            return $this->Validator;
159        }
160        throw new Exception("Unknown property: " . $name);
161    }
162
163    public function __set(string $name, mixed $value)
164    {
165        if ($name === "ArcherID") {
166            $this->ArcherID = $value;
167        }
168        if ($name === "ArcherFirstName") {
169            $this->ArcherFirstName = $value;
170        }
171        if ($name === "ArcherLastName") {
172            $this->ArcherLastName = $value;
173        }
174        if ($name === "ArcherGender") {
175            $this->ArcherGender = $value;
176        }
177        if ($name === "ArcherDOB") {
178            $this->ArcherDOB = $value;
179        }
180        if ($name === "TotalRecords") {
181            $this->TotalRecords = $value;
182        }
183        if ($name === "Validator") {
184            $this->Validator = $value;
185        }
186        else {
187            throw new Exception("Unknown property: " . $name);
188        }
189    }
190
191    public function __isset(string $name): bool
192    {
193        if ($name === "ArcherID") {
194            return $this->ArcherID !== null;
195        }
196        if ($name === "ArcherFirstName") {
197            return $this->ArcherFirstName !== null;
198        }
199        if ($name === "ArcherLastName") {
200            return $this->ArcherLastName !== null;
201        }
202        if ($name === "ArcherGender") {
203            return $this->ArcherGender !== null;
204        }
205        if ($name === "ArcherDOB") {
206            return $this->ArcherDOB !== null;
207        }
208        if ($name === "TotalRecords") {
209            return $this->TotalRecords !== null;
210        }
211        if ($name === "Validator") {
212            return $this->Validator !== null;
213        }
214        return false;
215    }
216
217    public function __unserialize(mixed $data)
218    {
219        $this->Validator = null;
220    }
221
222    public function __serialize()
223    {
224        $this->Validator = null;
225    }
226
227    public function __clone()
228    {
229        $this->Validator = null;
230    }
231
232    public function __wakeup()
233    {
234        $this->Validator = null;
235    }
236
237    public function __sleep()
238    {
239        $this->Validator = null;
240    }
241
242    public function __destruct()
243    {
244        $this->Validator = null;
245    }
246
247    public function __get(string $name)
248    {
249        if ($name === "ArcherID") {
250            return $this->ArcherID;
251        }
252        if ($name === "ArcherFirstName") {
253            return $this->ArcherFirstName;
254        }
255        if ($name === "ArcherLastName") {
256            return $this->ArcherLastName;
257        }
258        if ($name === "ArcherGender") {
259            return $this->ArcherGender;
260        }
261        if ($name === "ArcherDOB") {
262            return $this->ArcherDOB;
263        }
264        if ($name === "TotalRecords") {
265            return $this->TotalRecords;
266        }
267        if ($name === "Validator") {
268            return $this->Validator;
269        }
270        throw new Exception("Unknown property: " . $name);
271    }
272
273    public function __set(string $name, mixed $value)
274    {
275        if ($name === "ArcherID") {
276            $this->ArcherID = $value;
277        }
278        if ($name === "ArcherFirstName") {
279            $this->ArcherFirstName = $value;
280        }
281        if ($name === "ArcherLastName") {
282            $this->ArcherLastName = $value;
283        }
284        if ($name === "ArcherGender") {
285            $this->ArcherGender = $value;
286        }
287        if ($name === "ArcherDOB") {
288            $this->ArcherDOB = $value;
289        }
290        if ($name === "TotalRecords") {
291            $this->TotalRecords = $value;
292        }
293        if ($name === "Validator") {
294            $this->Validator = $value;
295        }
296        else {
297            throw new Exception("Unknown property: " . $name);
298        }
299    }
300
301    public function __isset(string $name): bool
302    {
303        if ($name === "ArcherID") {
304            return $this->ArcherID !== null;
305        }
306        if ($name === "ArcherFirstName") {
307            return $this->ArcherFirstName !== null;
308        }
309        if ($name === "ArcherLastName") {
310            return $this->ArcherLastName !== null;
311        }
312        if ($name === "ArcherGender") {
313            return $this->ArcherGender !== null;
314        }
315        if ($name === "ArcherDOB") {
316            return $this->ArcherDOB !== null;
317        }
318        if ($name === "TotalRecords") {
319            return $this->TotalRecords !== null;
320        }
321        if ($name === "Validator") {
322            return $this->Validator !== null;
323        }
324        return false;
325    }
326
327    public function __unserialize(mixed $data)
328    {
329        $this->Validator = null;
330    }
331
332    public function __serialize()
333    {
334        $this->Validator = null;
335    }
336
337    public function __clone()
338    {
339        $this->Validator = null;
340    }
341
342    public function __wakeup()
343    {
344        $this->Validator = null;
345    }
346
347    public function __sleep()
348    {
349        $this->Validator = null;
350    }
351
352    public function __destruct()
353    {
354        $this->Validator = null;
355    }
356
357    public function __get(string $name)
358    {
359        if ($name === "ArcherID") {
360            return $this->ArcherID;
361        }
362        if ($name === "ArcherFirstName") {
363            return $this->ArcherFirstName;
364        }
365        if ($name === "ArcherLastName") {
366            return $this->ArcherLastName;
367        }
368        if ($name === "ArcherGender") {
369            return $this->ArcherGender;
370        }
371        if ($name === "ArcherDOB") {
372            return $this->ArcherDOB;
373        }
374        if ($name === "TotalRecords") {
375            return $this->TotalRecords;
376        }
377        if ($name === "Validator") {
378            return $this->Validator;
379        }
380        throw new Exception("Unknown property: " . $name);
381    }
382
383    public function __set(string $name, mixed $value)
384    {
385        if ($name === "ArcherID") {
386            $this->ArcherID = $value;
387        }
388        if ($name === "ArcherFirstName") {
389            $this->ArcherFirstName = $value;
390        }
391        if ($name === "ArcherLastName") {
392            $this->ArcherLastName = $value;
393        }
394        if ($name === "ArcherGender") {
395            $this->ArcherGender = $value;
396        }
397        if ($name === "ArcherDOB") {
398            $this->ArcherDOB = $value;
399        }
400        if ($name === "TotalRecords") {
401            $this->TotalRecords = $value;
402        }
403        if ($name === "Validator") {
404            $this->Validator = $value;
405        }
406        else {
407            throw new Exception("Unknown property: " . $name);
408        }
409    }
410
411    public function __isset(string $name): bool
412    {
413        if ($name === "ArcherID") {
414            return $this->ArcherID !== null;
415        }
416        if ($name === "ArcherFirstName") {
417            return $this->ArcherFirstName !== null;
418        }
419        if ($name === "ArcherLastName") {
420            return $this->ArcherLastName !== null;
421        }
422        if ($name === "ArcherGender") {
423            return $this->ArcherGender !== null;
424        }
425        if ($name === "ArcherDOB") {
426            return $this->ArcherDOB !== null;
427        }
428        if ($name === "TotalRecords") {
429            return $this->TotalRecords !== null;
430        }
431        if ($name === "Validator") {
432            return $this->Validator !== null;
433        }
434        return false;
435    }
436
437    public function __unserialize(mixed $data)
438    {
439        $this->Validator = null;
440    }
441
442    public function __serialize()
443    {
444        $this->Validator = null;
445    }
446
447    public function __clone()
448    {
449        $this->Validator = null;
450    }
451
452    public function __wakeup()
453    {
454        $this->Validator = null;
455    }
456
457    public function __sleep()
458    {
459        $this->Validator = null;
460    }
461
462    public function __destruct()
463    {
464        $this->Validator = null;
465    }
466
467    public function __get(string $name)
468    {
469        if ($name === "ArcherID") {
470            return $this->ArcherID;
471        }
472        if ($name === "ArcherFirstName") {
473            return $this->ArcherFirstName;
474        }
475        if ($name === "ArcherLastName") {
476            return $this->ArcherLastName;
477        }
478        if ($name === "ArcherGender") {
479            return $this->ArcherGender;
480        }
481        if ($name === "ArcherDOB") {
482            return $this->ArcherDOB;
483        }
484        if ($name === "TotalRecords") {
485            return $this->TotalRecords;
486        }
487        if ($name === "Validator") {
488            return $this->Validator;
489        }
490        throw new Exception("Unknown property: " . $name);
491    }
492
493    public function __set(string $name, mixed $value)
494    {
495        if ($name === "ArcherID") {
496            $this->ArcherID = $value;
497        }
498        if ($name === "ArcherFirstName") {
499            $this->ArcherFirstName = $value;
500        }
501        if ($name === "ArcherLastName") {
502            $this->ArcherLastName = $value;
503        }
504        if ($name === "ArcherGender") {
505            $this->ArcherGender = $value;
506        }
507        if ($name === "ArcherDOB") {
508            $this->ArcherDOB = $value;
509        }
510        if ($name === "TotalRecords") {
511            $this->TotalRecords = $value;
512        }
513        if ($name === "Validator") {
514            $this->Validator = $value;
515        }
516        else {
517            throw new Exception("Unknown property: " . $name);
518        }
519    }
520
521    public function __isset(string $name): bool
522    {
523        if ($name === "ArcherID") {
524            return $this->ArcherID !== null;
525        }
526        if ($name === "ArcherFirstName") {
527            return $this->ArcherFirstName !== null;
528        }
529        if ($name === "ArcherLastName") {
530            return $this->ArcherLastName !== null;
531        }
532        if ($name === "ArcherGender") {
533            return $this->ArcherGender !== null;
534        }
535        if ($name === "ArcherDOB") {
536            return $this->ArcherDOB !== null;
537        }
538        if ($name === "TotalRecords") {
539            return $this->TotalRecords !== null;
540        }
541        if ($name === "Validator") {
542            return $this->Validator !== null;
543        }
544        return false;
545    }
546
547    public function __unserialize(mixed $data)
548    {
549        $this->Validator = null;
550    }
551
552    public function __serialize()
553    {
554        $this->Validator = null;
555    }
556
557    public function __clone()
558    {
559        $this->Validator = null;
560    }
561
562    public function __wakeup()
563    {
564        $this->Validator = null;
565    }
566
567    public function __sleep()
568    {
569        $this->Validator = null;
570    }
571
572    public function __destruct()
573    {
574        $this->Validator = null;
575    }
576
577    public function __get(string $name)
578    {
579        if ($name === "ArcherID") {
580            return $this->ArcherID;
581        }
582        if ($name === "ArcherFirstName") {
583            return $this->ArcherFirstName;
584        }
585        if ($name === "ArcherLastName") {
586            return $this->ArcherLastName;
587        }
588        if ($name === "ArcherGender") {
589            return $this->ArcherGender;
590        }
591        if ($name === "ArcherDOB") {
592            return $this->ArcherDOB;
593        }
594        if ($name === "TotalRecords") {
595            return $this->TotalRecords;
596        }
597        if ($name === "Validator") {
598            return $this->Validator;
599        }
600        throw new Exception("Unknown property: " . $name);
601    }
602
603    public function __set(string $name, mixed $value)
604    {
605        if ($name === "ArcherID") {
606            $this->ArcherID = $value;
607        }
608        if ($name === "ArcherFirstName") {
609            $this->ArcherFirstName = $value;
610        }
611        if ($name === "ArcherLastName") {
612            $this->ArcherLastName = $value;
613        }
614        if ($name === "ArcherGender") {
615            $this->ArcherGender = $value;
616        }
617        if ($name === "ArcherDOB") {
618            $this->ArcherDOB = $value;
619        }
620        if ($name === "TotalRecords") {
621            $this->TotalRecords = $value;
622        }
623        if ($name === "Validator") {
624            $this->Validator = $value;
625        }
626        else {
627            throw new Exception("Unknown property: " . $name);
628        }
629    }
630
631    public function __isset(string $name): bool
632    {
633        if ($name === "ArcherID") {
