Automated Payroll System

Andrew Levak

Omardeen Rose

Term Project 3

Abstract

Getting paid is something we all look forward to, whether this process occurs weekly, bi-weekly or even monthly. This is necessary to promote and ensure workers are being productive and keep the moral of the workplace positive. Getting paid is an important step for us to all proceed with our daily lives. Without being paid we would not be able to pay our mortgage, utility bills, car notes, insurance and our creditors. In today's world where the economic recession has brought the issues of pay to the forefront along with expenses, because companies must manage effectively to remain successful, the subject has been getting much attention.

An often overlooked part of getting paid is the actual payroll process. This process includes the jobs and roles performed by the payroll department, managers and yes, even us, the employees. Going forward, we will delve deep into this world of payroll and its internal processes shared among many in the companies we all invest in or even where we work on a daily basis.

Introduction

Payroll departments across the nation operate as the hub or centralized location for all things involving payroll within the companies they operate. We will take a closer look at once such company - this company’s core function is operating as a call center. Close analysis of the daily activities of each of the employees, managers, payroll department and roles played by databases and systems will be examined in an effort to automate such a process. Included in the analysis will be high level overviews of each process with their respective stakeholders. Multiple diagrams, including activity, class, sequence, swimlane, and use cases, will be utilize to illustrate the inner workings of the detailed processes they represent. After review of the analysis done, we invite you to draw parallels to your company or employer on ways in which some of your payroll process may be automated to reduce the overall operational cost for the business.

Method

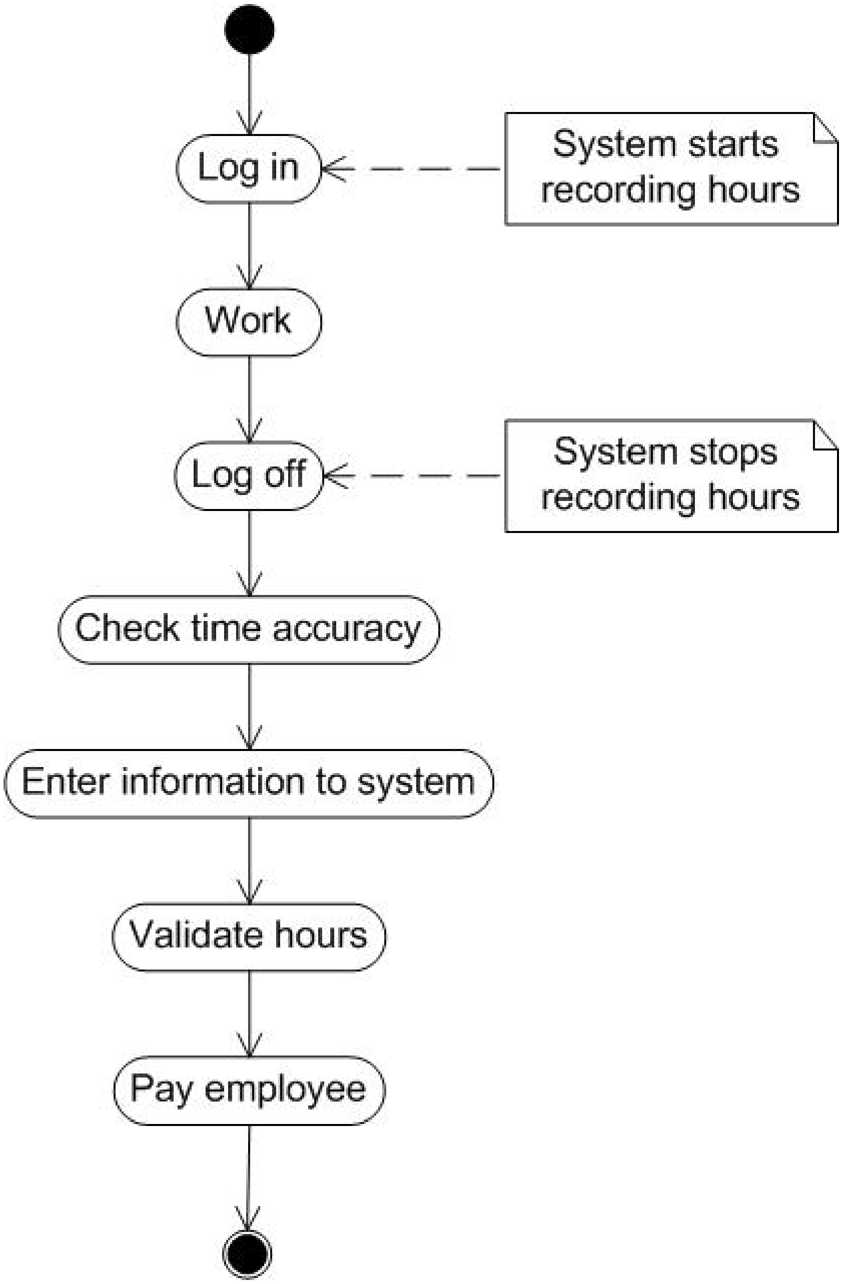
For the start of the term project, we begin to break down what happens throughout the process, first from a macro scale, ignoring almost all technical details or use cases, and then we begin to narrow it down and enter the micro stage. To simplify the scope of our automated payroll system, we break down the activities involved for our three stakeholders: the employee, the manager, and the payroll department, showing what their specific responsibilities or interactions are within the system. Activity diagrams are designed to show what happens during an operation or process for a specific individual, or just the broad scope of the project itself.

From the activity diagrams, a swimlane diagram can be manufactured to show each stakeholder's responsibility within the framework of the entire system. This helps one visualize the different roles that they are involved with throughout the project. These diagrams can further be broken down into class diagrams, which deal with the technical details that we had left out of the activity and swimlane diagrams. The class diagram groups similar attributes and processes to form a 'class'. This is a step down in the design phase dealing closely with the micro stage. It is when the project starts to flesh itself out while more and more details are being presented. Using the class diagrams, we can construct use cases which deal with specific instances of the processes as well as sequence diagrams which illustrate timelines and interaction between multiple stakeholders.

Results/Discussion

The first activity diagram deals with the complete scope of the intended project itself as shown below.

Figure 1.1 – System Overview Activity Diagram


Log in
System starts recording hours
Work
Log off
System stops recording hours
Check time accuracy
Enter information to system
Validate hours
Pay employee

As can be seen from the above activity diagram in Figure 1.1, the general idea of the system is that when the employee logs in to his specific system at work, the work hours start to be tracked. As he continues his work throughout the day, his time will be monitored by the system until the day ends and he logs out of his system. The manager will then check the accuracy of the hours, accounting for exceptions which will be shown in further detail in the more in-depth activity diagrams. Once the hours have been quickly gone over by the manager, the payroll department will have access to those records from the database and will validate the hours. When the hours have been validated, the employee will receive monetary compensation based upon the number of hours they legitimately worked.

For another type of diagram, the swimlane shows the system overview in terms of specific stakeholders and their responsibilities. Figure 1.2 on the next page shows the swimlane diagram in terms of the employee, the manager, and the payroll department as well as their general incorporation into the processes of the entire system.

