3. SYSTEM DESIGN

3.1 ER-DIAGRAM FOR COMPANY EMPLOYEE MANAGEMENT SYSTEM.

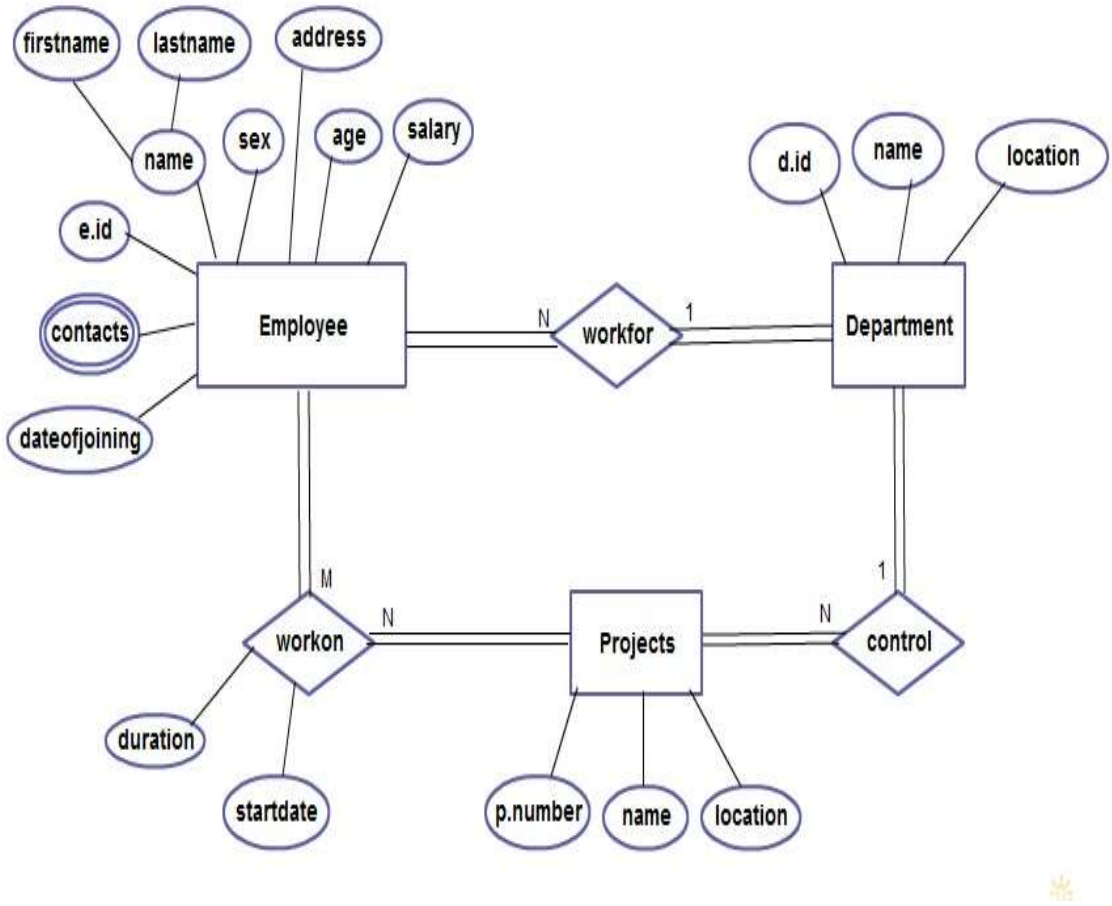

Fig 3.1 Er-Diagram

3.2 SCHEMA DIAGRAM F OR COMPANY EMPLOYMENT MANAGEMENT SYSTEM

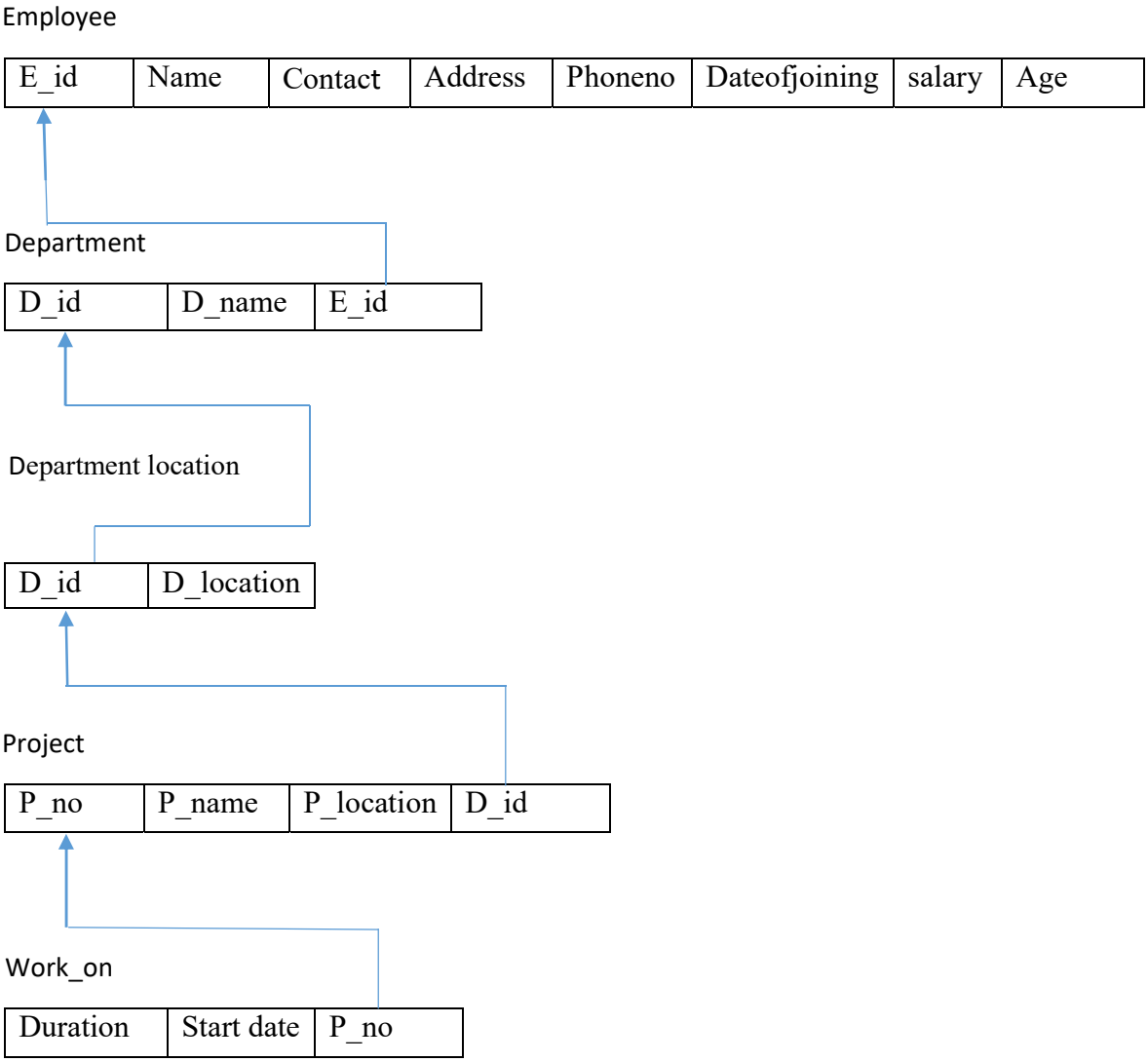

Fig 3.2 Schema Diagram

3.3 Overview of GUI

The graphical user interface (GUI), is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep learning curve of command-line interfaces (CLIs), which require commands to be typed on a computer keyboard. I have used Google chrome as GUI.Windows provides very primitive tools for building user interfaces. The system provides a few basic controls and native window containers, but building a custom user interface is difficult. Since we desired a differentiated aesthetic for Chromium, we have had to build a framework on Windows to accelerate our development of custom UI. This system is called **views**. **views** is a rendering system not unlike that used in WebKit or Gecko to render web pages. The user interface is constructed of a tree of components called *Views*. These Views are responsible for rendering, layout, and event handling. Each View in the tree represents a different component of the UI. An analog is the hierarchical structure of an HTML document.At the root of a View hierarchy is a **Widget**, which is a native window. The native window receives messages from Windows, converts them into something the View hierarchy can understand, and then passes them to the RootView.

Painting and layout are done in a similar way. A View in the View tree has its own bounds (often imbued upon it by its containing View's Layout method), and so when it is asked to Paint, it paints into a canvas clipped to its bounds, with the origin of rendering translated to the View's top left. Rendering for the entire View tree is done into a single canvas set up and owned by the Widget when it receives a Paint message. Rendering itself is done using a combination of Skia and GDI calls — GDI for text and Skia for everything else.Several UI controls in Chromium's UI are not rendered using Views, however. Rather, they are native Windows controls hosted within a special kind of View that knows how to display and size a native widget. These are used for buttons, tables,radio buttons, checkboxes, text fields, and other such controls.

