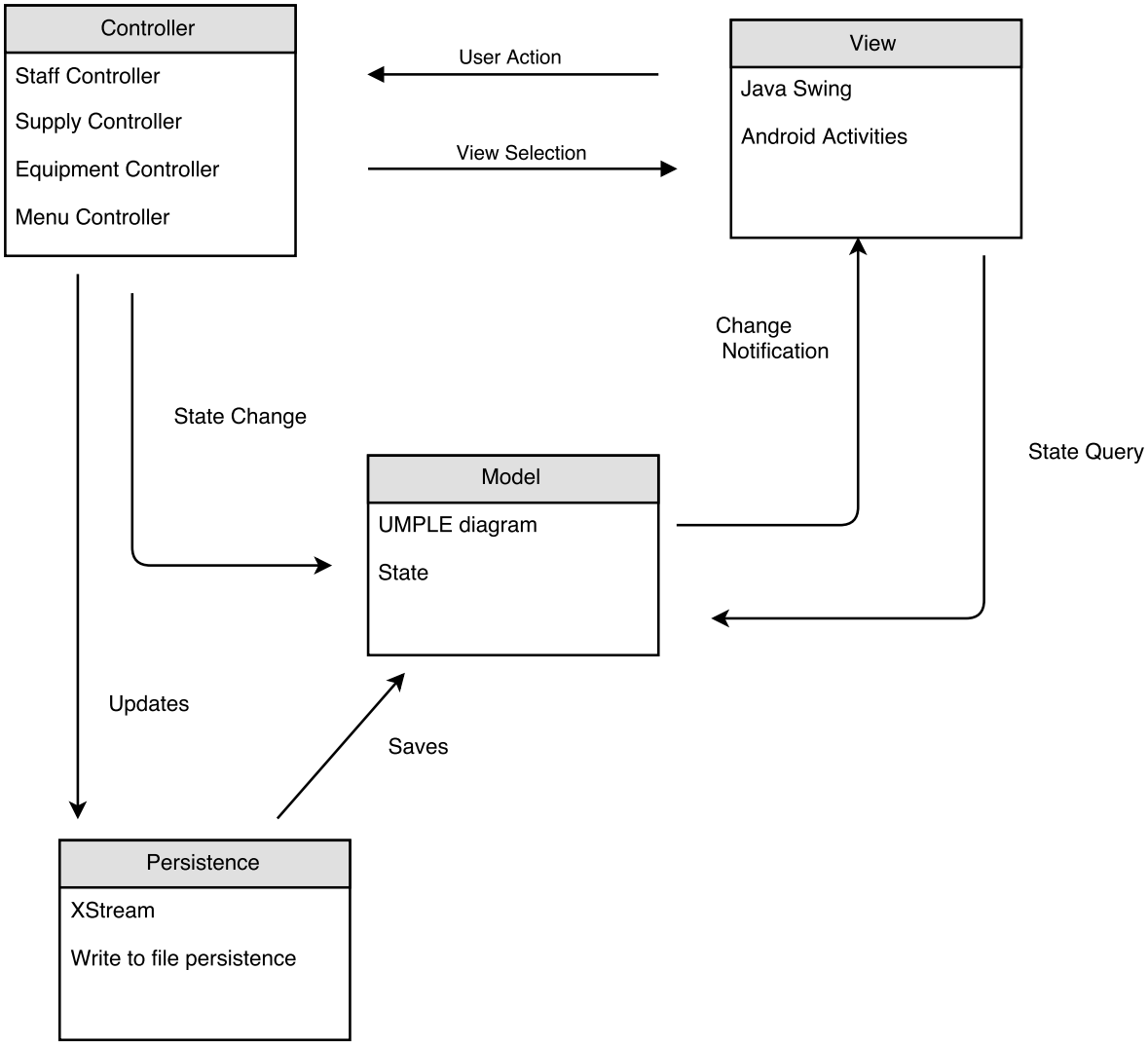
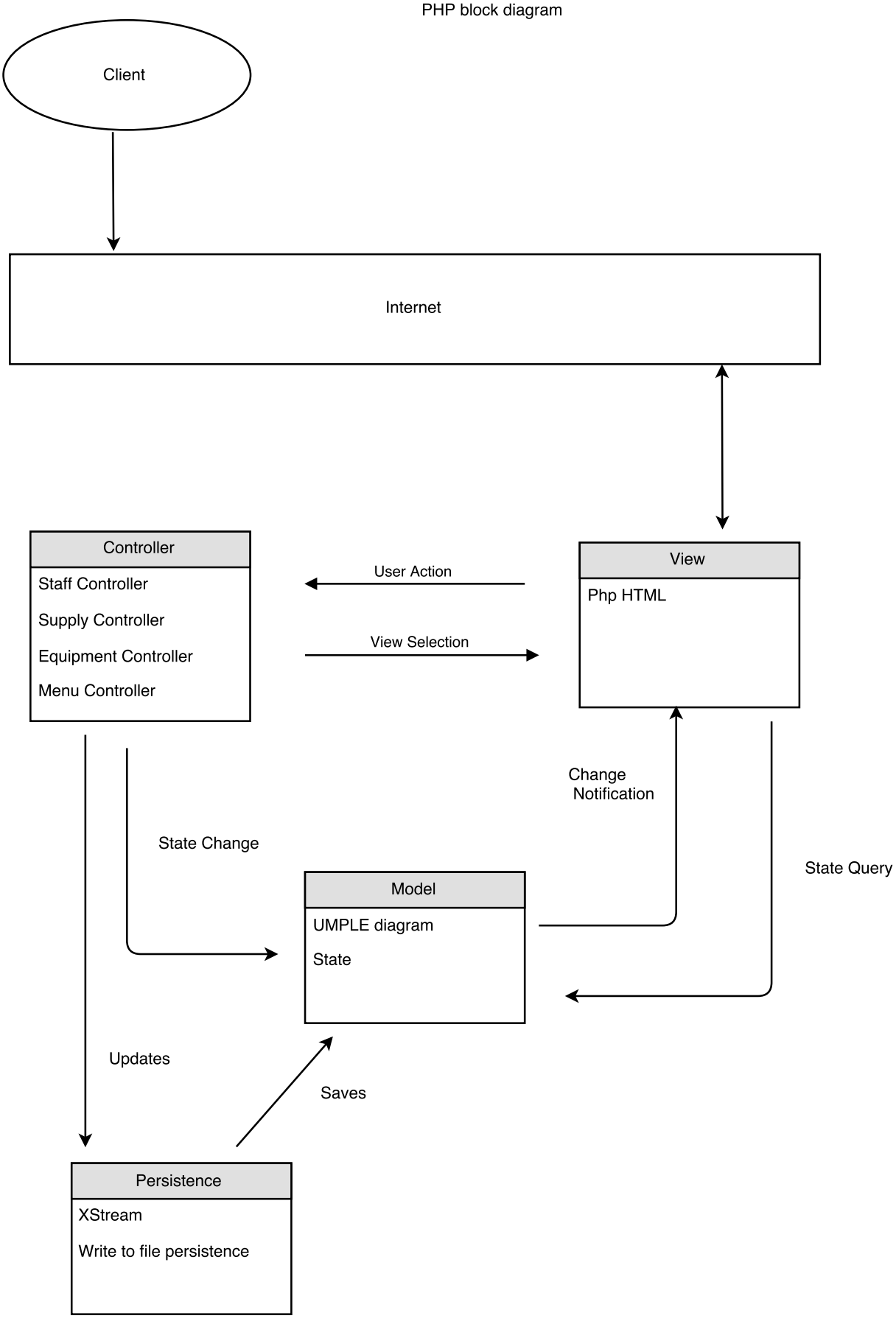