Figure 1.2 – Swimlane diagram

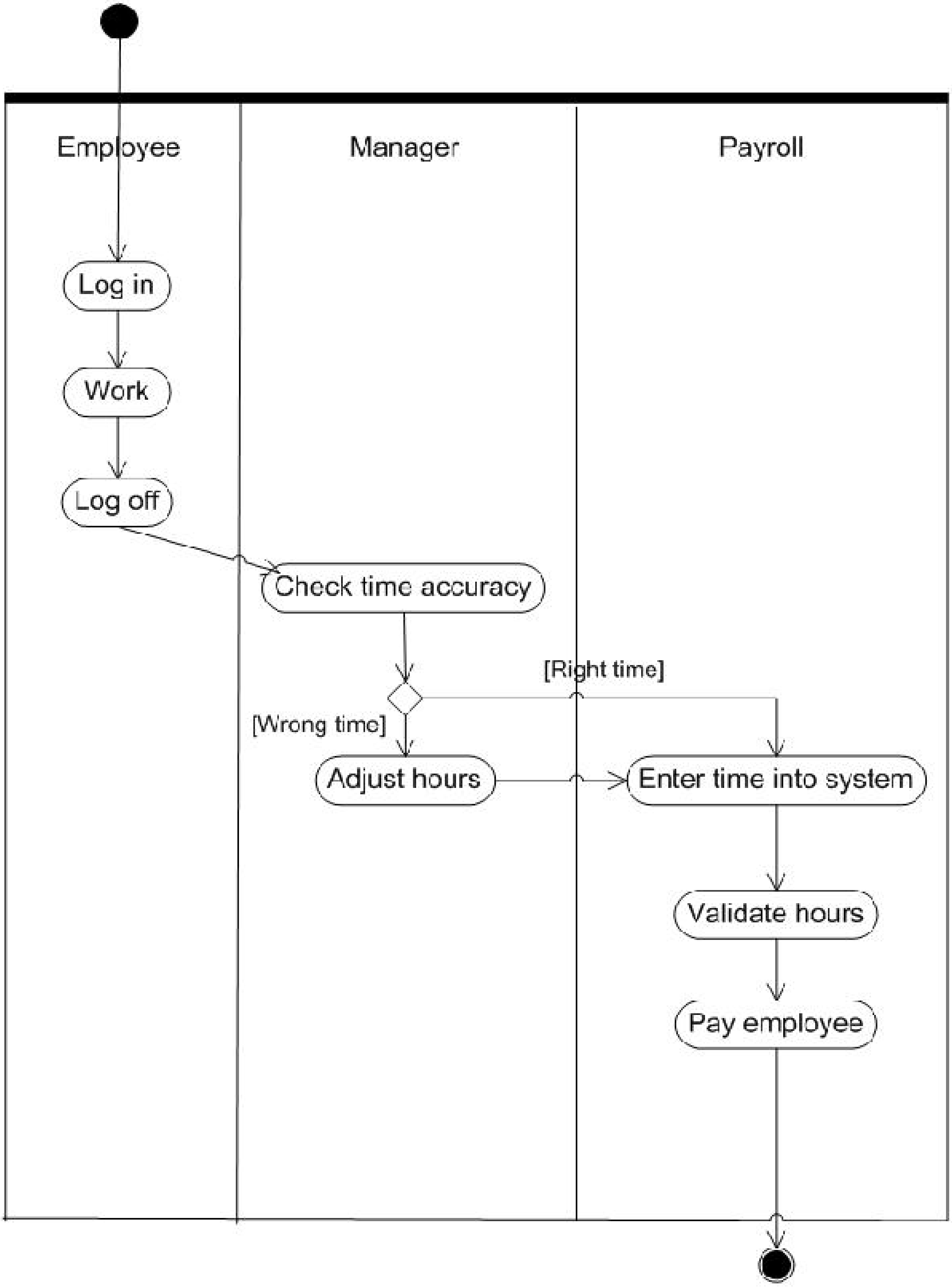
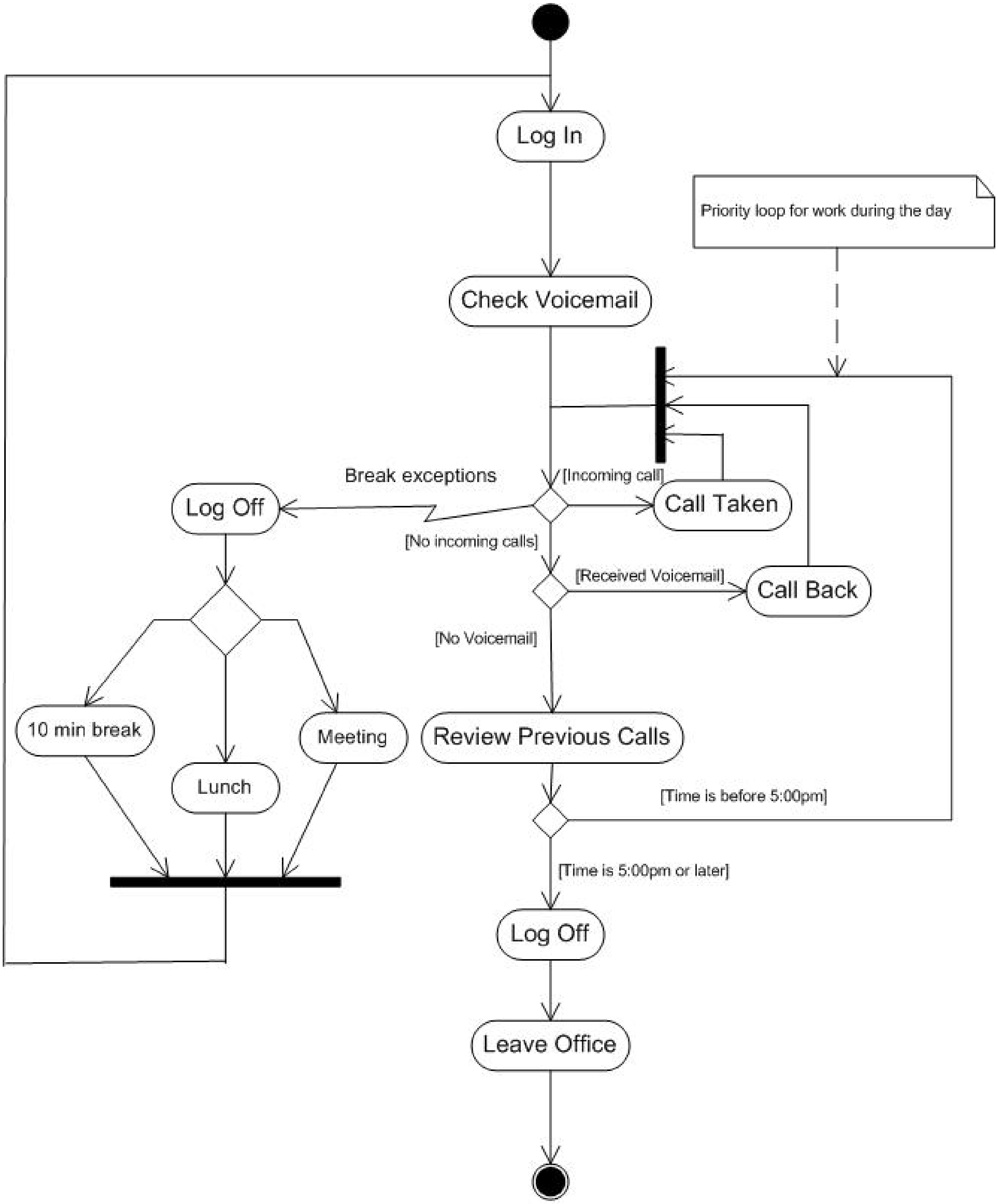


Figure 1.3 – Employee Activity Diagram



In Figure 1.3, we show the employee's activity diagram. This diagram is more specific to the job responsibilities of the employee in regards to the payroll system. The true hours that an employee will be paid for will be based upon his activity throughout the day by taking calls, making calls etc with little idle time in between. It begins by the employee logging into their station. They will check their voicemail to see if they have any customers that they will need to call back from the previous work day. The main concern of the job is that the employee will be taking calls from either existing clients or potential clients. If there are currently no calls to be taken, they will call back the clients based on the voicemail messages received earlier in the day.

If the employee has returned all calls and does not have any incoming calls in which to take, they will review their previous calls for ways in which to improve their service or help the client. This cycle is infinitely repeated throughout the day except for a few exceptions. Exceptions can technically occur at any time throughout the day, such as a meeting being held, a 10 minute break, or a lunch break. These exceptions will all require that the employee log off his station. Because certain exceptions still require that the employee be paid for that particular time spent, part of the system will be to handle those exceptions and pay the employee accordingly. We have not worked out those specific details on how it will be handled yet, but it will be taken care of later in the project.

The work process continues until the employee gets out of work, in this case 5:00pm, and they log out of their station. This timeframe that occurs when the employee first logs in, and finally logs out at the end of the day, is the timeframe in which they will be paid. The system optimally will take that time, handle the idle time, account for the exceptions, and store it in the employee's records in the database where the manager and payroll department have access.