Since they use native controls, these Views are also not especially portable, except perhaps in API.Barring platform-specific rendering code, code that sizes things based on system metrics, and so on, the rest of the View system is not especially *un*portable, since most rendering is done using the cross-platform Skia library. For historical reasons, many of View's functions take

Windows or ATL types, but we have since augmented ui/gfx/ with many platform independent types that we can eventually replace these with.

Some other features of phpmyadmin IDE include-

- Easy and efficient Project Management
- Rapid user Interface Development
- Write bug free code
- Cross Platform support

3.4 Normalization

Database normalization, or simply normalization, is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity. Normalization is also the process of simplifying the design of a database so that it achieves the optimal structure composed of atomic elements. It was first proposed by Edgar F. Codd, as an integral part of a relational model.Normalization involves arranging attributes in relations based on dependencies between attributes, ensuring that the dependencies are properly enforced by database integrity constraints. Normalization is accomplished by applying some formal rules either by a zprocess of synthesis or decomposition. Synthesis creates a normalized database design based on a known set of dependencies. Decomposition takes an existing (insufficiently normalized) database design and improves it based on the known set of dependencies.

A basic objective of the first normal form defined by Codd in 1970 was to permit data to be queried and manipulated using a "universal data sub-language" grounded in first-order logic.In my project normalization can be applied to make the data free of redundancy. As we know there are several anomalies like update, insert and delete which can be removed by applying normalization. So in my project 1NF is applied but for further enhancement 2NF and 3NF can also be implemented.

The objectives of normalization beyond 1NF (first normal form) were stated as follows:

1. To free the collection of relations from undesirable insertion, update and deletion dependencies;
2. To reduce the need for restructuring the collection of relations, as new types of data are introduced, and thus increase the life span of application programs;
3. To make the relational model more informative to users;
4. To make the collection of relations neutral to the query statistics, where these statistics are liable to change as time goes by.