634            return $this->ArcherID !== null;
635        }
636        if ($name === "ArcherFirstName") {
637            return $this->ArcherFirstName !== null;
638        }
639        if ($name === "ArcherLastName") {
640            return $this->ArcherLastName !== null;
641        }
642        if ($name === "ArcherGender") {
643            return $this->ArcherGender !== null;
644        }
645        if ($name === "ArcherDOB") {
646            return $this->ArcherDOB !== null;
647        }
648        if ($name === "TotalRecords") {
649            return $this->TotalRecords !== null;
650        }
651        if ($name === "Validator") {
652            return $this->Validator !== null;
653        }
654        return false;
655    }
656
657    public function __unserialize(mixed $data)
658    {
659        $this->Validator = null;
660    }
661
662    public function __serialize()
663    {
664        $this->Validator = null;
665    }
666
667    public function __clone()
668    {
669        $this->Validator = null;
670    }
671
672    public function __wakeup()
673    {
674        $this->Validator = null;
675    }
676
677    public function __sleep()
678    {
679        $this->Validator = null;
680    }
681
682    public function __destruct()
683    {
684        $this->Validator = null;
685    }
686
687    public function __get(string $name)
688    {
689        if ($name === "ArcherID") {
690            return $this->ArcherID;
691        }
692        if ($name === "ArcherFirstName") {
693            return $this->ArcherFirstName;
694        }
695        if ($name === "ArcherLastName") {
696            return $this->ArcherLastName;
697        }
698        if ($name === "ArcherGender") {
699            return $this->ArcherGender;
700        }
701        if ($name === "ArcherDOB") {
702            return $this->ArcherDOB;
703        }
704        if ($name === "TotalRecords") {
705            return $this->TotalRecords;
706        }
707        if ($name === "Validator") {
708            return $this->Validator;
709        }
710        throw new Exception("Unknown property: " . $name);
711    }
712
713    public function __set(string $name, mixed $value)
714    {
715        if ($name === "ArcherID") {
716            $this->ArcherID = $value;
717        }
718        if ($name === "ArcherFirstName") {
719            $this->ArcherFirstName = $value;
720        }
721        if ($name === "ArcherLastName") {
722            $this->ArcherLastName = $value;
723        }
724        if ($name === "ArcherGender") {
725            $this->ArcherGender = $value;
726        }
727        if ($name === "ArcherDOB") {
728            $this->ArcherDOB = $value;
729        }
730        if ($name === "TotalRecords") {
731            $this->TotalRecords = $value;
732        }
733        if ($name === "Validator") {
734            $this->Validator = $value;
735        }
736        else {
737            throw new Exception("Unknown property: " . $name);
738        }
739    }
740
741    public function __isset(string $name): bool
742    {
743        if ($name === "ArcherID") {
744            return $this->ArcherID !== null;
745        }
746        if ($name === "ArcherFirstName") {
747            return $this->ArcherFirstName !== null;
748        }
749        if ($name === "ArcherLastName") {
750            return $this->ArcherLastName !== null;
751        }
752        if ($name === "ArcherGender") {
753            return $this->ArcherGender !== null;
754        }
755        if ($name === "ArcherDOB") {
756            return $this->ArcherDOB !== null;
757        }
758        if ($name === "TotalRecords") {
759            return $this->TotalRecords !== null;
760        }
761        if ($name === "Validator") {
762            return $this->Validator !== null;
763        }
764        return false;
765    }
766
767    public function __unserialize(mixed $data)
768    {
769        $this->Validator = null;
770    }
771
772    public function __serialize()
773    {
774        $this->Validator = null;
775    }
776
777    public function __clone()
778    {
779        $this->Validator = null;
780    }
781
782    public function __wakeup()
783    {
784        $this->Validator = null;
785    }
786
787    public function __sleep()
788    {
789        $this->Validator = null;
790    }
791
792    public function __destruct()
793    {
794        $this->Validator = null;
795    }
796
797    public function __get(string $name)
798    {
799        if ($name === "ArcherID") {
800            return $this->ArcherID;
801        }
802        if ($name === "ArcherFirstName") {
803            return $this->ArcherFirstName;
804        }
805        if ($name === "ArcherLastName") {
806            return $this->ArcherLastName;
807        }
808        if ($name === "ArcherGender") {
809            return $this->ArcherGender;
810        }
811        if ($name === "ArcherDOB") {
812            return $this->ArcherDOB;
813        }
814        if ($name === "TotalRecords") {
815            return $this->TotalRecords;
816        }
817        if ($name === "Validator") {
818            return $this->Validator;
819        }
820        throw new Exception("Unknown property: " . $name);
821    }
822
823    public function __set(string $name, mixed $value)
824    {
825        if ($name === "ArcherID") {
826            $this->ArcherID = $value;
827        }
828        if ($name === "ArcherFirstName") {
829            $this->ArcherFirstName = $value;
830        }
831        if ($name === "ArcherLastName") {
832            $this->ArcherLastName = $value;
833        }
834        if ($name === "ArcherGender") {
835            $this->ArcherGender = $value;
836        }
837        if ($name === "ArcherDOB") {
838            $this->ArcherDOB = $value;
839        }
840        if ($name === "TotalRecords") {
841            $this->TotalRecords = $value;
842        }
843        if ($name === "Validator") {
844            $this->Validator = $value;
845        }
846        else {
847            throw new Exception("Unknown property: " . $name);
848        }
849    }
850
851    public function __isset(string $name): bool
852    {
853        if ($name === "ArcherID") {
854            return $this->ArcherID !== null;
855        }
856        if ($name === "ArcherFirstName") {
857            return $this->ArcherFirstName !== null;
858        }
859        if ($name === "ArcherLastName") {
860            return $this->ArcherLastName !== null;
861        }
862        if ($name === "ArcherGender") {
863            return $this->ArcherGender !== null;
864        }
865        if ($name === "ArcherDOB") {
866            return $this->ArcherDOB !== null;
867        }
868        if ($name === "TotalRecords") {
869            return $this->TotalRecords !== null;
870        }
871        if ($name === "Validator") {
872            return $this->Validator !== null;
873        }
874        return false;
875    }
876
877    public function __unserialize(mixed $data)
878    {
879        $this->Validator = null;
880    }
881
882    public function __serialize()
883    {
884        $this->Validator = null;
885    }
886
887    public function __clone()
888    {
889        $this->Validator = null;
890    }
891
892    public function __wakeup()
893    {
894        $this->Validator = null;
895    }
896
897    public function __sleep()
898    {
899        $this->Validator = null;
900    }
901
902    public function __destruct()
903    {
904        $this->Validator = null;
905    }
906
907    public function __get(string $name)
908    {
909        if ($name === "ArcherID") {
910            return $this->ArcherID;
911        }
912        if ($name === "ArcherFirstName") {
913            return $this->ArcherFirstName;
914        }
915        if ($name === "ArcherLastName") {
916            return $this->ArcherLastName;
917        }
918        if ($name === "ArcherGender") {
919            return $this->ArcherGender;
920        }
921        if ($name === "ArcherDOB") {
922            return $this->ArcherDOB;
923        }
924        if ($name === "TotalRecords") {
925            return $this->TotalRecords;
926        }
927        if ($name === "Validator") {
928            return $this->Validator;
929        }
930        throw new Exception("Unknown property: " . $name);
931    }
932
933    public function __set(string $name, mixed $value)
934    {
935        if ($name === "ArcherID") {
936            $this->ArcherID = $value;
937        }
938        if ($name === "ArcherFirstName") {
939            $this->ArcherFirstName = $value;
940        }
941        if ($name === "ArcherLastName") {
942            $this->ArcherLastName = $value;
943        }
944        if ($name === "ArcherGender") {
945            $this->ArcherGender = $value;
946        }
947        if ($name === "ArcherDOB") {
948            $this->ArcherDOB = $value;
949        }
950        if ($name === "TotalRecords") {
951            $this->TotalRecords = $value;
952        }
953        if ($name === "Validator") {
954            $this->Validator = $value;
955        }
956        else {
957            throw new Exception("Unknown property: " . $name);
958        }
959    }
960
961    public function __isset(string $name): bool
962    {
963        if ($name === "ArcherID") {
964            return $this->ArcherID !== null;
965        }
966        if ($name === "ArcherFirstName") {
967            return $this->ArcherFirstName !== null;
968        }
969        if ($name === "ArcherLastName") {
970            return $this->ArcherLastName !== null;
971        }
972        if ($name === "ArcherGender") {
973            return $this->ArcherGender !== null;
974        }
975        if ($name === "ArcherDOB") {
976            return $this->ArcherDOB !== null;
977        }
978        if ($name === "TotalRecords") {
979            return $this->TotalRecords !== null;
980        }
981        if ($name === "Validator") {
982            return $this->Validator !== null;
983        }
984        return false;
985    }
986
987    public function __unserialize(mixed $data)
988    {
989        $this->Validator = null;
990    }
991
992    public function __serialize()
993    {
994        $this->Validator = null;
995    }
996
997    public function __clone()
998    {
999        $this->Validator = null;
1000    }
1001
1002    public function __wakeup()
1003    {
1004        $this->Validator = null;
1005    }
1006
1007    public function __sleep()
1008    {
1009        $this->Validator = null;
1010    }
1011
1012    public function __destruct()
1013    {
1014        $this->Validator = null;
1015    }
1016
1017    public function __get(string $name)
1018    {
1019        if ($name === "ArcherID") {
1020            return $this->ArcherID;
1021        }
1022        if ($name === "ArcherFirstName") {
1023            return $this->ArcherFirstName;
1024        }
1025        if ($name === "ArcherLastName") {
1026            return $this->ArcherLastName;
1027        }
1028        if ($name === "ArcherGender") {
1029            return $this->ArcherGender;
1030        }
1031        if ($name === "ArcherDOB") {
1032            return $this->ArcherDOB;
1033        }
1034        if ($name === "TotalRecords") {
1035            return $this->TotalRecords;
1036        }
1037        if ($name === "Validator") {
1038            return $this->Validator;
1039        }
1040        throw new Exception("Unknown property: " . $name);
1041    }
1042
1043    public function __set(string $name, mixed $value)
1044    {
1045        if ($name === "ArcherID") {
1046            $this->ArcherID = $value;
1047        }
1048        if ($name === "ArcherFirstName") {
1049            $this->ArcherFirstName = $value;
1050        }
1051        if ($name === "ArcherLastName") {
1052            $this->ArcherLastName = $value;
1053        }
1054        if ($name === "ArcherGender") {
1055            $this->ArcherGender = $value;
1056        }
1057        if ($name === "ArcherDOB") {
1058            $this->ArcherDOB = $value;
1059        }
1060        if ($name === "TotalRecords") {
1061            $this->TotalRecords = $value;
1062        }
1063        if ($name === "Validator") {
1064            $this->Validator = $value;
1065        }
1066        else {
1067            throw new Exception("Unknown property: " . $name);
1068        }
1069    }
1070
1071    public function __isset(string $name): bool
1072    {
1073        if ($name === "ArcherID") {
1074            return $this->ArcherID !== null;
1075        }
1076        if ($name === "ArcherFirstName") {
1077            return $this->ArcherFirstName !== null;
1078        }
1079        if ($name === "ArcherLastName") {
1080            return $this->ArcherLastName !== null;
1081        }
1082        if ($name === "ArcherGender") {
1083            return $this->ArcherGender !== null;
1084        }
1085        if ($name === "ArcherDOB") {
1086            return $this->ArcherDOB !== null;
1087        }
1088        if ($name === "TotalRecords") {
1089            return $this->TotalRecords !== null;
1090        }
1091        if ($name === "Validator") {
1092            return $this->Validator !== null;
1093        }
1094        return false;
1095    }
1096
1097    public function __unserialize(mixed $data)
1098    {
1099        $this->Validator = null;
1100    }
1101
1102    public function __serialize()
1103    {
1104        $this->Validator = null;
1105    }
1106
1107    public function __clone()
1108    {
1109        $
```