Figure 1.4 – Manager Activity Diagram

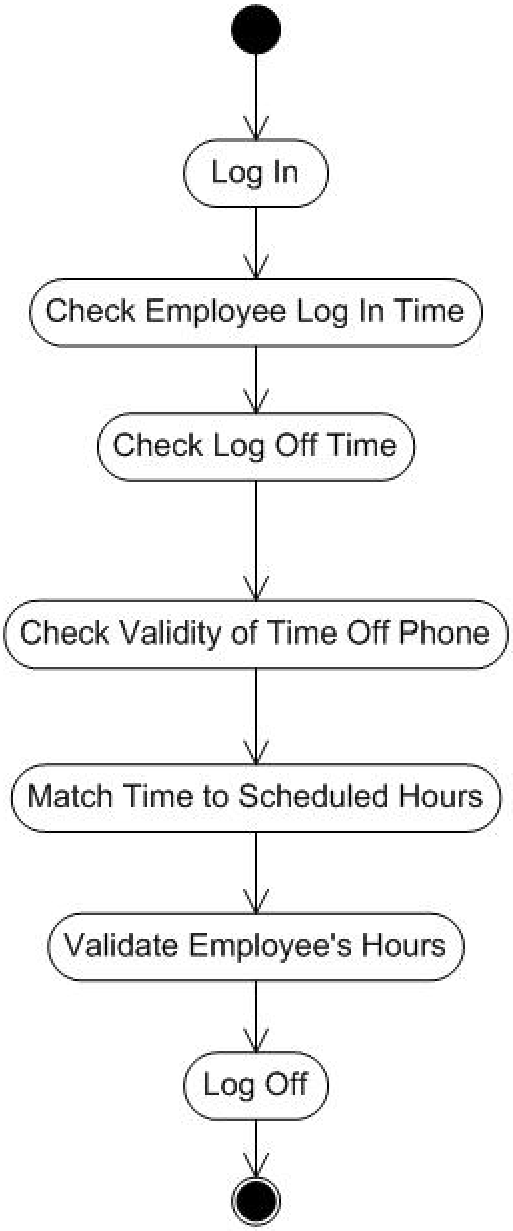
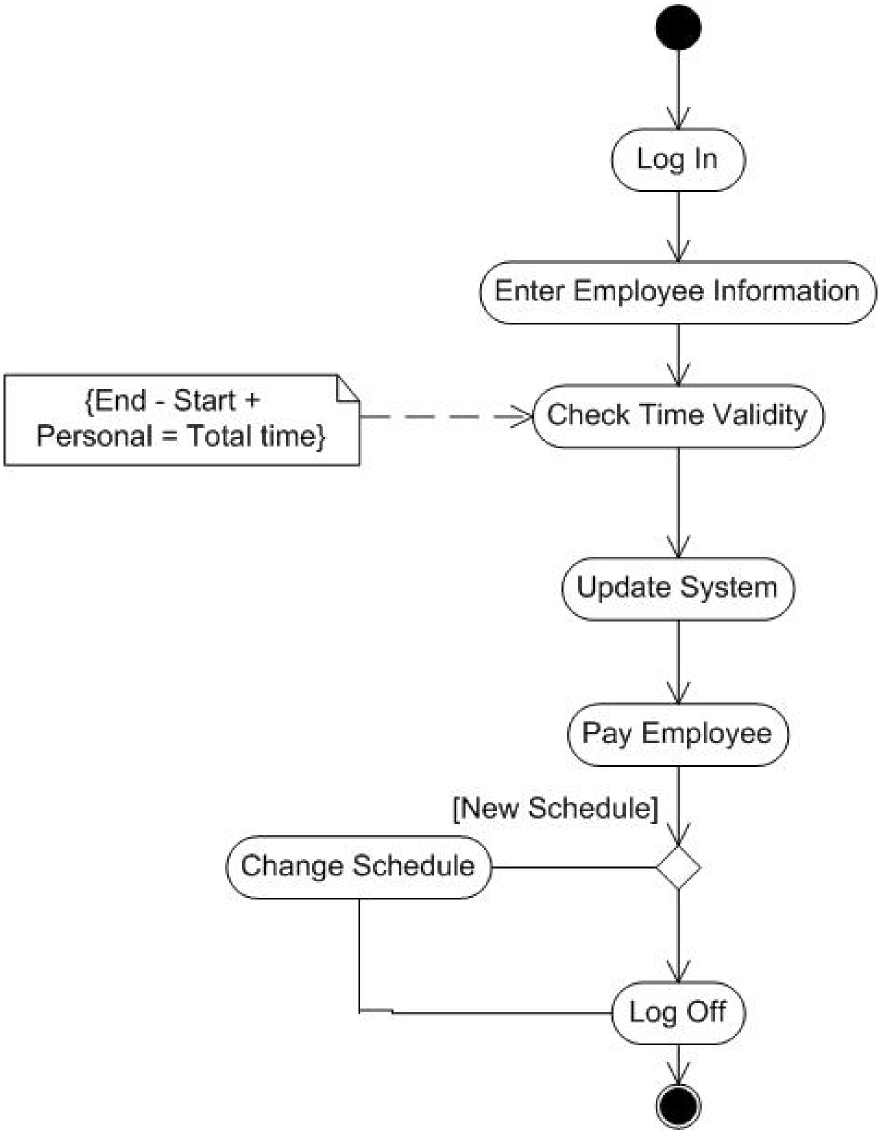

Log In
Check Employee Log In Time
Check Log Off Time
Check Validity of Time Off Phone
Match Time to Scheduled Hours
Validate Employee's Hours
Log Off

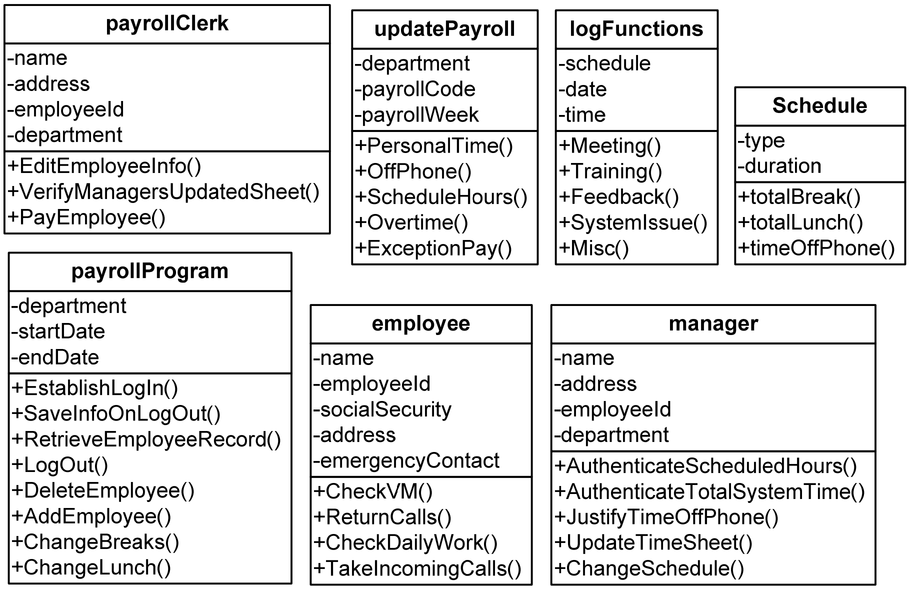
Figure 1.4, above, is the manager's side of the process shown in its own activity diagram. The manager will log into the system, just the same as the other stakeholders. He or she will quickly double check the system's assessment of the employee's hours and pay for that time, such as when they logged in, how much idle time was spent, paid time for meetings, breaks, or lunch, account for sick time or vacation time, and when the employee logged off. That time will be matched against the employee's scheduled hours and then validated for the payroll department.

Figure 1.5 – Payroll Activity Diagram



In Figure 1.5, the responsibilities of the payroll stakeholder are displayed in their own activity diagram. The payroll employee's role in the process is similar to the manager's. Because payroll is such an integral and important part of any business, double and triple checking can easily save money or time in the long run by eliminating any potential problems. "An ounce of prevention is worth a pound of cure." The payroll department checks the time for the employee and updates the system accordingly. This can include anything from slightly adjusting their pay rate per hour, hours worked, adding or deleting an employee if they no longer work there, transferring their information to another branch if they are moved, or changing contact information. Once the system has been updated from their end they will issue checks if it is correct pay day, or just accumulate a current check. They will then either keep the same schedule, or issue a different one and log out of the system.

