

Sure, I'd be happy to help expand on the slide headlines you've provided. Here's a more detailed version:

Slide 1: Introduction Title: Overview of Proposed Application for Internal Ticket Management Content: "In an effort to streamline our internal processes and improve efficiency, we are proposing a shift from the current Excel-based system to a robust application specifically designed for internal ticket management. This application aims to simplify the process, making it a seamless part of your day-to-day operations."

Slide 2: Key Features Title: Ticket Management Content: "Our proposed application will feature an organized and user-friendly ticket management system. This will ensure that no ticket is lost or overlooked, improving our response times and customer satisfaction."

Slide 3: Key Features Title: Delivery Tracking Content: "The application will provide real-time updates on ticket status, allowing you to track progress and anticipate delivery times. This feature aims to keep everyone informed and ensure transparency in our processes."

Slide 4: Key Features Title: Team Activity Dashboard Content: "A dedicated dashboard will provide a snapshot of team activities, fostering collaboration and transparency. This feature will allow managers to monitor progress and allocate resources effectively."

Slide 5: Key Features Title: Jira Integration Content: "Our application will seamlessly integrate with Jira, a popular project management tool. This will allow for efficient management of projects and tasks, further streamlining our processes."

Slide 6: Key Features Title: Leave Management Content: "The application will also simplify the process of leave requests and approvals. An easy-to-use interface will make leave management hassle-free for both employees and managers."

Slide 7: Key Features Title: Performance Metrics Content: "Our application will provide comprehensive performance metrics, allowing us to measure success and identify areas for improvement. This data-driven approach will help us continually refine our processes and strategies."

Slide 8: Work Progress Title: Backend Development Content: "Our team has been focusing on developing a strong foundation for our application. This includes implementing user authentication, user management, and enhanced security features."

Slide 9: Work Progress Title: Frontend Development Content: "On the frontend, we are working on designing an intuitive and user-friendly interface. Our goal is to ensure compatibility across different devices and provide a seamless user experience."

Slide 10: Work Progress Title: Database Setup Content: "We are setting up a database designed to efficiently store and retrieve data. We are implementing data validation and constraints to maintain data integrity and consistency."

Slide 11: Work Progress Title: Design Work Content: "Our design team is working on developing diagrams and refining the UI/UX design based on user feedback. We believe that good design is integral to a great user experience."

Slide 12: Conclusion Title: The Road Ahead: Delivering a Successful Internal Ticket Management Solution Content: "We are excited about the progress we've made and are looking forward to the future. With our user-centric design, technical expertise, and collaborative approach, we are confident in our ability to deliver a successful internal ticket management solution."

I hope this helps! Let me know if you need further assistance.

For user authentication, several security measures are typically put in place to ensure the safety and privacy of user data:

Secure Password Storage: Passwords are never stored in plain text. They are hashed using algorithms like bcrypt or Argon2 before they are stored in the database. This means even if someone gains access to the database, they won’t be able to see the actual passwords.

Salted Hashes: A unique value (or ‘salt’) is added to the password before it’s hashed. This makes it much harder for an attacker to use pre-computed tables (rainbow tables) to attack the hash.

HTTPS: Secure communication over the network is ensured using HTTPS (Hypertext Transfer Protocol Secure), which encrypts the data sent between the user and the server.

JWT (JSON Web Tokens): JWTs are used for maintaining sessions. They allow the server to authenticate the user on every request without having to log in multiple times.

Rate Limiting: This prevents brute force attacks by limiting the number of login attempts from a single IP address.

Two-Factor Authentication (2FA): 2FA adds an extra layer of security by requiring users to verify their identity using a second method, in addition to username and password.

Account Lockouts: After a certain number of failed login attempts, the account is locked for a period of time. This helps prevent brute force attacks.

Password Complexity Requirements: Enforcing a certain level of password complexity (length, mix of characters, etc.) helps protect against brute force attacks.

Remember, the exact security measures can vary based on the specific requirements of the project and the current best practices in cybersecurity. It’s always a good idea to consult with a security expert or team when building a secure authentication system.

* How are session tokens managed and invalidated?
* What measures are in place to prevent token leakage or theft?
* Is there any logging of authentication events for auditing purposes?

Sure, here’s a brief text in French for your slides:

Slide 1: Introduction Titre: Embarquement pour un voyage: Une nouvelle ère de gestion des tickets internes Contenu: “Imaginez un monde où la gestion des tickets internes n’est plus une corvée, mais une partie intégrante de votre journée. Nous sommes en mission pour remplacer le système actuel basé sur Excel par une application innovante qui révolutionnera notre approche de la gestion des tickets.”

