Here’s the **comprehensive sentence-by-sentence study note breakdown** of your *“Ticketing System Notes”* document, formatted professionally for Word, numbered for clarity, and ensuring no critical information is omitted.

**Ticketing System – Study Notes**

1. **Purpose of Ticketing Systems**
   * Used to manage **requests, incidents, and problems** submitted by users.
   * Systems vary in interface and features, but core functions remain consistent.
   * Examples: Freshdesk, osTicket, BMC Remedy, Zendesk, Intercom.
   * For exams, focus on understanding **how to use and interpret ticket data**, not on specific brands.
2. **User Information in Tickets (1)**
   * Tickets are linked to a **user account/record** containing identifiers such as **phone number, email, name, employee ID.**
     + **User has to be associated with the ticket**
     + **This maybe based on their phone, email, name, and id.**
     + **Or some other piece of identifying information.**
   * Whenever a new ticket is created it’s going to be tied back to that user and that way you can see a history of everything that user has ever had issues with.
     + This is good to better support the user.
   * Capturing user history helps in **personalized and efficient support**.
   * Example: In Dion Training you can see which platform their students is using (Dion Training, Udemy, LinkedIn) from which the user submitted the ticket.
   * Build a **comprehensive user profile** over time.
     + Gathering more information about that users will start building up a larger user profile.
3. **Device Information in Tickets (2)**
   * Especially important in large IT environments.
   * Gather data via **asset ID/code**, device type (desktop, laptop, tablet, smartphone), and OS details.
     + Once you have the asset id/code
       1. Identify which type of computer.
       2. Whether it’s a desktop, a laptop,, a smartphone, or a tablet, and which OS was being run on that system.
   * Device information links recurring hardware issues to **specific devices**.
   * Helps distinguish between **isolated incidents** and **wider problems**.
4. **Incident vs. Problem**
   * **Incident** – One-time, isolated issue affecting a single user/device.
   * **Problems** – Recurring issue across multiple users/devices with similar symptoms.
   * Device and user info helps in accurate classification.
5. **Problem Description in Tickets**
   * Must be **detailed and specific** for effective troubleshooting.
   * Bad Example: "Cannot access internet" is vague.
   * Good Example: "Facebook.com returns 'site not found' error" is actionable.
     + This gives more details and a technician can better understand the problem and is able to troubleshoot you.
   * Anytime you’re gathering information about a problem or an issue you should:
     + Gather as much detail as possible.
     + Ask clarifying questions.
     + This is to make sure to get a good assessment of what the issue is.
   * Include cause, context, symptoms, and observed effects.
6. **Ticket Categorization**
   * Categories improve **routing to the correct support agent/team**.
   * These categories can be based on different cases and departments depending on how your organization likes to work.
     + Examples: Software, hardware, networking, mobile devices, billing, exam vouchers.
     + Dion Training Category examples:
       1. Udemy
       2. Website
       3. Billing
       4. Exam Vouchers
     + Groups can allow users to get in contact with the right agents who are experts in those areas to be able to provide the users with better support.
   * Can also break categories or classify by **ticket type**:
     + **Request** – Asking for a new feature, function, or resource (e.g., account setup, equipment request).
     + **Incident** – Unexpected issue/error affecting a single user/system.
       1. **Ex**: A single user Logs Into a website and your account is not working because it was locked out.
     + **Problem** – Multiple incidents with common cause.
       1. **Ex**: Multiple Users Log Into a website and tell several users that it was locked.
          1. **That means it is not an issue with the user but it’s an issue with the servers**.
          2. **Problems are generally going to be up to a helpdesk manager or supervisor or to a higher-level tier to figure out why it’s happening**.
     + **Change** – Complex modification requiring formal change management.
7. **Severity Levels/Priority**
   * Common levels: Low, Medium, High, Urgent, Critical, Major, and Minor.
   * Assignment can be manual or automated via AI/machine learning.
   * Example: “Refund” keyword auto-assigns urgent priority.
     + E-commerce companies that have their priority set if the word “refund” exists inside the subject line of the email or ticket.
       1. This will be marked as urgent.
       2. Therefor it gets the highest priority.
       3. This is because this they have an upset customer and they want to solve that quickly before that customer leaves a bad review.
   * Alternative Example:
   * Technicians should work with the **highest priority first** and adjust workflow as new urgent tickets arrive.
8. **Escalation Procedures**
   * Support is structured in **levels/tiers**:
     + **Tier 0** – Self-service via portals/knowledge base.
     + **Tier 1** – Basic troubleshooting and quick fixes.
     + **Tier 2** – Senior techs/system admins for advanced fixes.
     + **Tier 3** – Engineers, developers, high-level managers.
   * Most tickets (~70–80%) resolved at Tier 1; Tier 3 handles 1–3%.
   * **Shifting left** – Moving resolution to lower tiers reduces cost/time.
9. **Ticket Documentation Best Practices**
   * Include:
     + **Problem Description** – Initial issue as reported.
     + **Progress Notes** – Actions taken and findings during troubleshooting.
     + **Problem Resolution** – Final fix and how it was achieved.
   * Clear, detailed notes ensure continuity across shifts and facilitate faster future resolutions.
   * Example: Documenting APIPA address troubleshooting (IP config commands, DHCP checks) for reference in similar future cases.
10. **Final Key Principles**
    * Accurate categorization, priority assignment, escalation, and documentation are critical for efficient IT support.
    * Well-documented tickets improve **team collaboration**, reduce resolution time, and enhance customer satisfaction.