Figure 1.6 – Class Diagrams

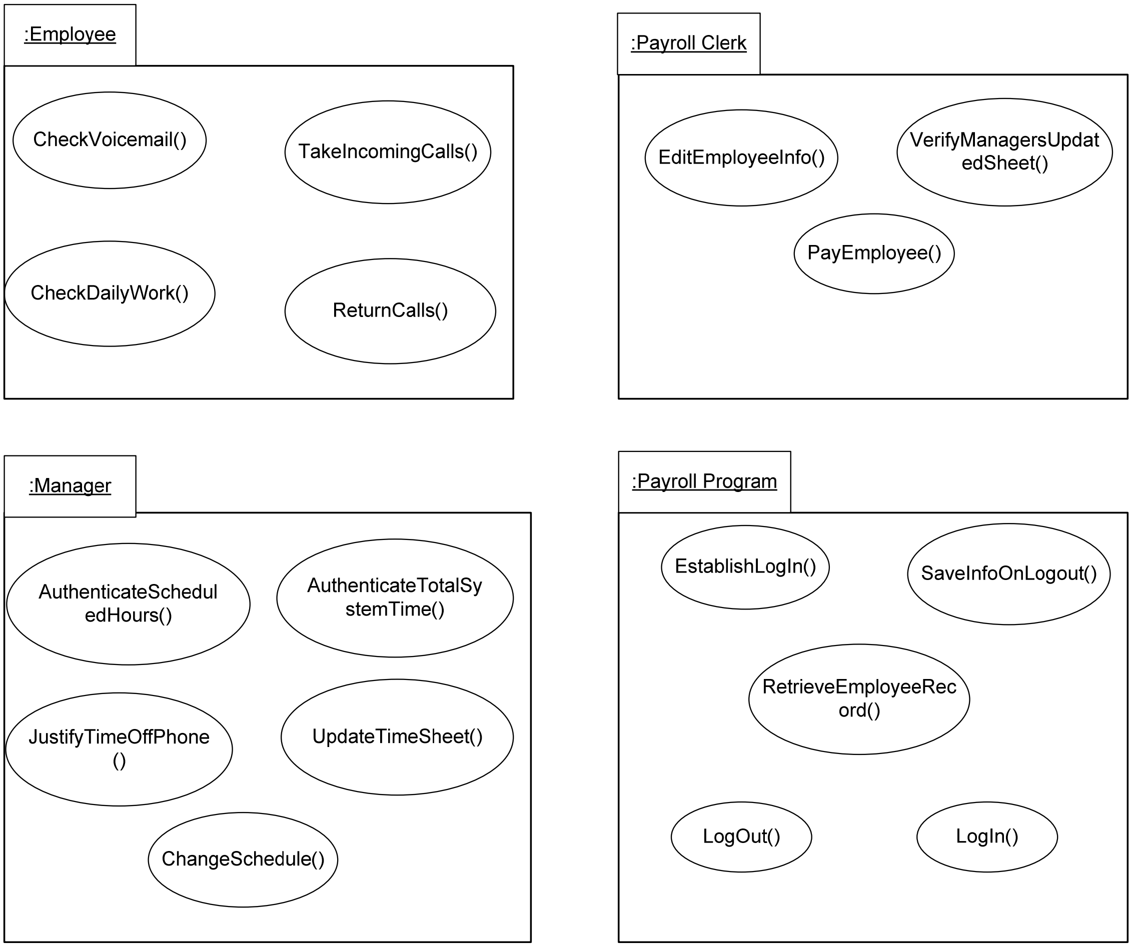


While the activity diagrams show a general overview of what a stakeholder's job is in terms of the system, they don't deal with specific details. In Figure 1.6, we show a class diagram. A class diagram deals more with the technical details of the project including attributes. It is generally made by removing the nouns and verbs from the activity diagrams and grouping them accordingly.

The *Schedule* class deals with some of the exceptions talked about earlier, such as the different breaks, either a 10 minute personal break or a 30 minute lunch break. It also includes the idle time spent off the phone while working throughout the day. *logFunctions* deals with other exceptions such as meetings or training that occur requiring the employee to log off but time that still needs to be paid for. The *payrollProgram* class deals with the actual database of the system which stores all of the employee records and work hours. It can accept changes such as lunch times, schedules, employee information, add or delete an employee record, as well as other general functions for controlling the employee's records.

The *updatePayroll* class is the one that one that actually changes information in the database. It takes personal time, time spent off the phone, scheduled hours, overtime, and exception pay as parameters that will be used with the *payrollProgram* class. The *employee* class has multiple attributes to differentiate between employees and signs in and out of the system as well as meetings, as well as using the system to do his daily work. The *manager* class also has attributes similar to the employees and its parameters are the verification of employee records, the time that the employee spends working on various activities, as well as the ability to change his schedule. *payrollClerk* has similar attributes to the manager, and its parameters are the abilities to change employee information, verify the manager's updated time sheet which is basically just a double check of the pay of that employee, as well as distribute the pay.

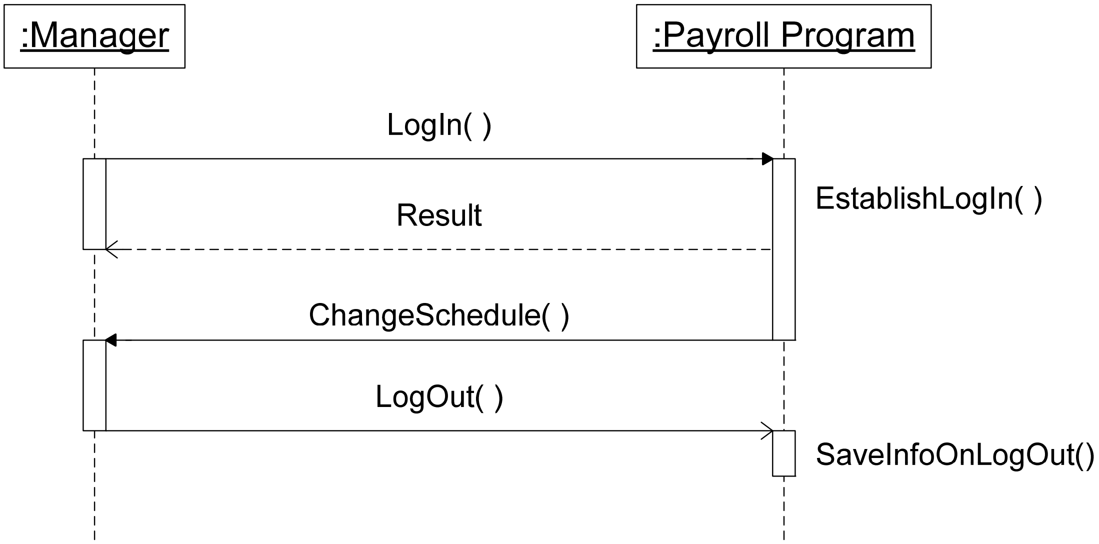
Figure 1.7 – Use Cases



Use cases in package diagram form are designed to help analysts work with users to determine the system usage on the user level. By determining the use cases, it allows us to discover new functions that each stakeholder would apply during the course of the system. These new discoveries show up in updated class diagrams. The specifics on the actual functions were discussed in detail in the class diagram section, so we won't need to explain them again.