Controller
Staff Controller
Supply Controller
Equipment Controller
Menu Controller
User Action
View Selection
View
Java Swing
Android Activities
Change
Notification
State Change
State Query
Model
UMPLE diagram
State
Updates
Saves
Persistence
XStream
Write to file persistence

PHP block diagram



Description of Architecture

Discussion/Rationale:

This system uses the MVC architectural pattern. This style is being applied because the design of this application is very similar to that of assignment 1 and so we decided that this would be the best option. The same architecture is used in the desktop, mobile and web applications. The only difference is the view based on the platform.

The user manipulates the view and this sends the user interactions to the controller that then executes the appropriate functionality. After that, the controller updates the model. If an error were to occur the controller would update the view to communicate this and not update the model. If there are no errors, the controller saves the changes to the model to the persistence layer. The view then gets its current state from the persistence layer. For example, is an employee was added, the model state was changed and this new employee has been saved to the persistence layer. The view needs to access persistence to give the user the list of current employees if the user wishes to then do something to that employee's data.

Other architectures styles were not applied for the simple reason that MVC was best suited to this situation. Layered architecture would be a pain to implement mainly due to the way that all the data is shared. Therefore, the repository architecture wasn't applied either. Finally, pipe and filter was not used because the processing of information in this case was simple enough to be handled by MVC. Pipe and filter is useful for more complex systems where several steps of data processing occur and that is not the case here. Client server architecture however is being used for the PHP web application in conjunction with the MVC style. Client server is being used due to the nature of PHP. Since it is a web application, it makes sense to use that architecture.

The way that the web application is going to work is like the way the rest of the system works. However, the client will connect to the server and then have access to the application. So, the internet server will be the layer that connects the user to the view. In the block diagram, this is shown.