```

38     }
39 }
40 }
41 final class CreateArcherDTO implements IDataTransferObject
42 {
43     public ?string $ArcherFirstName;
44     public ?string $ArcherLastName;
45     public ?string $ArcherGender;
46     public ?string $ArcherDOB;
47     private IValidatorStrategy $Validator;
48     public function __construct(?array $data)
49     {
50         $this->Validator = new ValidatorStrategy();
51         $this->ArcherFirstName = $this->Validator->sanitize($data['ArcherFirstName']);
52         $this->ArcherLastName = $this->Validator->sanitize($data['ArcherLastName']);
53         $this->ArcherGender = $this->Validator->sanitize($data['ArcherGender']);
54         $this->ArcherDOB = $this->Validator->sanitize($data['ArcherDOB']);
55     }
56     public function validate(): bool
57     {
58         if (empty($this->ArcherFirstName) && empty($this->ArcherLastName) && empty($this->ArcherGender) &&
59             throw new Exception("Unable to create archer. Data is incomplete.");
60     } else if ($this->ArcherGender != "Male" && $this->ArcherGender != "Female") {
61         throw new Exception("Unable to create archer, Archer Gender must be Male or Female");
62     } else if (!preg_match("/^([0-9]{4})-([0-9]{2})-([0-9]{2})$/", $this->ArcherDOB)) {
63         throw new Exception("Unable to create archer, Archer DOB must be in the format YYYY-MM-DD");
64     } else if (!$this->Validator->isSanitized($this->ArcherFirstName) && !$this->Validator->isSanitized($this->ArcherLastName) && !$this->Validator->isSanitized($this->ArcherGender) && !$this->Validator->isSanitized($this->ArcherDOB)) {
65         throw new Exception("Invalid data. Data is not sanitized.");
66     } else {
67         return true;
68     }
69 }
70 }
71
72 final class UpdateArcherDTO implements IDataTransferObject
73 {
74     public ?string $ArcherID;
75     public ?string $ArcherFirstName;
76     public ?string $ArcherLastName;
77     public ?string $ArcherGender;
78     public ?string $ArcherDOB;
79     private int $TotalRecords;
80     private IValidatorStrategy $Validator;
81     public function __construct(?array $data, int $totalRecords)
82     {
83         $this->Validator = new ValidatorStrategy();
84         $this->ArcherID = $this->Validator->sanitize($data['ArcherID']);
85         $this->ArcherFirstName = $this->Validator->sanitize($data['ArcherFirstName']);
86         $this->ArcherLastName = $this->Validator->sanitize($data['ArcherLastName']);
87         $this->ArcherGender = $this->Validator->sanitize($data['ArcherGender']);
88         $this->ArcherDOB = $this->Validator->sanitize($data['ArcherDOB']);
89         $this->TotalRecords = $this->Validator->sanitize($totalRecords);
90     }
91     public function validate(): bool
92     {
93         if (empty($this->ArcherID) && empty($this->ArcherFirstName) && empty($this->ArcherLastName) && empty($this->ArcherDOB) && empty($this->TotalRecords)) {
94             throw new Exception("Invalid data. All fields are required.");
95         } else if ($this->ArcherID > $this->TotalRecords) {

```