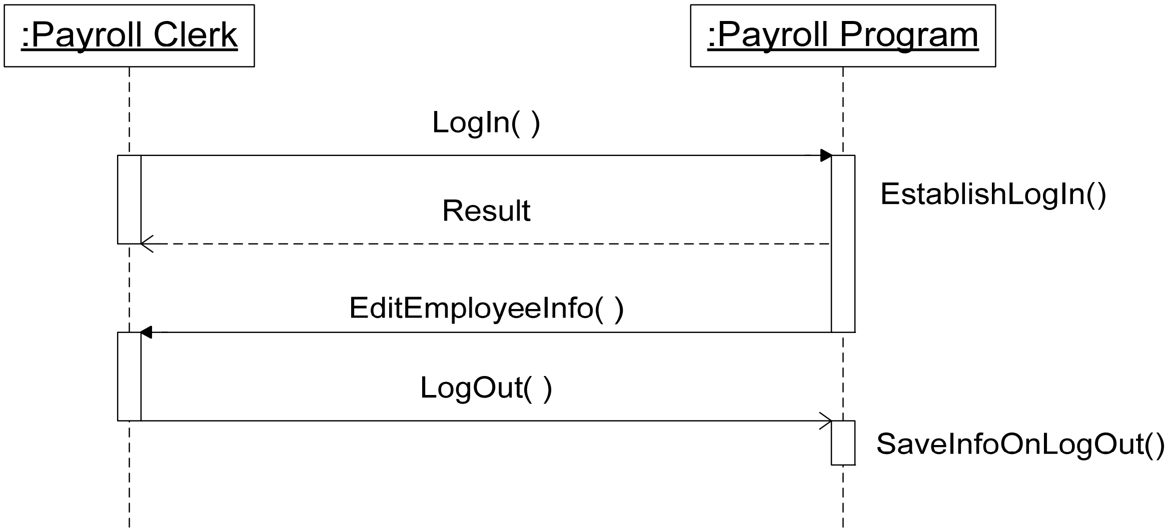
Sequence Diagrams

Figure 1.8 – Employee Schedule Change



Sequence diagrams are used to emphasize the timeline structure, as well as the stakeholder interaction of the use cases. In Figure 1.8, a simple sequence diagram is shown that deals with changing an employee's schedule. The manager logs into the system by calling the payroll program's function "Log In". If the result is a successful login, payroll program calls the manager’s function “ChangeSchedule( )”, logs out, and then saves the information into the database.

Figure 1.9 – Employee Information Change



This sequence diagram is similar to the one in Figure 1.8, except the payroll clerk is the stakeholder using the payroll program to edit an employee’s information, such as employee id, address, emergency contact, per hour salary etc.

Figure 1.10 – Employee's Daily Work

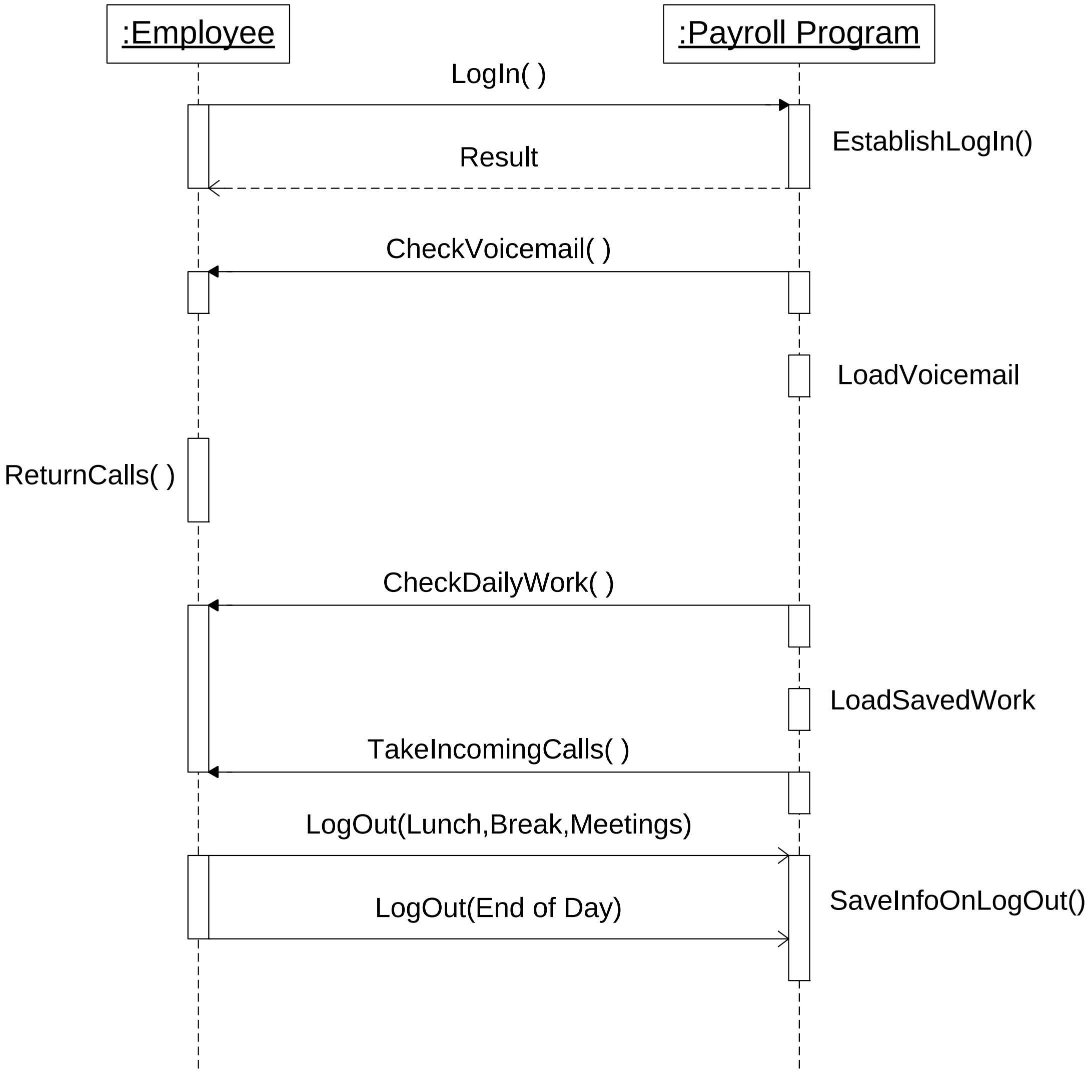
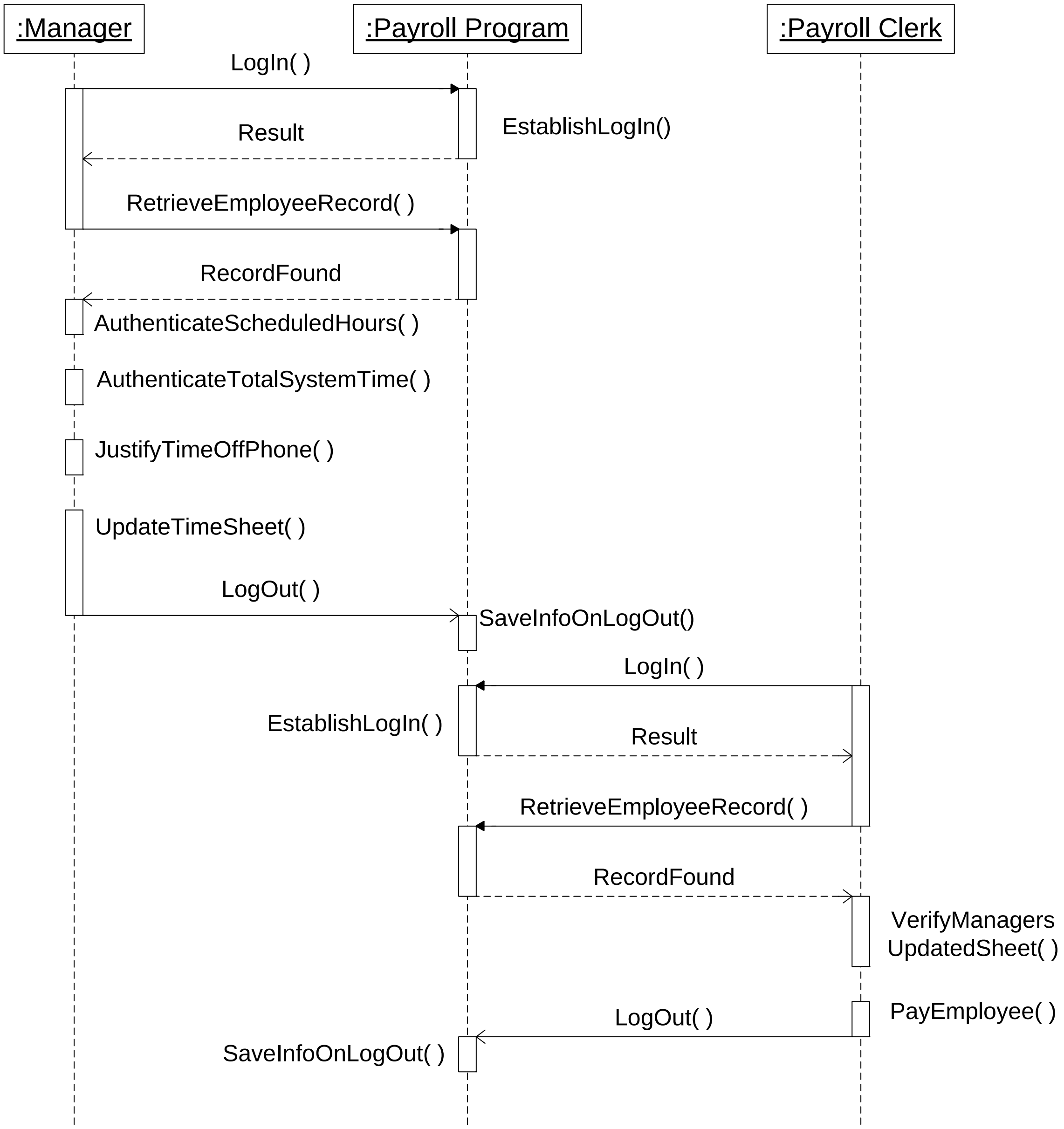


