The specification document was not very detailed, so I made some adjustments where appropriate to help build a useful and user friendly (as possible) application. This document attempts to describe all of my high-level design decisions.

* I don’t consider just name and Id number as a reliably unique key. There is a small but definite chance of collisions with that as a non-PK unique key, so I added an **IdNumber** column to the **Person** table.
* The one-to-many relationship with **Person** and **Employee** was a slight challenge as to how to interpret it. I considered that one person, especially a contractor, might well be employed by the same company several times. That addresses the one person to many employee records issue, even if most people only are only ever employed once. I don’t see how else this relationship could be interpreted.
* With the two tables as in the specification, one Person cannot have multiple employees, that would be better implemented as a self-join on a single **Employee** table. Therefore, I have established that one person can have many employments at a company, and that preserves the one-to-many relationship. I have renamed the table and class to **Employment** and added a **Position** column to that table to help discern between one persons many employments. E.g. Joe Schmoe was employed as an intern, then an administrator, and finally a director. Now one record for Joe has three easily distinguishable employments.
* I dislike duplicated and redundant code, so instead of creating an individual view for each action, I created one **Edit** view that is used for the create, edit, and delete actions for Person and Employee.