Marisa Loraas

CSE 326: Software Engineering

3rd Assignment - System Design

1. System Design Goals
   1. Transparency/Ease of Use

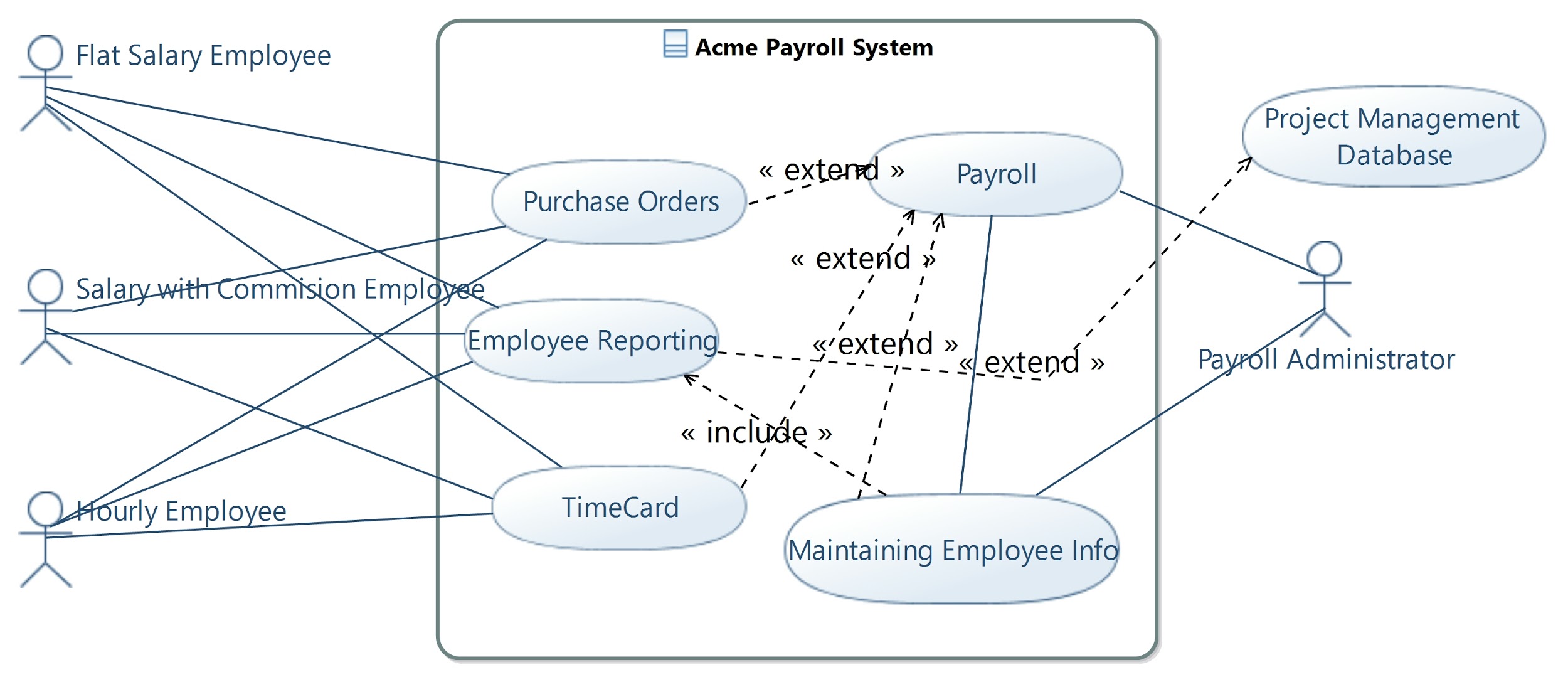
The system should provide a single, logical service to all employees of the company. It’s crucial that all employees can easily login, access and edit their time cards, quiere the system for personal information, and so system administrators can easily update employee information and run payroll.