```

96         throw new Exception("Unable to update archer. Archer does not exist.");
97     } else if ($this->ArcherGender != "Male" && $this->ArcherGender != "Female") {
98         throw new Exception("Unable to update archer, Archer Gender must be Male or Female");
99     } else if (!preg_match("/^([0-9]{4}-[0-9]{2}-[0-9]{2})$/", $this->ArcherDOB)) {
100        throw new Exception("Unable to update archer, Archer DOB must be in the format YYYY-MM-DD");
101    } else if (!$this->Validator->isSanitized($this->ArcherID) && !$this->Validator->isSanitized($this->
102        throw new Exception("Invalid data. Data is not sanitized.");
103    } else {
104        return true;
105    }
106}
107}
108
109 final class DeleteArcherDTO implements IDataTransferObject
110 {
111     public ?string $ArcherID;
112     private int $TotalRecords;
113     private IValidatorStrategy $Validator;
114     public function __construct(?array $data, int $totalRecords)
115     {
116         $this->Validator = new ValidatorStrategy();
117         $this->ArcherID = $this->Validator->sanitize($data['ArcherID']);
118         $this->TotalRecords = $this->Validator->sanitize($totalRecords);
119     }
120     public function validate(): bool
121     {
122         if (empty($this->ArcherID)) {
123             throw new Exception("Invalid data. ArcherID is required.");
124         } else if ($this->ArcherID > $this->TotalRecords) {
125             throw new Exception("Unable to delete archer. Archer does not exist.");
126         } else if (!$this->Validator->isSanitized($this->ArcherID) && !$this->Validator->isSanitized($this->
127             throw new Exception("Invalid data. ArcherID is not sanitized.");
128         } else {
129             return true;
130         }
131     }
132 }
133 ?>
```

Controller

- GET Method

```

1  <?php
2  include_once "api/Service/ArcherTable/ArcherTable.service.php";
3  include_once "api/configs/database.connect.php";
4  header("Content-Type: application/json; charset=UTF-8");
5  $db = new Database();
6  $conn = $db->connectDatabase();
7  $archer = new ArcherTableService($conn);
8  $page = isset($_GET['page']) ? $_GET['page'] : 1;
9  $limit = isset($_GET['limit']) ? $_GET['limit'] : 10;
10 $offset = ($page - 1) * $limit;
11 $result = $archer->read($limit, $offset);
12 $num = mysqli_num_rows($result);
13
14 $total_query = "SELECT COUNT(*) FROM ArcherTable";
15 $total_result = mysqli_query($conn, $total_query);
16 $total_row = mysqli_fetch_row($total_result);
```

```

17 $total_records = $total_row[0];
18 $total_pages = ceil($total_records / $limit);
19
20 $archerArray = array();
21 $archerArray['results'] = array();
22 $archerArray['pagination'] = array(
23     "total_records" => $total_records,
24     "total_pages" => $total_pages,
25     "current_page" => $page,
26     "limit" => $limit
27 );
28 if ($num > 0) {
29     while ($row = mysqli_fetch_assoc($result)) {
30         $archerItem = array(
31             "ArcherID" => $row['ArcherID'],
32             "ArcherFirstName" => $row['ArcherFirstName'],
33             "ArcherLastName" => $row['ArcherLastName'],
34             "ArcherGender" => $row['ArcherGender'],
35             "ArcherDOB" => $row['ArcherDOB']
36         );
37
38         array_push($archerArray['results'], $archerItem);
39     }
40     http_response_code(200);
41     echo json_encode($archerArray);
42 } else {
43     http_response_code(404);
44     echo json_encode(array("message" => "Archer is empty."));
45 }
46
47 http_response_code(200);
48 ?>

```

- GET Method By Specific Information

```

1 <?php
2 include_once "api/Service/ArcherTable/ArcherTable.service.php";
3 include_once "api/configs/database.connect.php";
4 include_once "api/DTO/archer/archer.dto.php";
5 header("Content-Type: application/json; charset=UTF-8");
6
7 $db = new Database();
8 $conn = $db->connectDatabase();
9 $archer = new ArcherTableService($conn);
10 $data = json_decode(file_get_contents("php://input"), true);
11
12 $total_query = "SELECT COUNT(*) FROM ArcherTable";
13 $total_result = mysqli_query($conn, $total_query);
14 $total_row = mysqli_fetch_row($total_result);
15 $total_records = $total_row[0];
16 $archerArray = array();
17 $archerArray['results'] = array();
18
19 try {
20     $dto = new ReadArcherDTO($data, $total_records);
21     $result = $archer->readBy($dto);
22
23     if ($result && mysqli_num_rows($result) > 0) {
24         while ($row = mysqli_fetch_assoc($result)) {

```

```

25     $archerItem = array(
26         "ArcherID" => $row['ArcherID'],
27         "ArcherFirstName" => $row['ArcherFirstName'],
28         "ArcherLastName" => $row['ArcherLastName'],
29         "ArcherGender" => $row['ArcherGender'],
30         "ArcherDOB" => $row['ArcherDOB']
31     );
32     array_push($archerArray['results'], $archerItem);
33 }
34 http_response_code(200);
35 echo json_encode($archerArray);
36 } else {
37     http_response_code(404);
38     echo json_encode(array("message" => "No archer found."));
39 }
40 } catch (Exception $e) {
41     http_response_code(400);
42     echo json_encode(array("message" => $e->getMessage()));
43 }
44

```

- PUT Method

```

1 <?php
2 include_once "api/Service/ArcherTable/ArcherTable.service.php";
3 include_once "api/configs/database.connect.php";
4 include_once "api/DTO/archer/archer.dto.php";
5 header("Content-Type: application/json; charset=UTF-8");
6 $db = new Database();
7 $conn = $db->connectDatabase();
8 $archer = new ArcherTableService($conn);
9
10 $data = json_decode(file_get_contents("php://input"), true);
11 $total_query = "SELECT COUNT(*) FROM ArcherTable";
12 $total_result = mysqli_query($conn, $total_query);
13 $total_row = mysqli_fetch_row($total_result);
14 $total_records = $total_row[0];
15
16 try {
17     $dto = new UpdateArcherDTO($data, $total_records);
18     $result = $archer->update($dto);
19     if ($result) {
20         http_response_code(200);
21         echo json_encode(array("message" => "Archer was updated."));
22     } else {
23         http_response_code(503);
24         echo json_encode(array("message" => "Unable to update archer."));
25     }
26 } catch (Exception $e) {
27     http_response_code(400);
28     echo json_encode(array("message" => $e->getMessage()));
29 }
30 ?>

```

- POST Method

```

1 <?php
2 include_once "api/Service/ArcherTable/ArcherTable.service.php";
3 include_once "api/configs/database.connect.php";

```