Figure 1.10 deals with the employee's daily work and interaction with the payroll program. Once he logs in, he can call to check his voicemail, which the payroll program will load for him. From the voicemail he received, he might want to return those calls if

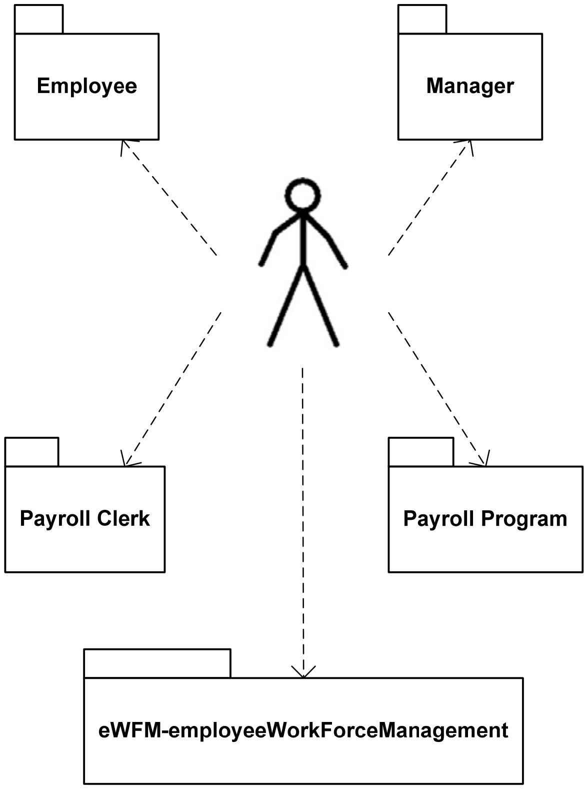
he received any. He can review his daily work or he can take incoming calls, which is the main function of the company, since it is a call center. When he needs to log out, either from breaks, meetings, lunch, or just because it is the end of the day, the payroll program saves his information.

Figure 1.11 – Payday Process



This sequence diagram deals with the process of payroll every two weeks, which is when the employees would get paid. The manager logs into the system, then pulls a specific employee's record up. He would then match his scheduled hours up to the amount he actually worked, adjusting for such things as overtime etc. He will then check the employee's time off the phone, which is tracked by the payroll program, and account for things such as breaks. If too much time was spent off the phone which wasn't a result of breaks, his pay will be adjusted accordingly. He then updates the time sheet and logs out for the system to save. The payroll clerk will then log in, retrieve that same employee's record, and double check the changes the manager has made. The payroll clerk will then confirm, pay the employee, log out, and the system will save the information.

Figure 1.12 – Overview



The above diagram (Figure 1.12) is a high level overview; it serves to show the relationships that exists and communication among processes, database and users. While the diagram is not a defined UML diagram the purpose it serves may actually be more powerful in communicating it's purpose than the sequence, class or package diagrams.

User Interfaces

Figure 1.14 – Log In UI

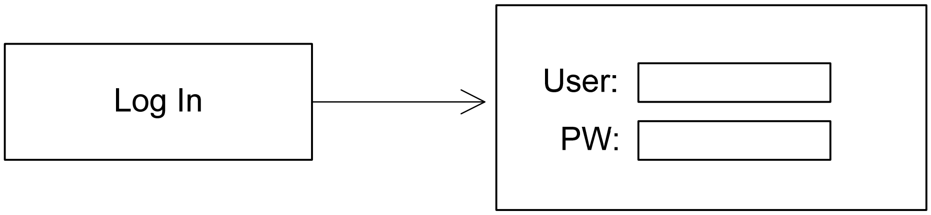


Figure 1.14 is an actual representation of the log In screen, this would be the default log In screen used by all stakeholders that exist within the Automated Payroll System domain.

Figure 1.15 – Manager UI

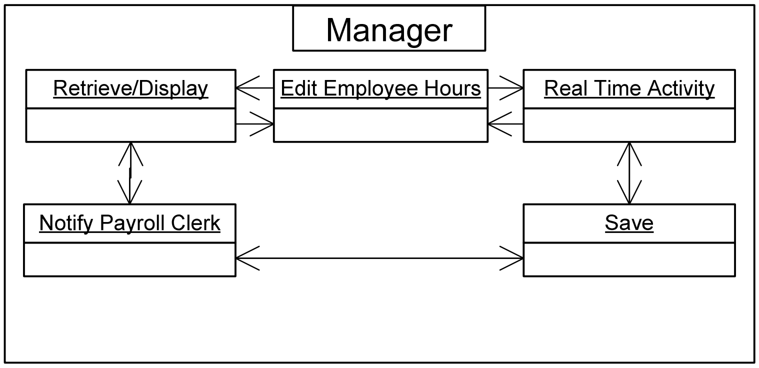


Figure 1.15 is an actual representation of the user interface used by the manager. This screen contains the menu options that correlates to the used cases and encapsulates all the actions necessary by a manager.

Figure 1.16 – Payroll Clerk UI

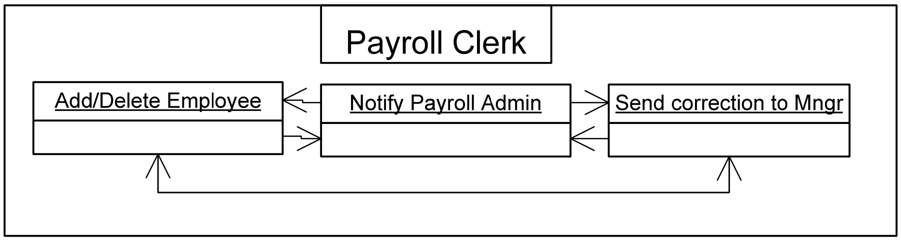
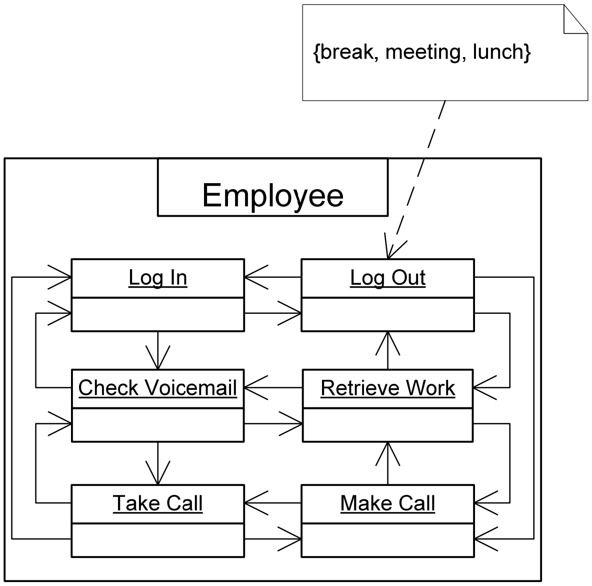


