Now that you have reached the last chapter, it is time for us to tie it all together. During the book we have tried to set a course for how to bring an enterprise application safely to shore. We have shown a lot of tips and tricks for how you should design your products. As we walk through our demo application, you’ll see that we will not be able to apply all techniques to the application. Neither would we be able to in a real-world application. It all varies depending on the circumstances for each project, so you should carefully consider what best suits your application and apply those principles as needed. We are going to design and build an enterprise time reporting application. It will be used for employees to report the amount of time they have spent on projects and tasks during the week. Let us begin by looking at some assumptions about this application.

Application Assumptions

We have made some assumptions regarding the fictional company we are creating this example application for in Table 9-1.

<table>
<thead>
<tr>
<th>Table 9-1. The R &amp; R Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company name</td>
</tr>
<tr>
<td>Number of employees</td>
</tr>
<tr>
<td>Number of countries with branch offices</td>
</tr>
</tbody>
</table>

Every Friday by 5:00 p.m., R & R staff have to report the time they have spent at work during the week. This means that from around 4:00 to 5:30 p.m. the time reporting application will be heavily used. Luckily, all 35,000 employees do not report to the same application; they report to one application in each country R & R has offices in. The load can still be considerably high, especially in a country like the U.S., where more than 9000 R & R employees work. The design of the application must be able to handle this load.
Application Requirements

Management has a number of requirements for the new application:

- Employees must be able to create and save a weekly report. They should then be able to reopen it and continue adding hours and projects to it, until they submit it for approval to their immediate manager.

- Once submitted, employees cannot recall the report in order to make changes. In case something needs to be changed, an employee needs to contact his or her manager. The manager can then reject the report, and the employee can make the necessary changes after that.

- A time report can be rejected on the row level. That is, a manager can reject only a part of the time report so the employee does not have to go through the whole report to find the error.

- A root project can have any number of subprojects.

- There are no limits to how many levels of subprojects can exist under one root project.

- Different employees who have access to the same root project can be shown different subsets of the subprojects. (A salesperson does not have the same subprojects as a project manager, for example.)

- Each employee should have access to only the specific root projects and subprojects that are relevant.

- An administrator can tie projects to an employee.

- The normal working hours for an employee are 40 hours per week. The system must check a report so that exactly 40 hours are reported as normal time each week.

- Time above 40 hours must have another time type.

- An administrator can add new time types (such as overtime normal hours, flex time, and so on) for projects.

- The application must handle at least 2000 simultaneous users.

- Users should be placed in Active Directory (AD) or Active Directory Application Mode (ADAM). The choice depends on whether ADAM has been released or not when we start development.