```

4 include_once "api/DTO/archer/archer.dto.php";
5 header("Content-Type: application/json; charset=UTF-8");
6 $db = new Database();
7 $conn = $db->connectDatabase();
8 $archer = new ArcherTableService($conn);
9
10 $data = json_decode(file_get_contents("php://input"), true);
11
12 try {
13     $dto = new CreateArcherDTO($data);
14     $result = $archer->create($dto);
15     if ($result) {
16         http_response_code(201);
17         echo json_encode(array("message" => "Archer was created."));
18     } else {
19         http_response_code(503);
20         echo json_encode(array("message" => "Unable to create archer."));
21     }
22 } catch (Exception $e) {
23     http_response_code(400);
24     echo json_encode(array("message" => $e->getMessage()));
25 }
26 ?>

```

- DELETE Method

```

1 <?php
2 include_once "api/Service/ArcherTable/ArcherTable.service.php";
3 include_once "api/configs/database.connect.php";
4 include_once "api/DTO/archer/archer.dto.php";
5 header("Content-Type: application/json; charset=UTF-8");
6 $db = new Database();
7 $conn = $db->connectDatabase();
8 $archer = new ArcherTableService($conn);
9
10 $data = json_decode(file_get_contents("php://input"), true);
11 $total_query = "SELECT COUNT(*) FROM ArcherTable";
12 $total_result = mysqli_query($conn, $total_query);
13 $total_row = mysqli_fetch_row($total_result);
14 $total_records = $total_row[0];
15 try {
16     $dto = new DeleteArcherDTO($data, $total_records);
17     $result = $archer->delete($dto);
18     if ($result) {
19         http_response_code(200);
20         echo json_encode(array("message" => "Archer was deleted."));
21     } else {
22         http_response_code(503);
23         echo json_encode(array("message" => "Unable to delete archer."));
24     }
25 } catch (Exception $e) {
26     http_response_code(400);
27     echo json_encode(array("message" => $e->getMessage()));
28 }
29 ?>

```

▼ Entry Point

```

2 <?php
3 error_reporting(E_ERROR | E_PARSE);
4 header("Access-Control-Allow-Origin: *");
5 header("Content-Type: application/json; charset=UTF-8");
6 header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");
7 header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");
8
9 $reqMethod = $_SERVER["REQUEST_METHOD"];
10 $reqURI = $_SERVER["REQUEST_URI"];
11
12 function getJSON()
13 {
14     if ($_SERVER['CONTENT_TYPE'] === 'application/json') {
15         $json = file_get_contents('php://input');
16         return json_decode($json, true);
17     }
18     return null;
19 }
20 $uriSeg = explode('/', trim($reqURI, '/'));
21
22 switch ($uriSeg[0]) {
23     case '':
24         http_response_code(200);
25         echo json_encode(array("message" => "Welcome to the SimonMajor API."));
26         break;
27     case 'archer':
28         if ($reqMethod == "GET") {
29             if (getJSON() !== null)
30                 include_once "./api/Controller/archer/readBy.controller.php";
31             else
32                 include_once "./api/Controller/archer/read.controller.php";
33         }
34         if ($reqMethod == "POST")
35             include_once "./api/Controller/archer/create.controller.php";
36         if ($reqMethod == "PUT")
37             include_once "./api/Controller/archer/update.controller.php";
38         if ($reqMethod == "DELETE")
39             include_once './api/Controller/archer/delete.controller.php';
40         break;
41     default:
42         http_response_code(404);
43         echo json_encode(array("message" => "Invalid endpoint."));
44         break;
45 }
46 ?>

```

Brief explanation:

- This code works by taking the input from the user and pass to the Controller with the appropriate method, which will be in the form of JSON objects.
- Then it will be transform into the Data Transfer Object Form and validating each field with the `ValidatorStrategy` class in order to check if that field has any chances for an SQL Injection class.
- Finally, the Data Transfer Object will be pass to the `ArcherService` class for interacting with the database, such as retrieve data from the database, add data to the database, edit data from the database or delete the data.

Key Testing:

- I have created two new routes, one for testing if the data is sanitized or not, and the other will be used for exploiting SQL Injection in the case that the code does not have any sanitizations.

Test Cases

- Controller to handle checking if data is properly sanitized

```

1 <?php
2 include_once "api/Validator/Validator.strategy.php";
3 header("Content-Type: application/json; charset=UTF-8");
4 $validator = new ValidatorStrategy();
5 $data = json_decode(file_get_contents("php://input"), true);
6
7 if (!$data || !isset($data['input'])) {
8     http_response_code(400);
9     echo json_encode(array("message" => "Invalid input. Provide JSON with an 'input' field."));
10    exit;
11 }
12
13 $input = $data['input'];
14 $sanitizedInput = $validator->sanitize($input);
15 $isSanitized = $validator->isSanitized($sanitizedInput);
16
17 $response = array(
18     "original_input" => $input,
19     "sanitized_output" => $sanitizedInput,
20     "is_sanitized" => $isSanitized
21 );
22
23 http_response_code(200);
24 echo json_encode($response);
25 ?>
```

- Controller to handle vulnerabilities

```

1 <?php
2 include_once "api/configs/database.connect.php";
3 include_once "api/Service/ArcherTable/ArcherTable.service.php";
4 include_once "api/Validator/Validator.strategy.php";
5
6 header("Content-Type: application/json; charset=UTF-8");
7
8 $db = new Database();
9 $conn = $db->connectDatabase();
10
11 $data = json_decode(file_get_contents("php://input"), true);
12
13 if (!$data) {
14     http_response_code(400);
15     echo json_encode(array("message" => "Invalid input."));
16     exit;
17 }
18
19 $archerID = $data['ArcherID'] ?? '';
20 $archerFirstName = $data['ArcherFirstName'] ?? '';
21 $archerLastName = $data['ArcherLastName'] ?? '';
22 $archerGender = $data['ArcherGender'] ?? '';
23 $archerDOB = $data['ArcherDOB'] ?? '';
24
25 // Construct a vulnerable SQL query
```

```

26 $query = "SELECT ArcherID, ArcherFirstName, ArcherLastName, ArcherGender, ArcherDOB FROM ArcherTable WHERE Ar
27 $result = mysqli_query($conn, $query);
28
29 if ($result && mysqli_num_rows($result) > 0) {
30     $archerArray = array();
31     $archerArray['results'] = array();
32     while ($row = mysqli_fetch_assoc($result)) {
33         $archerItem = array(
34             "ArcherID" => $row['ArcherID'],
35             "ArcherFirstName" => $row['ArcherFirstName'],
36             "ArcherLastName" => $row['ArcherLastName'],
37             "ArcherGender" => $row['ArcherGender'],
38             "ArcherDOB" => $row['ArcherDOB']
39         );
40         array_push($archerArray['results'], $archerItem);
41     }
42     http_response_code(200);
43     echo json_encode($archerArray);
44 } else {
45     http_response_code(404);
46     echo json_encode(array("message" => "No archer found."));
47 }
48
49 $conn->close();
50 ?>

```

- Entry Point

```

1 <?php
2 error_reporting(E_ERROR | E_PARSE);
3 header("Access-Control-Allow-Origin: *");
4 header("Content-Type: application/json; charset=UTF-8");
5 header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");
6 header("Access-Control-Allow-Headers: Content-Type, Access-Control-Allow-Headers, Authorization, X-Requested-With");
7
8 $reqMethod = $_SERVER["REQUEST_METHOD"];
9 $reqURI = $_SERVER["REQUEST_URI"];
10
11 functiongetJSON()
12 {
13     if ($_SERVER['CONTENT_TYPE'] === 'application/json') {
14         $json = file_get_contents('php://input');
15         return json_decode($json, true);
16     }
17     return null;
18 }
19 $uriSeg = explode('/', trim($reqURI, '/'));
20
21 switch ($uriSeg[0]) {
22     case '':
23         http_response_code(200);
24         echo json_encode(array("message" => "Welcome to the SimonMajor API."));
25         break;
26     case 'archer':
27         if ($reqMethod == "GET") {
28             if (getJSON() != null)
29                 include_once "./api/Controller/archer/readBy.controller.php";
30             else
31

```

```

32         include_once "./api/Controller/archer/read.controller.php";
33     }
34     if ($reqMethod == "POST")
35         include_once "./api/Controller/archer/create.controller.php";
36     if ($reqMethod == "PUT")
37         include_once "./api/Controller/archer/update.controller.php";
38     if ($reqMethod == "DELETE")
39         include_once './api/Controller/archer/delete.controller.php';
40     break;
41 case 'validator':
42     if ($reqMethod == "POST")
43         include_once "./api/Controller/test/validator.controller.php";
44     break;
45 case 'vulnerable':
46     if ($reqMethod == "POST")
47         include_once "./api/Controller/test/vulnerable.controller.php";
48     break;
49 default:
50     http_response_code(404);
51     echo json_encode(array("message" => "Invalid endpoint."));
52     break;
53 }
54 ?>
55
56

```

- Here are the results for testing each cases.

▼ Result

The screenshot shows a browser-based testing interface. At the top, there are tabs for 'localhost:3300/archer' (selected), 'vulnerable.controller.php', 'index.php', and 'Validator.strategy.php M'. Below the tabs, the URL is set to 'http://localhost:3333/validator'. The 'Body' tab is selected, showing the JSON input:


```

1  {
2      "input": "Robert'); DROP TABLE Students;-- "
3  }
    
```

 The 'Response' tab is selected, displaying the JSON output:


```

1  {
2      "original_input": "Robert'); DROP TABLE Students;-- ",
3      "sanitized_output": "Robert"); DROP TABLE Students;-- ",
4      "is_sanitized": true
5  }
    
```

 The status bar at the bottom indicates: Status: 200 OK, Size: 134 Bytes, Time: 7 ms.

Figure 1: Check if the inputted data is properly sanitized

The screenshot shows a browser-based testing interface. The URL is 'http://localhost:3333/vulnerable'. The 'Body' tab is selected, showing the JSON input:


```

1  {
2      "ArcherID": "9 OR 1=1",
3      "ArcherFirstName": "Charlie",
4      "ArcherLastName": "LeClerc",
5      "ArcherGender": "Male",
6      "ArcherDOB": "1991-09-10"
7  }
    
```

 The 'Response' tab is selected, displaying the JSON output with multiple rows of results:


```

1  [
2      {
3          "ArcherID": "",
4          "ArcherFirstName": "Charlie",
5          "ArcherLastName": "LeClerc",
6          "ArcherGender": "Male",
7          "ArcherDOB": "1991-09-10"
8      },
9      {
10         "ArcherID": "9",
11         "ArcherFirstName": "Miki",
12         "ArcherLastName": "Jobs",
13         "ArcherGender": "Male",
14         "ArcherDOB": "2000-01-05"
15     },
16     {
17         "ArcherID": "156",
18         "ArcherFirstName": "Charlie",
19         "ArcherLastName": "LeClerc",
20         "ArcherGender": "Male",
21         "ArcherDOB": "1991-09-10"
22     },
23     {
24         "ArcherID": "557",
25         "ArcherFirstName": "Charlie",
26         "ArcherLastName": "LeClerc",
27         "ArcherGender": "Male",
28         "ArcherDOB": "1991-09-10"
29     },
30     {
31         "ArcherID": "558",
32         "ArcherFirstName": "Charlie",
33         "ArcherLastName": "LeClerc",
34         "ArcherGender": "Male",
35         "ArcherDOB": "1991-09-10"
36     },
37     {
38         "ArcherID": "559",
39         "ArcherFirstName": "Charlie",
40         "ArcherLastName": "LeClerc",
41         "ArcherGender": "Male",
42         "ArcherDOB": "1991-09-10"
43     },
44     {
45         "ArcherID": "560",
46         "ArcherFirstName": "Charlie",
47         "ArcherLastName": "LeClerc",
48         "ArcherGender": "Male",
49         "ArcherDOB": "1991-09-10"
50     }
    ]
    
```

 The status bar at the bottom indicates: Status: 200 OK, Size: 1.07 KB, Time: 59 ms.

Figure 2: SQL Injection Demonstration

- In the two given results, we could clearly see that in Figure 1, the inputted data has some special characters that could cause a SQL Injection. However, the inputted data has been sanitized successfully using the `sanitize()` function implemented in `ValidateStrategy` class, which resulted in reducing the chance for attackers to directly contact with the underlying database. Moreover, the `isSanitized()` function is also working correctly, checking that if the inputted function is sanitized or not.

- Moving to the second test case, we could see that although `ArcherID` is 9. However, with adding a small query behind the number 9, which is `1=1`, the database is successfully exploited and the data is not returning the Archer with ArcherID at number 9, but instead return another several Archers. Thus, the database is exploitable in the case where no sanitizations take in place.

Conclusion:

- In conclusion, SQL Injection is one of the most popular yet most vulnerable attack method that most hackers use in order to directly access into the database with any efforts.
- Although we can not always fully protected from SQL Injection, adding some sanitize functions could reduce the chance of getting prohibited access to the database by more than 70%. Thus, make the database hassle-free and more reliable.

Further information regarding the results of the Archer REST API will be shown in the below snippet.

Result of the Archer API

```

GET http://localhost:3333/archer
Status: 200 OK Size: 1.21 KB Time: 20 ms
Response Headers Cookies Results Docs
1 {
2   "results": [
3     {
4       "ArcherID": 1,
5       "ArcherFirstName": "Karin",
6       "ArcherLastName": "Jane",
7       "ArcherGender": "Male",
8       "ArcherDOB": "1980-05-14"
9     }
10    {
11      "ArcherID": 2,
12      "ArcherFirstName": "Gary",
13      "ArcherLastName": "Plastri",
14      "ArcherGender": "Male",
15      "ArcherDOB": "1980-02-21"
16    }
17    {
18      "ArcherID": 3,
19      "ArcherFirstName": "Oscar",
20      "ArcherLastName": "Edwards",
21      "ArcherGender": "Male",
22      "ArcherDOB": "1990-02-03"
23    }
24    {
25      "ArcherID": 4,
26      "ArcherFirstName": "Lebron",
27      "ArcherLastName": "James",
28      "ArcherGender": "Male",
29      "ArcherDOB": "1990-05-15"
30    }
31    {
32      "ArcherID": 5,
33      "ArcherFirstName": "Anthony",
34      "ArcherLastName": "Derrick",
35      "ArcherGender": "Male",
36      "ArcherDOB": "1990-05-22"
37    }
38    {
39      "ArcherID": 6,
40      "ArcherFirstName": "Ethan",
41      "ArcherLastName": "Hamilton",
42      "ArcherGender": "Male",
43      "ArcherDOB": "1990-06-18"
44    }
45    {
46      "ArcherID": 7,
47      "ArcherFirstName": "Liam",
48      "ArcherLastName": "Owen",
49      "ArcherGender": "Male",
50      "ArcherDOB": "1990-07-22"
51    }
52  ]
53 }
54 
```

Figure 3: GET Method

```

GET http://localhost:3333/archer
Status: 200 OK Size: 130 Bytes Time: 13 ms
Response Headers Cookies Results Docs
1 {
2   "results": [
3     {
4       "ArcherID": 3,
5       "ArcherFirstName": "Oscar",
6       "ArcherLastName": "Edwards",
7       "ArcherGender": "Male",
8       "ArcherDOB": "1990-02-03"
9     }
10   ]
11 }
12 
```

Figure 4: GET Method with specific details

The screenshot shows a REST client interface with a PUT request to `http://localhost:3333/archer`. The request body contains the following JSON:

```

1 {
2   "ArcherID": 7,
3   "ArcherFirstName": "charlie",
4   "ArcherLastName": "mazouz",
5   "ArcherGender": "Male",
6   "ArcherDOB": "1991-09-18"
7 }
  
```

The response status is 200 OK, size is 35 bytes, and time is 9 ms. The response body is:

```

1 {
2   "message": "Archer was updated."
3 }
  
```

Figure 5: PUT Method to update Archer's information

The screenshot shows a REST client interface with a POST request to `http://localhost:3333/archer`. The request body contains the following JSON:

```

1 {
2   "ArcherFirstName": "Minh",
3   "ArcherLastName": "Bee",
4   "ArcherGender": "Male",
5   "ArcherDOB": "1991-09-18"
6 }
  
```

The response status is 201 Created, size is 35 bytes, and time is 12 ms. The response body is:

```

1 {
2   "message": "Archer was created."
3 }
  
```

Figure 6: POST Method to create Archer's Information

The screenshot shows a REST client interface with a DELETE request to `http://localhost:3333/archer`. The request body contains the following JSON:

```

1 {
2   "ArcherID": 500
3 }
  
```

The response status is 200 OK, size is 35 bytes, and time is 13 ms. The response body is:

```

1 {
2   "message": "Archer was deleted."
3 }
  
```

Figure 7: DELETE method to delete Archer's information

For CyberSecurity Major, I have implemented the Database User Roles and Privileges.

As they specified in MySQLWorkBench.

Administrator Role: Admin

In order to preserve security and guarantee effective operations, it is imperative for managers of strong database systems to clearly define user roles and rights. Our database system has two different users, and each has roles and permissions customised to suit their needs.

The administrator role has been assigned to Admin, who has full access to and control over the database. With the ability to carry out a broad range of duties, an admin user can guarantee the database's integrity and proper operation. Among the admin's privileges are:

Creating, Modifying, and Deleting Tables.

Executing Queries.

Managing User Permissions.

Maintaining Database Security.

The admin user's detailed privileges span a variety of administrative responsibilities, ensuring that they can efficiently manage and maintain the database. These positions include DBA (Database Administrator), MaintenanceAdmin, ProcessAdmin, UserAdmin, and others, each with specialised competences required for comprehensive database management.

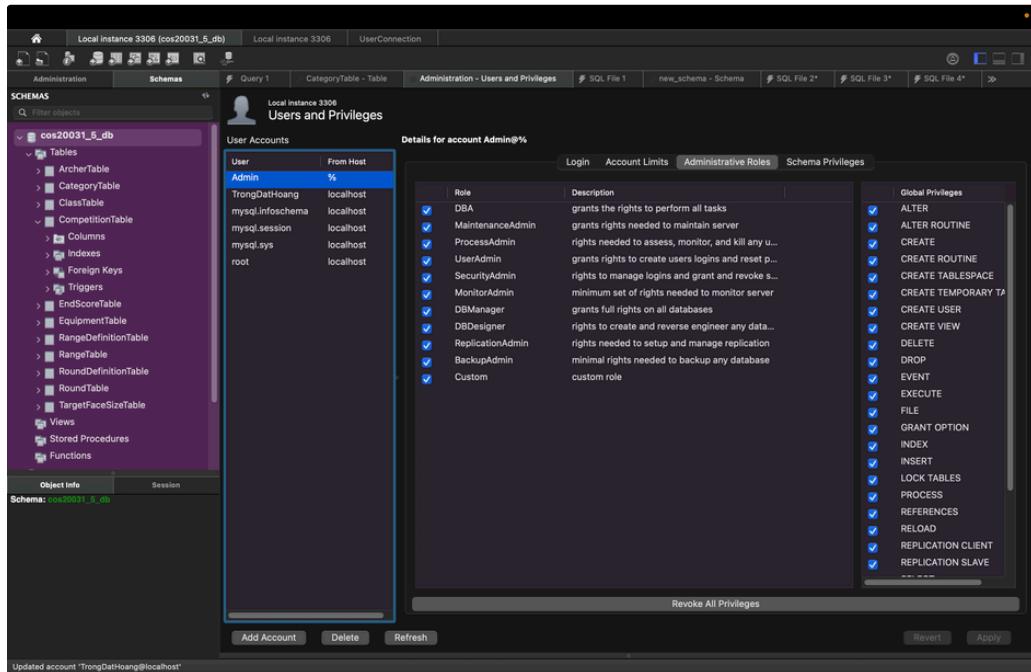


Figure 1: Configure Admin account's Administrative Roles.

In addition to the general administrative roles, the admin user has specialized schema-level permissions. These include the ability to build, modify, and delete routines, tables, and views, allowing the administrator to control the database structure extensively.

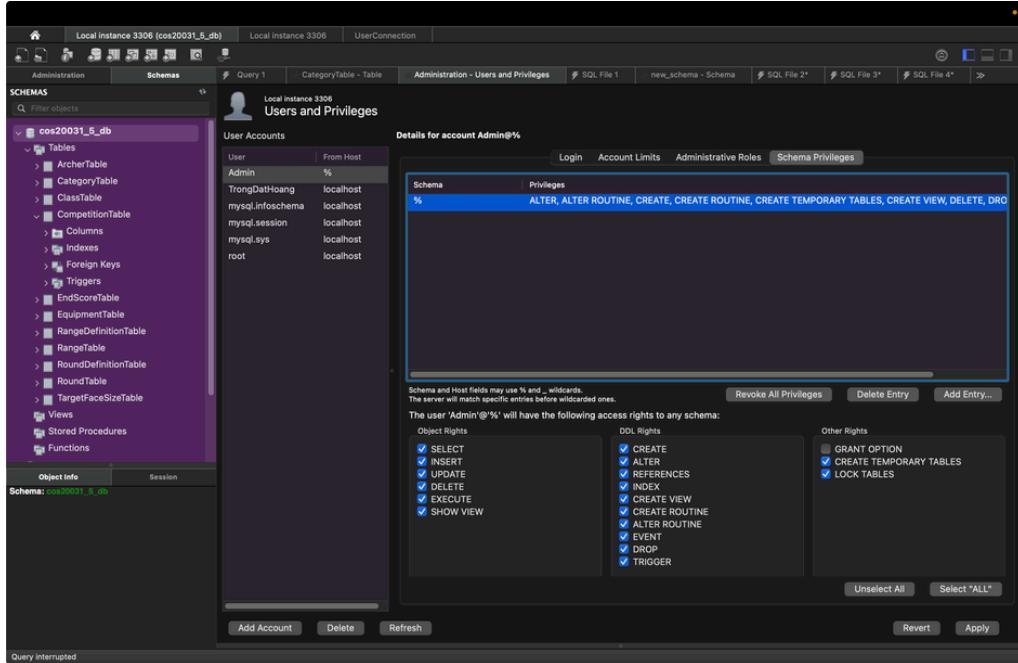


Figure 2: Configure Admin's Schema Privileges.

Restricted User Role: TrongDatHoang

A new user named 'TrongDatHoang' has been added to the database, with privileges appropriate to their role. TrongDatHoang's access is generally limited to data viewing, giving a secure method of analysing and retrieving information without the risk of causing database changes. TrongDatHoang has the following particular privileges:

SELECT Queries: The user can perform SELECT queries to examine and analyse data from the tables to which they have access.

Limited Schema Access: TrongDatHoang has restricted schema access, which means that their actions are limited to data retrieval and cannot alter the database structure or contents. This strictly regulated access allows TrongDatHoang to accomplish their tasks while maintaining the database's overall security and integrity.

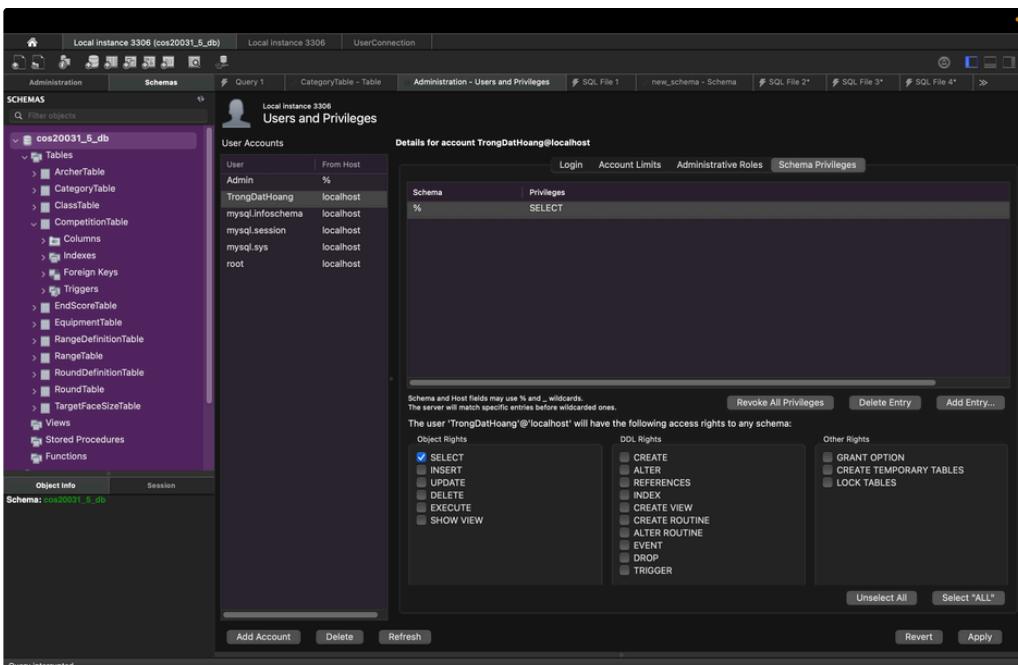


Figure 3: Configure TrongDatHoang user's Schema Privileges.

Summary:

By effectively defining and implementing these roles and privileges, the database system ensures that operations run efficiently and securely. The administrator may maintain overall control and security, allowing users like TrongDatHoang to do their responsibilities efficiently despite their limited access, resulting in a balanced and well-managed database environment.

Team Reflection (4Ls Retrospective)

📝 Overview

Our initiative aims to create a comprehensive database that will monitor archery performance for both individual competitors and archers. This solution will make it easier to run competitions and score accurately while giving everyone access to meaningful data analysis. Through this project, our team was able to learn the value of thorough initial designs, effective teamwork, and effective communication. We also acquired fresh skills in database management and project planning. What propels our group to success is the satisfaction of overcoming challenges as a team and our shared goal of creating a practical, efficient solution for archery score tracking.

Team	Five Guys
Team members	@XUAN TUAN MINH NGUYEN @MAI AN NGUYEN @TRONG DAT HOANG @Phuong Doanh HA @VINH LE
Date	4 Mar 2024 to 20 May 2024
Retrospective period	<ul style="list-style-type: none">Q1: Initial ER diagram and reviews (Weeks 3-5)Q2: Physical database creation and data documentation (Weeks 6-7)Q3: Use cases and performance optimisation (Weeks 8-9)Q4: Major-specific work and team reflection (Weeks 10-12)

💡 4Ls retrospective

Milestones	Loved	Longed for	Loathed	Learned
Q1: Initial ER Diagram and Reviews	Learning to use new diagramming tools	Clearer guidelines for ERD design	Ambiguities in defining relationships	Addressing and resolving ambiguities early
	Clearer guidelines for ERD design	Ambiguities in defining relationships	Addressing and resolving ambiguities early	Addressing and resolving ambiguities early
Q2: Physical Database Creation and Data Documentation	Hands-on experience with SQL and database management systems, Creating comprehensive data documentation	More detailed data documentation templates, Better tools for data migration	Challenges in data documentation, Time-consuming data migration processes	Importance of thorough and accurate documentation, Efficient data migration strategies
Q3: Use Cases and Performance Optimization	Applying use cases to real-world scenarios, Learning advanced SQL queries for optimization	More examples of use cases, Access to advanced optimization tools	Performance bottlenecks, Complexity in optimizing queries	Effective strategies for optimizing database performance, Identifying and addressing bottlenecks early

Q4: Major-specific Work and Team Reflection	Collaboration on major-specific projects, Effective teamwork and communication	More structured team reflection sessions, Additional resources for major-specific challenges	Coordination issues in team projects, Differences in work styles	Value of effective teamwork and communication, Benefits of structured reflection and feedback sessions
--	--	--	--	--

⚡ Action plan

Action	Owner	Due date	Action items
