

Database Systems CSC 675-03  
Spring 2019  
Milestone 1:  
The Semantic Model  
04/02/2019

# **CentiPayroll**

Team 5:  
Edwin Menjivar - Team Lead  
Kenneth Surban - Github Master  
Jeffrey Piercy  
Rowvin Dizon  
Andrew Keelin

### **Executive Summary**

Our team will be creating CentiPayroll, which is a payroll management web application. The team's motivation towards developing this application is that we would like to help small businesses by bringing them a web application that is not as expensive as other more complex and more expensive softwares that are in the market. The importance of developing this application is that our team is very familiar with the topic of payroll management, therefore we will be able to create a well developed database system.

CentiPayroll will focus on bringing a web service for small businesses who do not have a digital way of keeping track of their employees worked hours due to economic limitations. The purpose of CentiPayroll is to allow employees to be able to clock in and clock out during their shifts, and employers should be able to look at the hours' report. The way the website will work is by having the employer being able to login as an administrator or as a web time clock app. If the employer logs in as an administrator, they should be able to access all the information regarding their business. The information that they can access will range daily/weekly/monthly reports to being able to add/delete/edit employees. If the employer logs in as a time clock app, the website will display a numeric keypad. Each employee will have a pin number assigned to them, and they will type that pin in the numeric keypad in order to clock in or clock out.

What will be unique and special about CentiPayroll's database is that we will have multiple tables, which will be referenced from different tables. We will do this so that whenever a user tries to access some data, he will not be able to access everything in the database, unless they have specific information, therefore making our application more secure.

## Entities

Company	This is the administrator of the website. A company can add/remove employees, add/edit/remove their shifts, and fetch reports. When a new employee is hired, a company adds their information to the website, and an employee becomes a new entity. The company has full access to everything that has to do with that employee, and can add/edit/remove their shifts. At the end of the week, one company can fetch a report for each employee given any time range.
Employee	Employee is specialized in clocking in and clocking out before and after their daily shifts. When an employee arrives into their job, or when they take a break, the website will display a number pad, in which they will insert their unique pin to clock in. At the end of their shifts, or at the end of their breaks, the employee should use the same number pad displayed in the website to enter their unique pin again, meaning that they would be clocking out.
Shift	A shift is as set of two time punches by an employee. When the employee punches his pin once in the number pad, this is called clocking in, the second time the employee punches his pin, it is called clocking out. Each set of two of this punches is called a shift. A shift will be used to keep track of the total hours worked by each employee for any given time range, by using a report.
Report	A report is a set of shifts in the database. This one will be accessed by the company, whom will set a range to retrieve as many shifts as there are in that range for any number of employees. If the company wants to see the hours worked by an employee in the past week, they will set the time range to cover the entire week, and the screen should show all the shifts for that employee for that given time range, as well as the total hours worked for the entire time week.

### Business rules

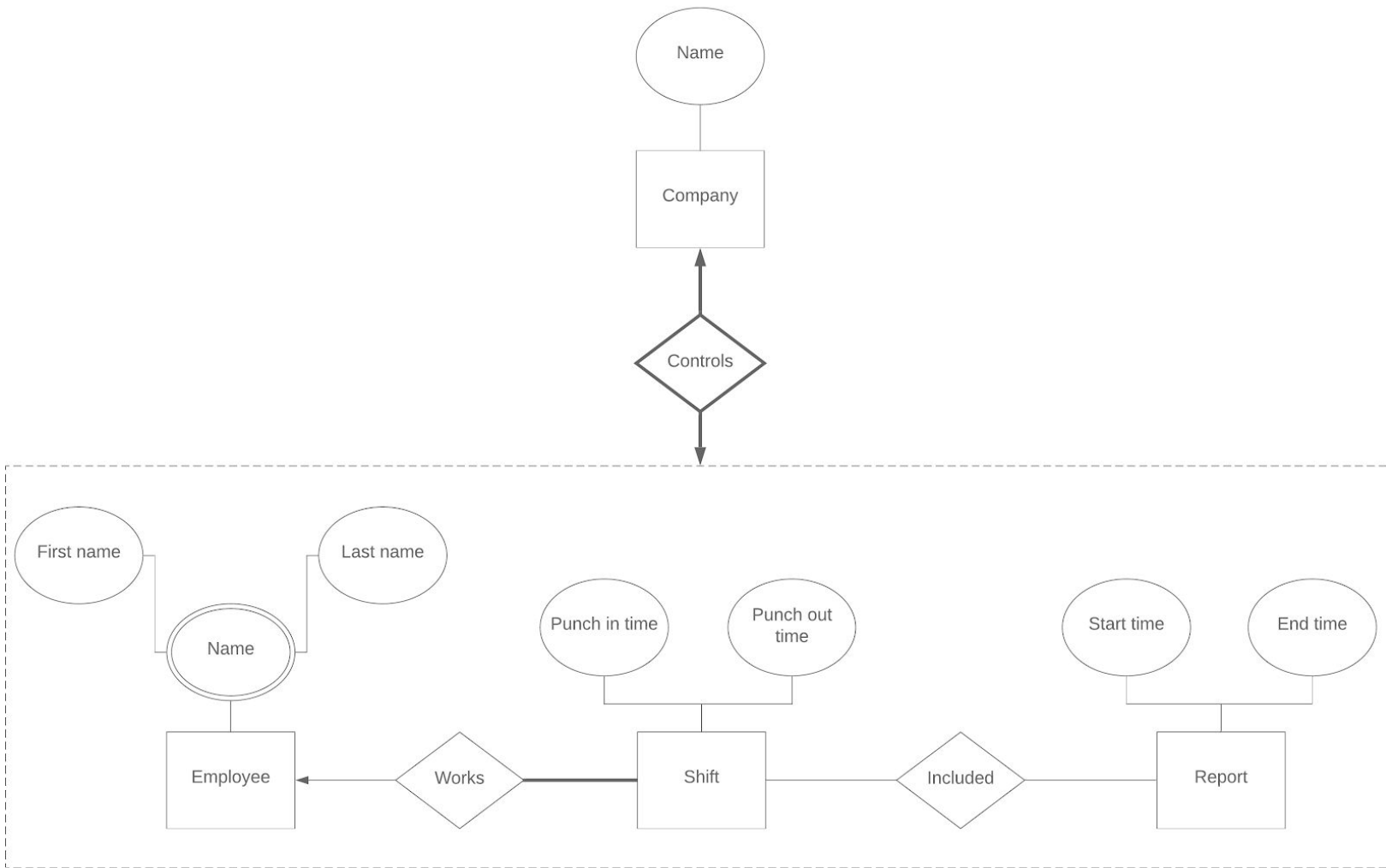
1. Company can add, edit, and remove Employees.
2. Company can access and edit Employees' shifts.
3. Employee works a shift.
4. Reports includes all the shifts in a given time range.
5. A shift must contain two unique time punches, with clock out time initially null.
6. Employees can clock in and out.

