From the project description doc file we’ve seen what a ticket system entails and some of the journeys you’ll face to complete your final rpoject!

Again, regarding the definition and working of a trouble ticket system for any help desk, note the following:

A trouble ticket (sometimes called a *trouble report*) is a mechanism used in an organization to track the detection, reporting, and resolution of some type of problem. Trouble ticketing systems originated in manufacturing as a paper-based reporting system; now most are Web-based and associated with customer relationship management ([CRM](http://searchcrm.techtarget.com/definition/CRM)) environments, such as [call center](http://searchcrm.techtarget.com/definition/call-center)s or [e-business](http://searchcio.techtarget.com/definition/e-business) Web sites, or with high-level technology environments such as network operations centers (NOCs).

Reference: <http://searchcrm.techtarget.com/definition/trouble-ticket>

So basically for the IT help desk ticket system, your looking at creating a trouble ticket system/tracker that helps any department track items that represent trouble to some user.

Ex. I am an accountant for the company and I call the help desk to state I am having trouble with my PC. So the Help Desk creates a trouble ticket noting the problem. And until the problem gets fixed, the ticket really remains active.

Think from now!

**Exercise: Complete a sample prototype for class discussion (no need to turn in…just be ready for an open discussion)**

THE DB

So think of creating a dbase table(s) that will help track a problem.

Ex. Users table, Ticket Table, Ticket History Table, Dept\_Loc…

Think of needed *fields* tailor made for each table that will help store data needed for reporting tickets that maybe active or even inactive.

What fields are needed ALWAYS depend on the reports that need to be generated. Note any tables that you may wish to link to one another.

THE UI

What will be needed for various Windows

Ex. login screen, report screens, data entry screens…

Be specific—plot or draw out visuals even a storyboard for navigations

What controls or objects will be needed for each UI appearance.

OTHER

Think of the full process of your app. The workflow from user entry to the backend processing!

Security needs (SQL injection preventions, login attempts, password hashing)