Make clear ERD design rules	@XUAN TUAN MINH NGUYEN	17 May 2024	Find best practices, write down the rules, and share with the team
Clear up relationship questions	@TRONG DAT HOANG	17 May 2024	Hold a team meeting to discuss and clear up relationship definitions
Create data documentation templates	@VINH LE	17 May 2024	Make templates, review with the team, and finalize
Improve database performance	@Phuong Doanh HA	17 May 2024	Find performance problems, fix them, and test
Hold team reflection meetings	@MAI AN NGUYEN	21 May 2024	Schedule meetings, prepare questions, and lead discussions
Improve team project coordination	@XUAN TUAN MINH NGUYEN	23 May 2024	Use project management tools, train the team, and track progress
Address issues early	All team members @Phuong Doanh HA @MAI AN NGUYEN @TRONG DAT HOANG @XUAN TUAN MINH NGUYEN @VINH LE	At the end of each week	Regularly review and update documents, keep communication open

Performance (indexes)

Stakeholders	Use Case	Scenario	Need for index	Query
Archer	Score Entry and Update Author: @TRONG DAT HOANG	The archer wants to enter scores via a handheld device immediately after the rounds just end to ensure accuracy and timeliness. They have to be able to record a date and time, round, equipment (some archers shoot several types of equipment on different days)	No. Single insert operation does not necessarily need indexing.	
	Retrieve and Sort Past Results Author: @MAI AN NGUYEN	To monitor progress in performance, an archer must be able to look up, filter and sort previous scores according to round type and date. Archer's best score for a round is determined if required.	Yes. Can improve the performance of this query by indexing the 'ArcherID', 'ShootDate' and 'ShootTime' column.	<pre> 1 -- Create Index 2 CREATE INDEX idx_archer_shoot_date_time 3 CREATE INDEX idx_end_id ON EndScoreTable 4 5 -- Test Index 6 EXPLAIN SELECT r.ShootDate, r.ShootTime, 7 FROM RoundTable r 8 JOIN EndScoreTable e ON r.EndID = e.EndID 9 WHERE r.ArcherID = 87 10 ORDER BY r.ShootDate DESC, r.ShootTime </pre>
	Round Definitions and Details (Including Equivalent Rounds) Author: @VNH LE	Archers need to be able to look up the requirement for thorough definitions of each round, including equivalent rounds based on current standards.	No. This query use primary key as main filter, which is indexed.	
	Competition lookup Author: @XUAN TUAN MINH NGUYEN	Archers could look up competition results and see how everyone has placed with their scores. Club's best score for a round and the archer who shot the round is determined if called.	Yes. Create index 'CompetitionID' in 'CompetitionTable' and 'ArcherID' in 'ArcherTable' can improve the performance.	<pre> 1 -- Create index 2 CREATE INDEX idx_competition_id ON CompetitionTable 3 CREATE INDEX idx_archer_id ON ArcherTable 4 CREATE INDEX idx_end_id ON EndScoreTable 5 CREATE INDEX idx_competition_id_round ON CompetitionTable 6 7 -- Check index 8 EXPLAIN SELECT a.ArcherFirstName, a.ArcherLastName, a.ArcherID, e.Score 9 FROM RoundTable r 10 JOIN ArcherTable a ON r.ArcherID = a.ArcherID 11 JOIN EndScoreTable e ON r.EndID = e.EndID </pre>

				<pre> 12 JOIN CompetitionTable c ON r.CompetitionID = c.CompetitionID 13 WHERE c.CompetitionID = 41 14 ORDER BY e.TotalEndScore DESC; </pre>
Recorder	Archers, Rounds, Competition, Entry Author: @XUAN TUAN MINH NGUYEN	The recorder could enter new archers, new rounds, and new competition. Some competitions have to be able to be identified as part of a club championship.	No. Single insert operation does not necessarily need indexing.	
Recorders have detailed score-recording capabilities · Author: @Phuong Doanh HA	To ensure proper records, the recorder must precisely submit detailed scores, including per-arrow and end statistics. Each end has to be identified as to its position in the round score (the range and the end order in that range). Arrows are recorded highest to lowest arrow score within an end. Some of the scores have to be able to be linked to a competition.	No. Single insert operation does not necessarily need indexing.		
Class (divisions) and equipment definition lookup Author: @MAI AN NGUYEN	The category can be identified in the absence of user input as there is data needed to identify the archer's division (age, gender) and a definition of the default equipment to link to.	Yes. Create index for 'AgeGroup' and 'Gender' in 'ClassTable' could improve the performance.	<pre> 1 -- Create index 2 CREATE INDEX idx_age_gender ON ClassTable(c.AgeGroup, c.Gender) 3 CREATE INDEX idx_class_id ON CategoryTable(c.ClassID) 4 CREATE INDEX idx_equipment_id ON EquipmentTable(e.EquipmentID) 5 -- Check index 6 EXPLAIN SELECT c.ClassID, c.AgeGroup, c.Gender 7 FROM ClassTable c 8 JOIN CategoryTable cat ON c.ClassID = cat.ClassID 9 JOIN EquipmentTable e ON cat.EquipmentID = e.EquipmentID 10 WHERE c.AgeGroup = 'Under 18' AND c.Gender = 'Male' </pre>	
Validation of historical data and past competition. Author: @TRONG DAT HOANG	The recorder necessitates that the system keep track of previous scores and competitions, particularly when round definitions change.	Yes. Create index for 'CompetitionID', 'ShootDate' and 'ShootTime' in 'RoundTable' could improve the performance.	<pre> 1 -- Create index 2 CREATE INDEX idx_competition_shoot_date ON RoundTable(r.CompetitionID, r.ShootDate, r.ShootTime) 3 CREATE INDEX idx_end_id ON EndScoreTable(e.EndID) 4 CREATE INDEX idx_round_definition_id ON RoundDefinitionTable(rd.RoundDefinitionID) 5 -- Check index 6 EXPLAIN SELECT r.ShootDate, r.ShootTime, r.RoundDefinitionID 7 FROM RoundTable r 8 JOIN EndScoreTable e ON r.EndID = e.EndID 9 JOIN RoundDefinitionTable rd ON r.RoundDefinitionID = rd.RoundDefinitionID 10 WHERE r.CompetitionID = 38 11 ORDER BY r.ShootDate DESC; </pre>	