### Functional requirements

<b>Entity:</b> Company (strong) <b>Relations (if any):</b> controls <b>Attributes:</b> <ul style="list-style-type: none"><li>• emprid (key)</li><li>• name (composite)</li></ul>
<b>Entity:</b> Employee (strong) <b>Relations (if any):</b> works <b>Attributes:</b> <ul style="list-style-type: none"><li>• empid (key)</li><li>• name (composite)</li><li>• pin (derived)</li></ul>
<b>Entity:</b> Shift (weak) <b>Relations (if any):</b> includes, works <b>Attributes:</b> <ul style="list-style-type: none"><li>• sid (key)</li><li>• time_in</li><li>• time_out</li></ul>
<b>Entity:</b> Report (weak) <b>Relations (if any):</b> includes <b>Attributes:</b> <ul style="list-style-type: none"><li>• rid (key)</li><li>• start_time</li><li>• stop_time</li></ul>

## Entity Relationship Diagrams(ERD)

Group 5 ERD



### ERDs Test

Business Rule #	Entity	Relationship	Type of Relationship	Entity	Pass/Fail	Modify	Notes
1	Company	controls	1 - 1	Employee	pass		
2	Company	controls	1 - 1	Shift	pass		
3	Report	included	M to N	Shift	pass		Will need junction for M - N in DB model
4	Shift				pass		
5	Employee	works	1 - M	Shift	pass		

### NON-functional requirements

#### Security

1. Login shall be required to make changes to the data.
2. Passwords shall be encrypted.
3. The security of database(s) shall be set that only authorized DB administrators are permitted from creating, editing, and removing data from database(s).

#### Audit

1. Only company administrators shall have website accounts.

#### Performance

1. Database shall also be optimized to add retrieve queries made quickly and efficiently.
2. Application shall be able to retrieve information from the database and display/act in a timely manner.
3. Website pages shall take no longer than two seconds to load

#### Capacity

1. The website shall handle at least 25 employees per company.
2. The website shall use no more than 75% of the total storage available.
3. The website shall be able to handle at least 100 shifts per employee.

**Data Integrity**

1. The integrity of of data shall be checked weekly/bi-weekly by administrators, back-end engineers, or Team Lead.
2. Data shall be backup during every check up.

**Recovery**

1. Recovering user data shall be the main priority in case of a complete fail.
2. The website shall not be down for more than 3 hours.

**Website Policies**

1. The website shall contain a link directed to the policies of the site.
2. Users shall agree to the site policies before creating an account.

**Conformance with coding standards and Scalability**

1. Application shall be developed, tested, and deployed using tools and servers approved by Class Professor and as agreed in M0.
2. Database(s) shall be optimized to fit ERD models and Business Rules as stated previously in this document.
3. Selected Database schema shall cover required columns and rows as dictated by ERD and application specifications.
4. Database shall scale relatively easy with all entries matching the column names listed in each database schema.
5. Data shall be stored in the team's chosen database (MySQL) on the team's deployment server.
6. The language used shall be in English.
7. Before data is pushed onto the database(s), it will be checked by administrators, back-end engineers, or Team Lead.
8. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.
9. Requests shall be handled in a asynchronous manner.

**Enumerate the work done by each team member (0-10)**

Edwin Menjivar (Team Lead)- 10

Kenneth Surban (Github Master)- 7

Jeffrey Piercy - 10

Rowvin Dizon - 7

Andrew Keelin - 10