* 1. Reliability

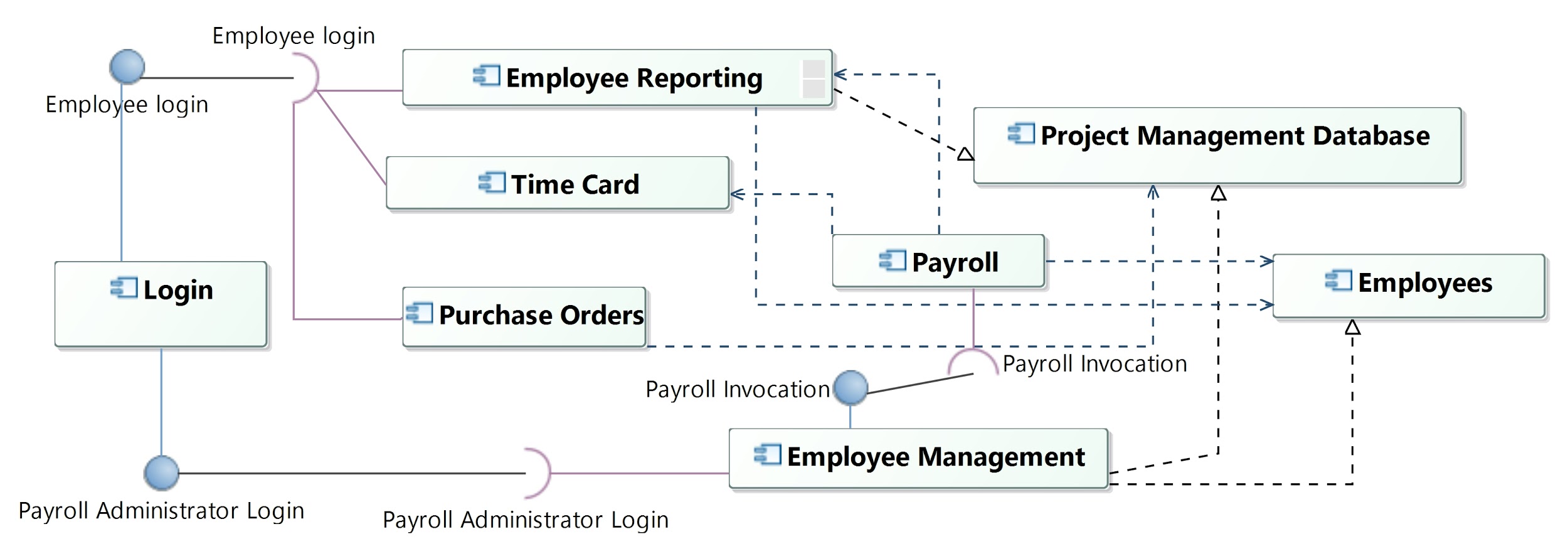
The system must be running the majority of the time (~98%) in order for employees to have the proper and accurate time cards for the week, and so administrators may keep information on employees and records the most up to date. It is especially essential that the system reliably works on paydays so that payroll runs on every Friday and the last working day of every month. The system should also survive any communication link problems.

1. System Design

Below is the Use Case Model I created for the ACME Payroll system from the 2nd Assignment.

****

Based on my comprehension of the ACME Payroll system and my UML Diagram, I’ve created the following Component Diagram to describe the system:

****

* 1. Architectural Style
     1. The ACME Payroll system has a Client and Server style architecture. In Client and Server style, the client subsystem calls on the server to provide it some sort of service. In our case, all employees have access to the client subsystem to do things like update time cards, put in purchase orders, and add or query the server subsystem for personal information. The server side of the system is responsible for executing payroll correctly, accessing and storing employee information correctly, and using the project management database when necessary. The utilization of the Project management database, and the database need to store all employee information (in component Employees) also gives this good precedence to be Client Server Style. The components that require employee logins in the component diagram given previously are the only components that users will have access to, which means they represent the client subsystem, and all other components represent the server subsystem.
  2. System Decomposition
     1. Client
* The client components from the ACME Payroll component diagram are as follows: Login, Employee Reporting, Timecard, Purchase Orders, and Employee Management.
* Login will determine employee access as well as payroll administrator access to the other components of the client subsystem.
* Employee Reporting will be a component that any employee can access and add or quirie their personal information, this requires the component to call from either the project management database to gain access to those records or the Employees component to gain a request changes to those records (both part of the server subsystem).
* Timecard merely lets an employee access and edit their time card and let payroll update that information.
* Purchase Orders would fulfill employee purchase orders for projects and send that information to the Project Management Database.
* Employee Management can only be accessed by the login of an Payroll Administrator, and allows them to maintain information on employees like adding and deleting employees and inputting all their personal information. This component will also allow the administrator to execute payroll, but not allow the administrator actual access to the payroll system, or any of the server systems.
  + 1. Server
* The components from the diagram that are considered to be apart of the server subsystem are as follows: Payroll, Project Management Database, and Employees
* The Payroll component pays all employees accordingly based on employee type, time cards, and sends the appropriate payment out based on how the employee has chosen to be paid. This must be invoked by a payroll administrator through their access to the Employee Management Component on every payday. Note, the payroll administrator does not have access to this component because it is a part of the server, even though they invoke it.
* The Project Management Database is a component not created by the ACME Payroll system, but implemented into the system in order to have access to purchase order information and to send the databases new purchase orders made by employees.
* Employees is the system that stores all essential information on all 5000+ employees at ACME. This component is the database to be accessed when employees have personal queries about their personal information, and is updated based on the Employee Management and Employee Reporting components. It is also used by payroll in order to pay each employee accordingly.
  1. Use Case Realization
     1. Maintain Employee Information