Figure 1.16 is an actual representation of the user interface used by the payroll clerk. This screen contains the menu options that correlates to the used cases and encapsulates all the

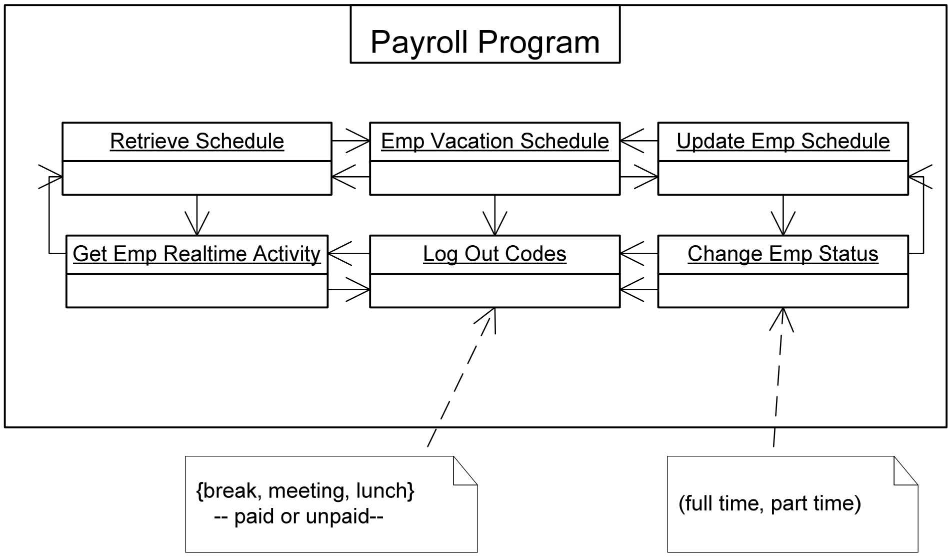
actions necessary. In this user interface a working assumption is that the payroll clerk will have the employees work schedule and hours automatically entered. The corrections needed would be minimal; this would be the best case scenario. In a worst case scenario there would be corrections would be needed, this would require the “edit/change” option not shown above.

Figure 1.17 – Employee UI



The most involved of all the user interfaces, this is because of layered user interfaces that exist. One specific case would be log out – in this case the log out command would bring up the log function menu which will in turn communicate with the payroll program (eWFM) and drive weather each option is ultimately paid or unpaid.

Figure 1.18 – Payroll Program UI



The payroll program user interface shown above otherwise called Employee Work force Management (eWFM) is the database management & communication platform. In essence it communicates with the manager, employee, payroll clerk and payroll department. It contains enhancement that are simulates as signals or messages that are displayed on a real time basis to aid the manager in managing the employees current work states. eWFM will send a display message stating which employee is “alarming” for being late to work or late from a schedule break or lunch.

Sequence Diagrams with UI

Figure 1.19 – Employee Daily Work with UI

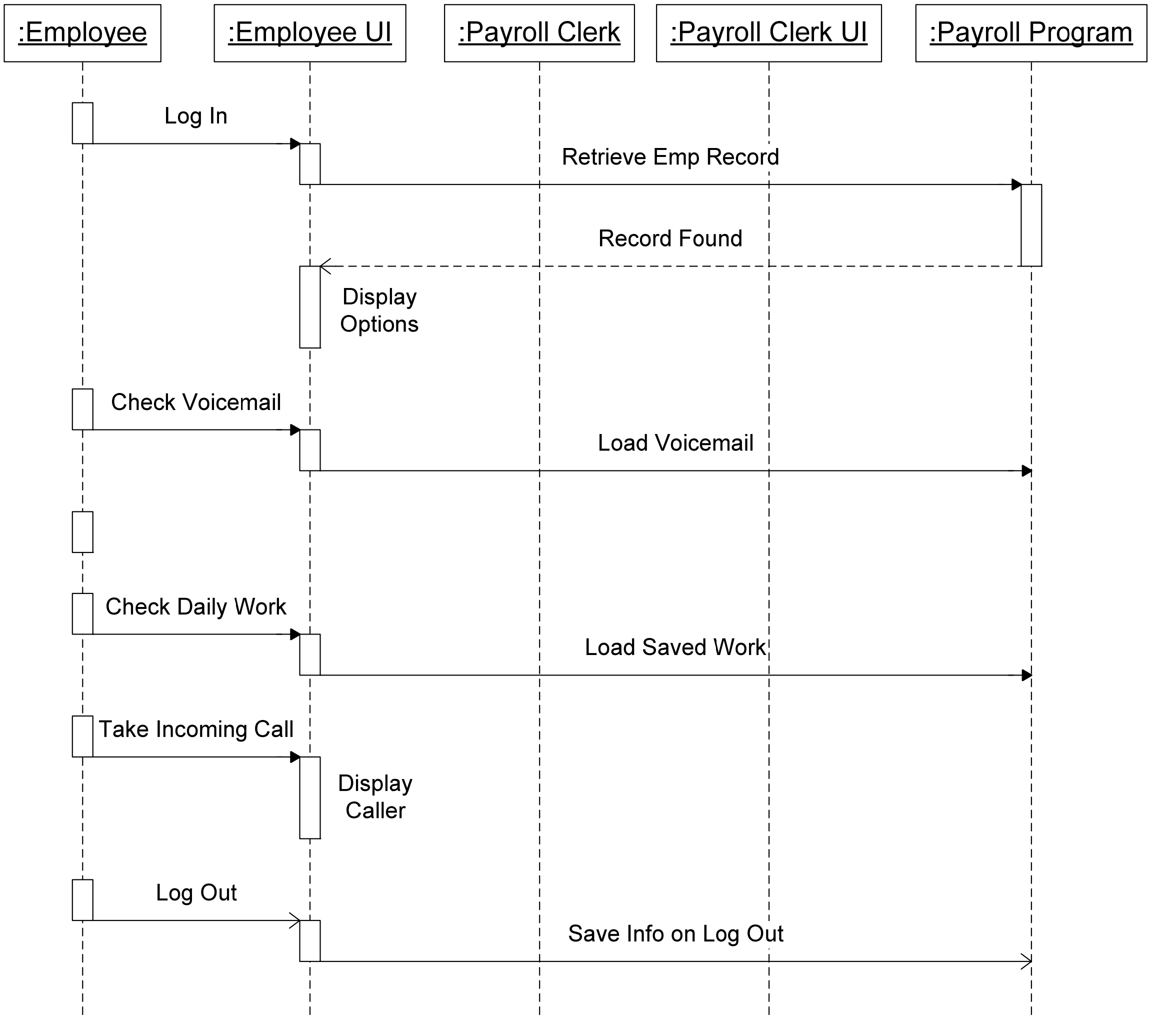
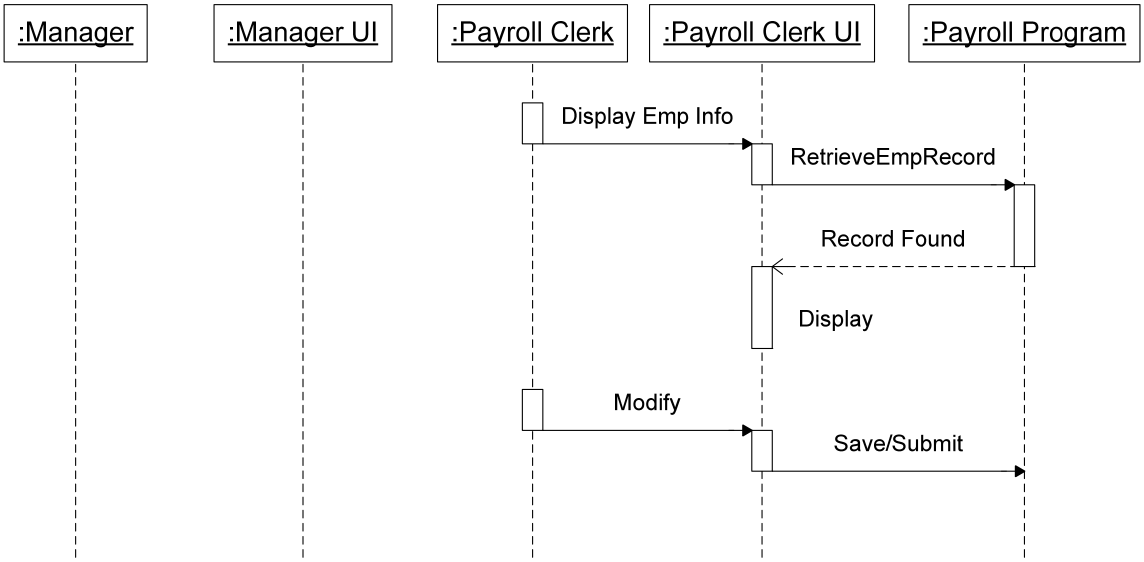