Slide 2: La Vision Titre: Notre vision: Simplifier les processus et améliorer la collaboration Contenu: “Notre vision est de créer une application qui non seulement simplifie le processus de gestion des tickets, mais favorise également la collaboration entre les membres de l’équipe. Nous envisageons un outil qui fournit des informations en temps réel, rendant chaque ticket, chaque tâche et chaque projet transparents et gérables.”

Slide 3: Caractéristiques Clés Titre: Les éléments constitutifs: Caractéristiques clés de notre application proposée Contenu: “Gestion des tickets: Dites adieu aux tickets perdus et bonjour à des informations organisées et facilement accessibles. Suivi des livraisons: Des mises à jour en temps réel à portée de main, pour que vous soyez toujours au courant. Tableau de bord de l’activité de l’équipe: Un aperçu de l’avancement de votre équipe, favorisant la transparence et la collaboration. Intégration avec Jira: Intégration transparente avec Jira, rendant la gestion de vos projets plus facile que jamais. Gestion des congés: Simplifiez les demandes et approbations de congés avec une interface facile à utiliser. Mesure de performance: Mesurez le succès et identifiez les domaines d’amélioration avec des mesures de performance complètes.”

Slide 4: Progrès du Travail Titre: Derrière les scènes: Notre progrès jusqu’à présent Contenu: “Nous avons travaillé dur pour transformer notre vision en réalité. Du développement backend au design frontend, nous faisons des progrès dans divers aspects du processus de développement. Notre objectif est de mettre en œuvre des principes de design centrés sur l’utilisateur et d’assurer une intégration transparente avec les systèmes existants.”

Slide 5: Développement Backend Titre: L’épine dorsale: Progrès du développement backend Contenu: “L’authentification des utilisateurs, la gestion des utilisateurs et les fonctionnalités de sécurité améliorées ne sont que quelques-uns des domaines sur lesquels nous nous sommes concentrés dans notre développement backend. Nous construisons une base solide pour notre application.”

Slide 6: Développement Frontend Titre: Le visage: Progrès du développement frontend Contenu: “De la conception de l’interface de connexion à l’assurance de la compatibilité sur différents appareils, notre équipe de développement frontend est dédiée à la création d’une expérience intuitive et conviviale pour l’utilisateur.”

Slide 7: Configuration de la Base de Données Titre: Le cerveau: Progrès de la configuration de la base de données Contenu: “Notre configuration de base de données est conçue pour stocker et récupérer efficacement les données. Nous mettons en œuvre la validation des données et les contraintes pour maintenir l’intégrité et la cohérence des données.”

Slide 8: Travail de Conception Titre: L’esthétique: Progrès du travail de conception Contenu: “Nous croyons que le bon design est intégral à une grande expérience utilisateur. C’est pourquoi nous développons des diagrammes, affinons le design UI/UX en fonction des retours des utilisateurs, et collaborons avec les parties prenantes.”

Slide 9: Conclusion Titre: La route à venir: Livrer une solution réussie de gestion des tickets internes Contenu: “Nous sommes excités par les progrès que nous avons réalisés et encore plus excités par ce qui est à venir. Avec un design centré sur l’utilisateur, une expertise technique et une collaboration, nous sommes confiants dans notre capacité à livrer une solution réussie de gestion des tickets internes.”

I hope this helps! Let me know if you need further assistance.

1. Presentation   
   Introduction:
   1. Overview of Proposed Application for Internal Ticket Management
2. Key Features:
   1. Ticket Management
   2. Delivery Tracking
   3. Team Activity Dashboard
   4. Jira Integration
   5. Leave Management
   6. Performance Metrics
3. Work Progress:
   1. Backend Development
   2. Frontend Development
   3. Database Setup
   4. Design Work
4. **Summary (Introduction):**
5. "In response to the need for efficient internal ticket management, a proposal was made to develop an application aimed at replacing the current Excel-based system. This presentation outlines the key features of the proposed application, including ticket management, delivery tracking, a team activity dashboard, Jira integration, leave management, and performance metrics. The application aims to streamline the ticket management process, providing real-time insights and enhancing collaboration among team members. Progress has been made in various aspects of the development process, with a focus on implementing user-centric design principles and ensuring seamless integration with existing systems."
6. **Work Progress and Key Features for Ongoing Work:**
7. **Backend Development:**
   1. Implementation of user authentication using JWT tokens to ensure secure access to the application.
   2. Creation of user management controller in Spring Boot backend for efficient user administration and role assignment.
   3. Integration of Spring Security for enhanced security features and access control.
8. **Frontend Development:**
   1. Design and development of frontend interface for login and user management using Angular framework.
   2. Implementation of responsive design principles to ensure compatibility across different devices and screen sizes.
   3. Integration of Angular Material components for consistent UI design and user experience.
9. **Database Setup:**
   1. Creation of databases for users, roles, and tasks management using MySQL or another suitable database management system.
   2. Implementation of data validation and constraints to maintain data integrity and consistency.
   3. Development of database schemas optimized for efficient data storage and retrieval.
10. **Design Work:**
    1. Development of diagrams for visualizing the application structure, including class diagrams, use case diagrams, and sequence diagrams.
    2. Iterative refinement of UI/UX design based on user feedback and usability testing.
    3. Collaboration with stakeholders to ensure alignment with business requirements and project objectives.
11. This expanded context provides more detailed information about the proposed application, its key features, and the ongoing work progress across various development areas. It highlights the importance of user-centric design, technical expertise, and collaboration in delivering a successful internal ticket management solution.