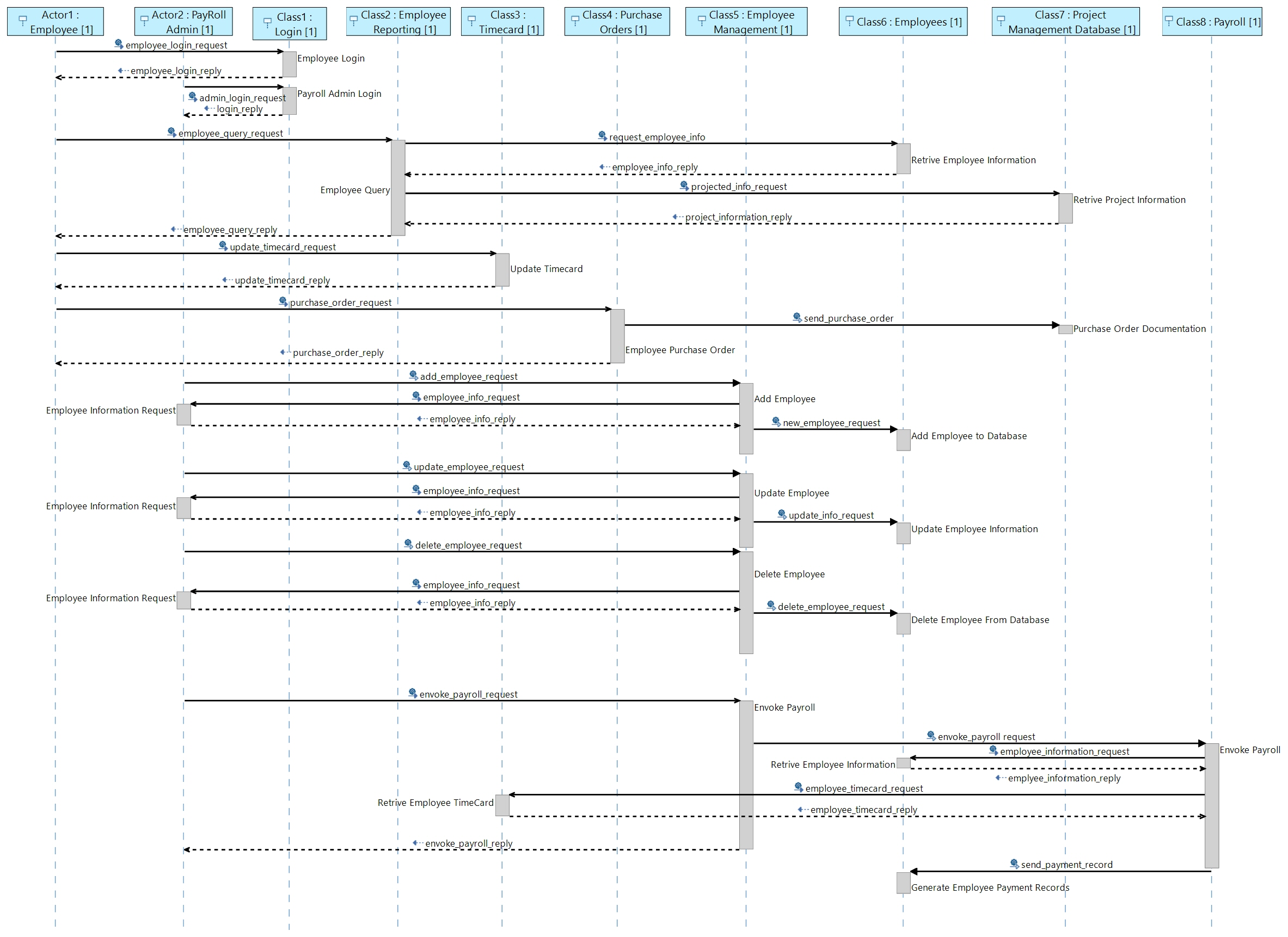
Maintain Employee information is realized mostly through the Employee Management component, as it requires Payroll administrator to login, and implements Add Employee, Update Employee, and Delete Employee. The Employee Management component will then send any changes to the Employees component to store and organize all employees information.

* + 1. Payroll

In my 2nd assignment, the other use case that I detailed was Payroll, which is very clearly realized by the Payroll component. Accessing Employee information is shown by the Payroll components access to the Employee component, giving it all information needed about all employees to get paid. The component is also in charge of generating payment records to be stored in Employees, and paying all employees properly when invoked by Employee Management. The component basically follows the exact flow described in the Payroll use case.