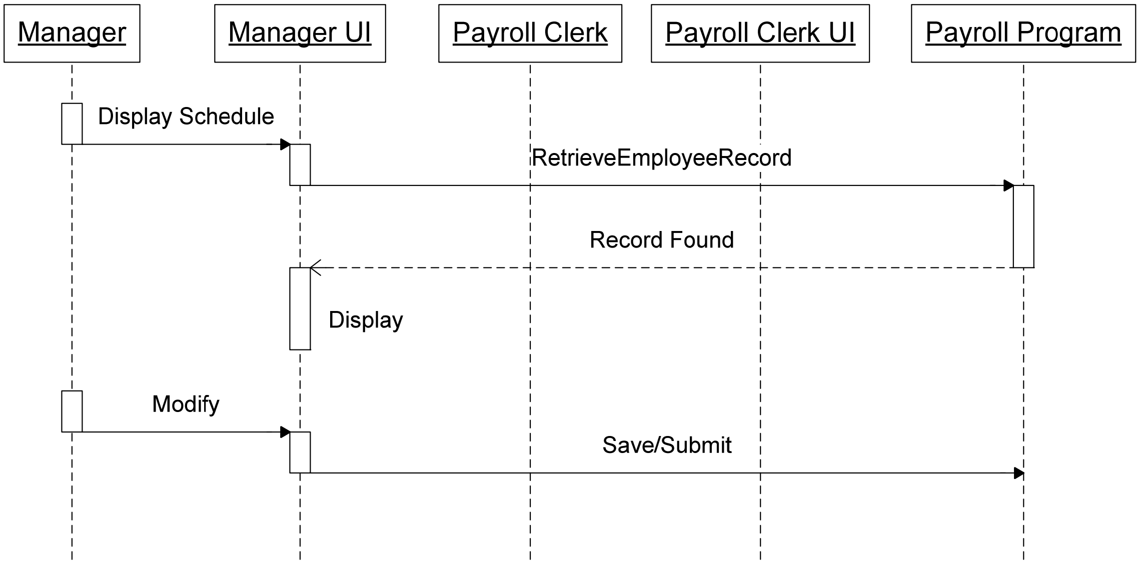
Figure 1.19 shows the sequence of a day in the life of the employee. The life cycle shown above represents the cyclical process that occurs. Based on the actions taken by the employee the payroll program will record and update the required databases.

Figure 1.20 – Employee Info Change with UI



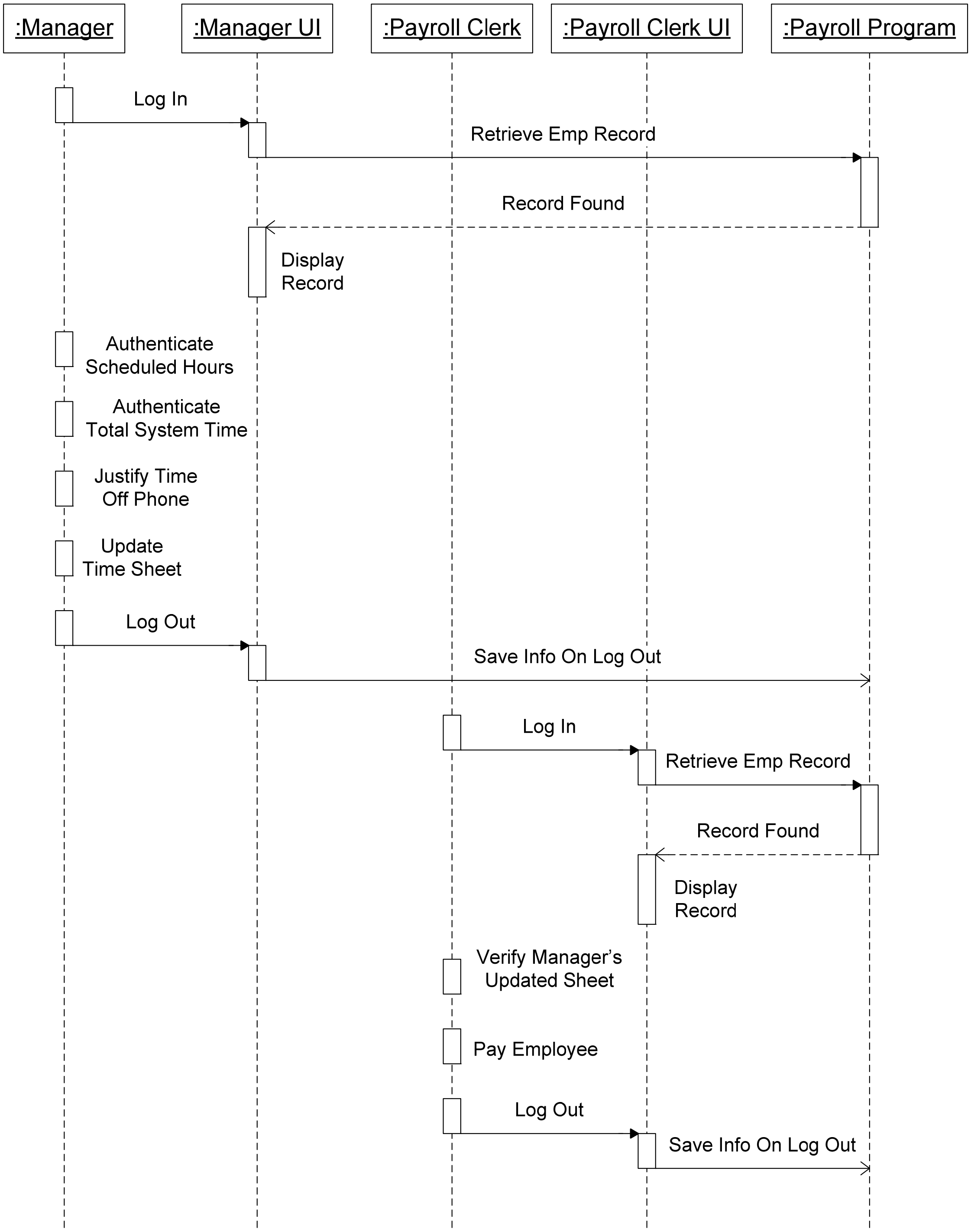
When a change is made there is communication between the payroll clerk, the payroll clerk UI and the payroll program itself.

Figure 1.21 – Schedule Change with UI



The above shows the manager making changes and the communication between the payroll program and the manager UI.

Figure 1.22 – Payday Process with UI


:Manager
:Manager UI
:Payroll Clerk
:Payroll Clerk UI
:Payroll Program
Log In
Retrieve Emp Record
Record Found
Display Record
Authenticate Scheduled Hours
Authenticate Total System Time
Justify Time Off Phone
Update Time Sheet
Log Out
Save Info On Log Out
Log In
Retrieve Emp Record
Record Found
Display Record
Verify Manager's Updated Sheet
Pay Employee
Log Out
Save Info On Log Out

Conclusion

The detailed analysis performed on each process has revealed the complexity and sometimes the simplicity of the payroll process. As a result of this the process designed on automation the system will benefit from a top down design. It will allow for exception time off the phone to be easily coded as paid or unpaid. Such changes will reduce the time spent by managers checking time accuracy, payroll clerks keying individual entries and thus by spending less time for each of these activities the cost of payroll can be reduced and the accuracy of the process increased. Please read the analysis done and draw your own conclusions.

References

Image from coversheet [1]

www.pfsacctg.com