Software Development Life Cycle (SDLC) Documentation

Project Name: Help Desk Ticketing System Prototype

1. **Planning**
   - Understanding the client's needs and establishing the projects scope are part of the first step.
   - In our situation, acquiring client needs and comprehending the parameters of the Help Desk ticketing system prototype are both part of the planning step.
2. **Requirements Analysis**
   - This step involves a thorough examination of the clients needs and operational procedures. In our situation, the thorough customer requirements supplied at the start of the project serve as a representation of the requirements analysis.
3. **Solution Design**
   - Plan how the various components of the system will interact.
   - Create a high-level design for the Ticket class and its methods.
4. **Detailed Design**
   - Specify the data structures and algorithms needed for ticket management.
   - Consider how to handle ticket statistics.
5. **Construction**
   - This is where the actual coding takes place.
   - Our Python code implementation falls under this stage, where we write the code for the Help Desk ticketing system.
   - Implement the Ticket class and its methods as per the requirements.
   - Ensure that the code is well-structured and follows best practices.
6. **System Testing**

   Before the developed system can be delivered to the client, it must go through rigorus testing to verify that the specified requirement has been fulfilled.
   - After construction, its crucial to thoroughly test the system.
   - Conduct thorough testing of the ticketing system to identify and resolve any bugs or issues.
   - Test various scenarios, Including ticket submission, response handling, reopening tickets, and statistics calculations.
   - Verify that the system meets the client's requirements.
7. **Deployment**
   - Deploy the ticketing system to the intended environment.
   - Ensure that all necessary configurations are in place.
   - Prepare for end-user access.
8. **Operation**
   - This stage involves operating and maintaining the system in a live environment.
   - For our project, this might be as simple as running the phyton script on a server or local machine.
9. **Maintenance**
   - This is where you correct errors or develop new functionality.
   - In our case, its important to keep the code up-to-date if any changes to requirements or issues arise.