* + 1. Design Goals Realization
       1. Transparency/Ease of Use is something that is probably most effective in the Client and Server architecture style. Users are only interacting with the client subsystem, which is the simple basic interactions that the system is planning to implement with its employees. This means that all of the intensive work and data accessing is done on the server end, which makes the system very comprehensible for basic users.
       2. Reliability is best understood in separating the system into client and server subsystems, which allows for the server subsystems to be doing most of the computing and error checking for the system. This means there's little chance for the system's reliability to be user dependent, which makes the whole system a lot more reliable and less likely to run into errors. The server subsystems working with each other also checks things and allows for a more reliable system.
    2. Sequence Diagram

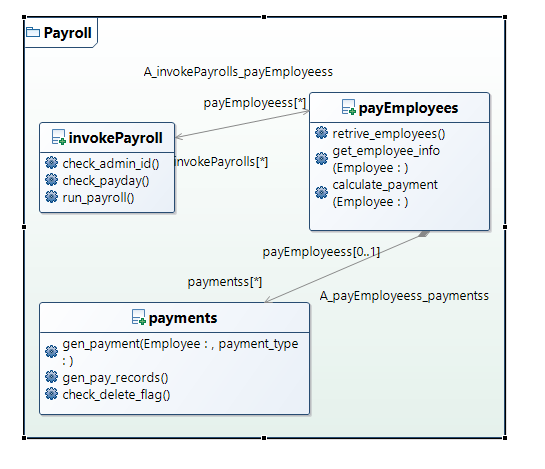
Below is the sequence diagram I’ve built for the ACME Payroll system based on the architecture system, design goals, and subsystem layout. Notice how actors don’t interact with Employees, Project Management Database, or Payroll, as they are a part of the server subsystem. This aligns perfectly with the Client and Server model I’ve discussed, and the following sequence diagram lays out that structure very well.



1. Subsystem Design
   1. Component 1: Payroll

* Payroll is a subsystem that is a part of the server system in the Client and Server style architecture.
* This subsystem should be invoked by a Payroll Administrator and the system should make sure the invocation is on a payday (a friday or the last day of the month); this is done in the *InvokePayroll* class.
* The system will then access the Employees Database subsystem to get Employee information needed to pay each employee properly(get\_employee\_info()). The component then will decide which employees need to be paid (retrive\_employees()), based on the pay day and employee type, then it will calculate the employees payment based on time card and/or employee type(calculate\_payment). This is all included in the *payEmployees* class.
* Payroll will then properly send payments to employees based on their choice of check or direct deposit (gen\_payment()), and generate/check payment records for the employee(gen\_pay\_records()), and delete the employee from the Employees component if it is their last pay day(check\_delete\_flag()). This is all in the *payments* class.
  + 1. Class Diagram

Below is the class diagram created for the Payroll component.

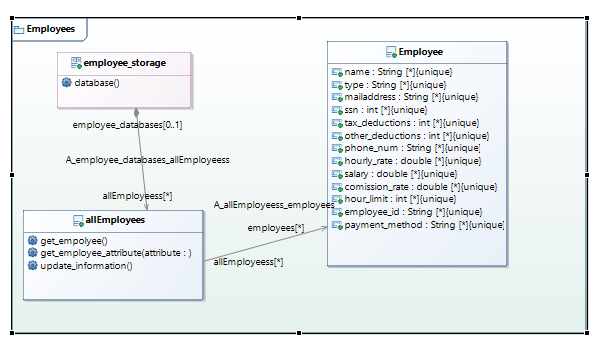


* + 1. Nested Subsystems

None

* 1. Component 2: Employees
* The Employees subsystem is responsible for maintaining all information on Employees. Though very simple, it is probably the most important component of the server, for it has all employee information that is needed for Payroll, Employee Management, and Employee Reporting to function properly.
* The *Employee* class contains all the attributes that the payroll system keeps track of for each employee. This includes: their name, employee type, mailing address, social security number, normal tax deductions, other tax deductions, phone number, hourly rate, salary, commission rate, hour limit, employee ID, and the employees choice of payment method.
* There should be some sort of database system here that allows for all of the information on the 5000+ employees. This is (I’m assuming) a nested subsystem within this subsystem. It is represented as the *employee\_storage* class in the class diagram.
* This component should also be able to retrieve information on employees from the database and consider updates to the employee database implemented by a payroll admin. This is represented in the class *allEmployees*
  + 1. Class Diagram

Below is the class diagram for Employees



* + 1. Nested Subsystems
* The employee\_storage class is supposed to represent some sort of database system within the employees that’s able to maintain all of this information. I have no knowledge of database development so I made this an abstract class in its representation in the class diagram, because honestly I don’t know what goes into database implementation. But I assume this would be some sort of nested subsystem.

1. Bonus
   1. I did not do the bonus, however I would like to point out that on Canvas, for this assignment *and* the last assignment, the **bonus** points are not discounted from the overall score of the assignment. It should be out of 100 not 110. We’ve tried talking to Dr. Shin about it for months and he still has done nothing to change this. I would appreciate whichever TA is grading this to look into that and maybe help us students out a little. Thank you so much.