First, let's review the current schema design:

* **RequestFlux** entity has a one-to-one relationship with **RequestFluxDetail** and **Ip**, and a one-to-many relationship with **Port**.
* **RequestFluxDetail** and **Ip** are both entities with their own tables.
* **Port** is not an entity, but a list of strings in the **RequestFlux** table.

Here are some suggestions to upgrade the schema design:

1. **Create a separate Port entity:** Create a **Port** entity with its own table, and establish a one-to-many relationship between **RequestFlux** and **Port**. This will allow you to query and manipulate the **Port** values separately from the **RequestFlux** entity.
2. **Add a PortType enum:** Add a **PortType** enum to the **Port** entity to distinguish between different types of ports (e.g., TCP, UDP, ICMP). This will allow you to add additional attributes and behavior to specific types of ports.
3. **Add a Network entity:** Create a **Network** entity with its own table, and establish a many-to-one relationship between **Ip** and **Network**. This will allow you to group IP addresses by network and add additional attributes and behavior to specific networks.
4. **Add a Service entity:** Create a **Service** entity with its own table, and establish a many-to-one relationship between **RequestFlux** and **Service**. This will allow you to group related **RequestFlux** entities by service and add additional attributes and behavior to specific services.
5. **Add a Security entity:** Create a **Security** entity with its own table, and establish a many-to-one relationship between **RequestFlux** and **Security**. This will allow you to add security-related attributes and behavior to specific **RequestFlux** entities.

These are just a few suggestions to upgrade the schema design. The specific upgrades you choose will depend on the requirements and design goals of your application.

Regarding the other types of schemas you mentioned, I can also provide more details and suggestions if you provide more context. For example, if you are interested in network schemas, I can explain the different types of VLANs, subnets, firewalls, and other network components, and suggest best practices for designing and implementing network schemas. If you are interested in software architecture schemas, I can explain the different types of microservices architectures, and suggest best practices for designing and implementing microservices. If you are interested in security schemas, I can explain the different types of authentication, authorization, and encryption mechanisms, and suggest best practices for securing your application.

1. implement a RESTful controller for the **Task** entity to handle CRUD operations. You can follow the example I provided earlier for the **Tasks** entity.
2. Create a user interface that allows users to create, view, update, and delete tasks. This can be done using a frontend framework like Angular or React.
3. Implement a feature for tracking task delivery status. You can use status markers to indicate the current state of tasks (e.g., in progress, pending, completed, etc.).
4. Create a dashboard that displays an overview of team activity, including the number of tasks completed, resolution times, etc. You can use charts and graphs to visualize the data.
5. Integrate your ticketing system with Jira to synchronize tasks and updates in real-time. You can use Jira's API to retrieve ticket data and map Jira statuses to your application's statuses.
6. Implement a feature for managing user vacations. This can include a calendar that displays planned vacation days and holidays.
7. Collect and track performance metrics related to tasks, such as resolution times, resolution rates, etc. You can display these metrics in dashboards and reports for in-depth analysis.
8. Implement user authentication using JWT to secure access to the application.
9. Use frameworks and libraries like Angular Material to design a consistent and responsive user interface.
10. Collaborate with stakeholders to define requiremsts, design the application architecture, and validate implemented features.
11. **Entities**: These are the Java classes that represent the tables in your database. You have **RequestFlux**, **RequestFluxDetail**, **Ip**, and **Port**.
12. **Repositories**: These are the interfaces that provide data access to the entities. You have **RequestFluxRepository**, **RequestFluxDetailRepository**, **IpRepository**, and **PortRepository**.
13. **Services**: These are the classes that encapsulate the business logic of your application. You have **RequestFluxService**, **RequestFluxDetailService**, **IpService**, and **PortService**.

You may also need to consider the following components:

1. **Controllers**: These are the classes that handle HTTP requests and responses. You'll need controllers to expose RESTful APIs for your application.
2. **DTOs (Data Transfer Objects)**: These are the classes that represent the data that's transferred between the client and server. You may need DTOs to convert between your entity objects and the data that's sent over the wire.
3. **Validators**: These are the classes that validate the data that's sent to your application. You may need validators to ensure that the data is correct and consistent.
4. **Security**: Depending on your application's requirements, you may need to implement security features such as authentication and authorization