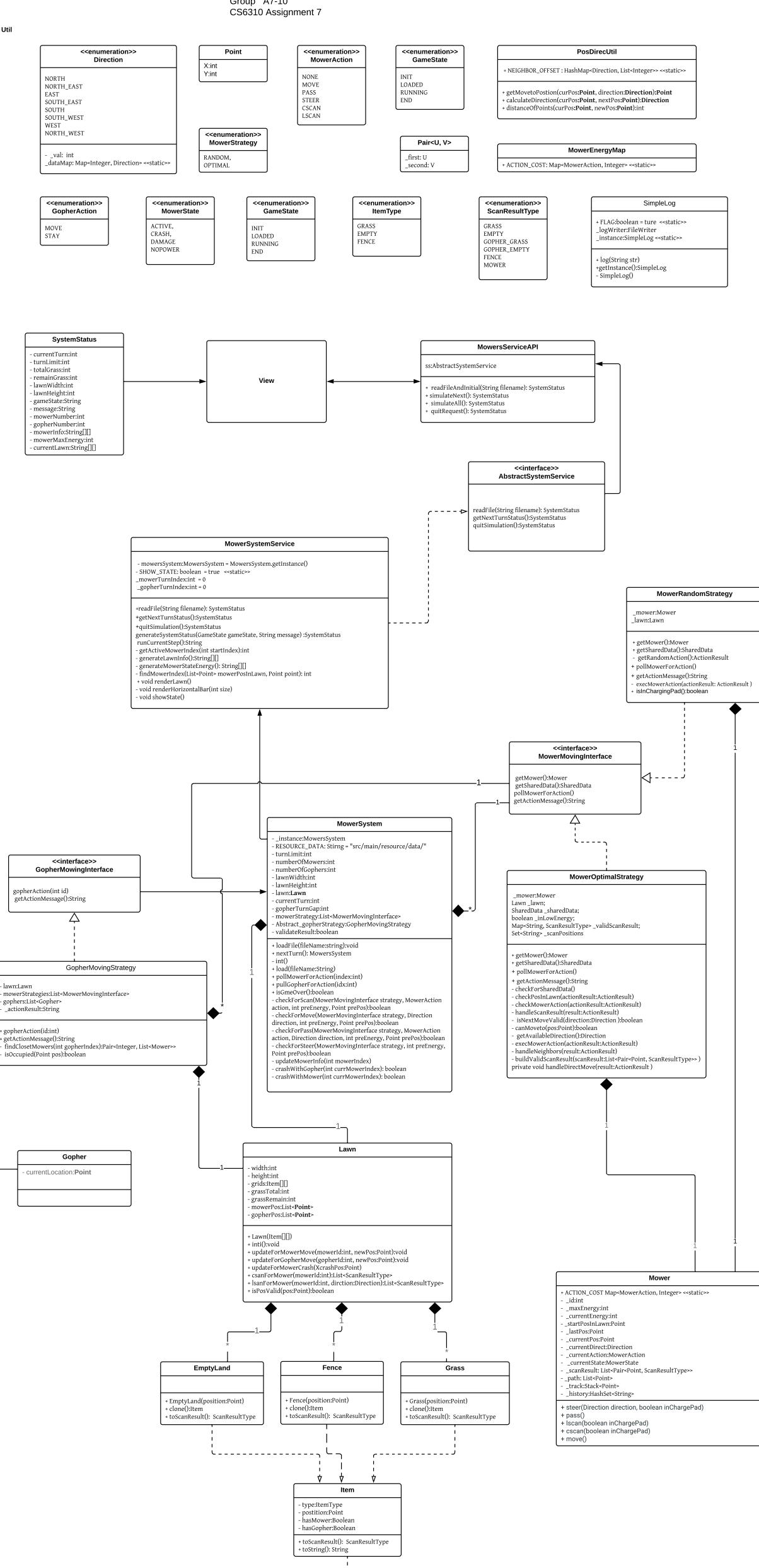
Enum and Util

MVC

Main class

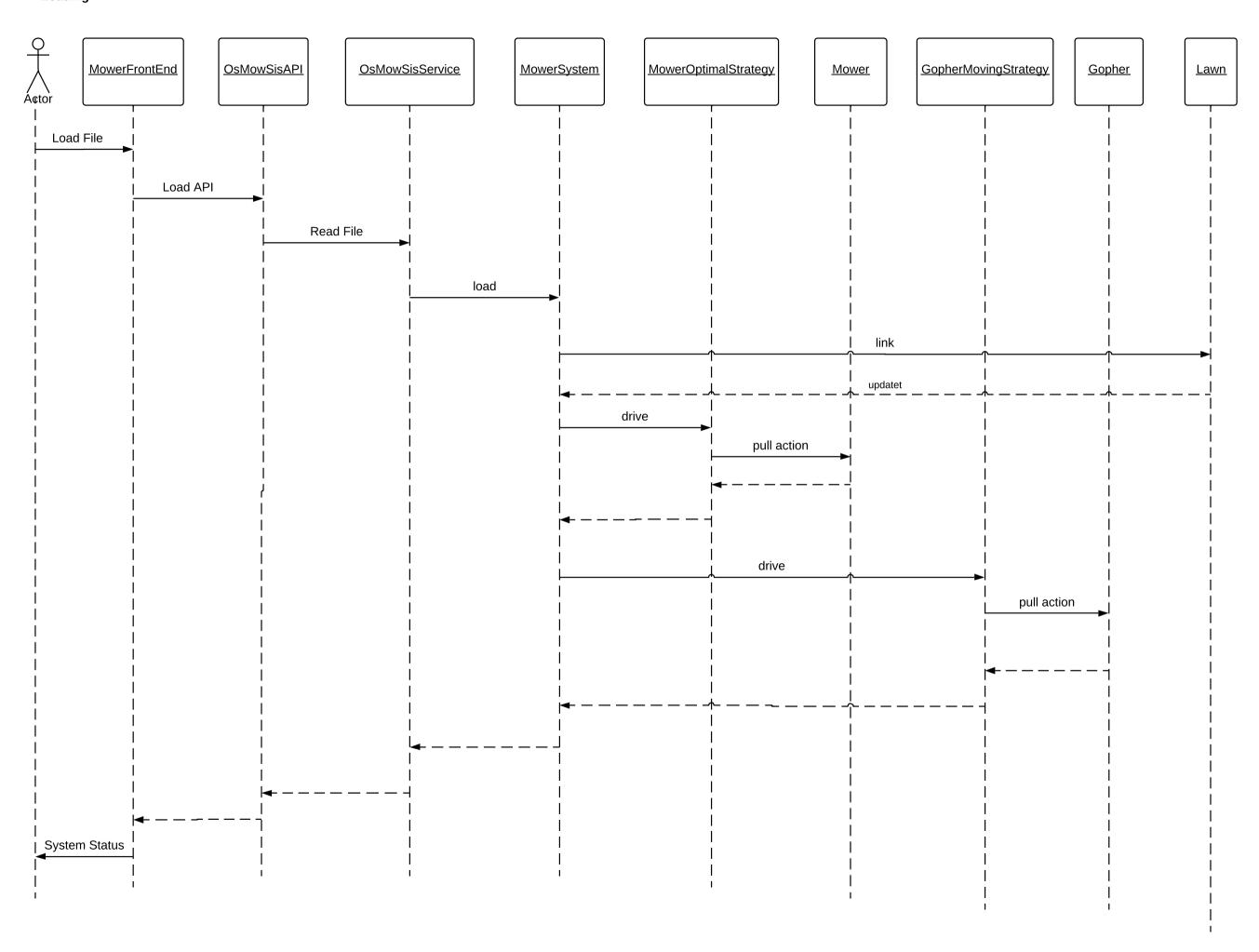


<<interface>>
Cloneable

OsMowSis Class Model (uml 2.0)

Group A7-10 CS6310 Assignment 7

Sequence Diagram 1: Loading



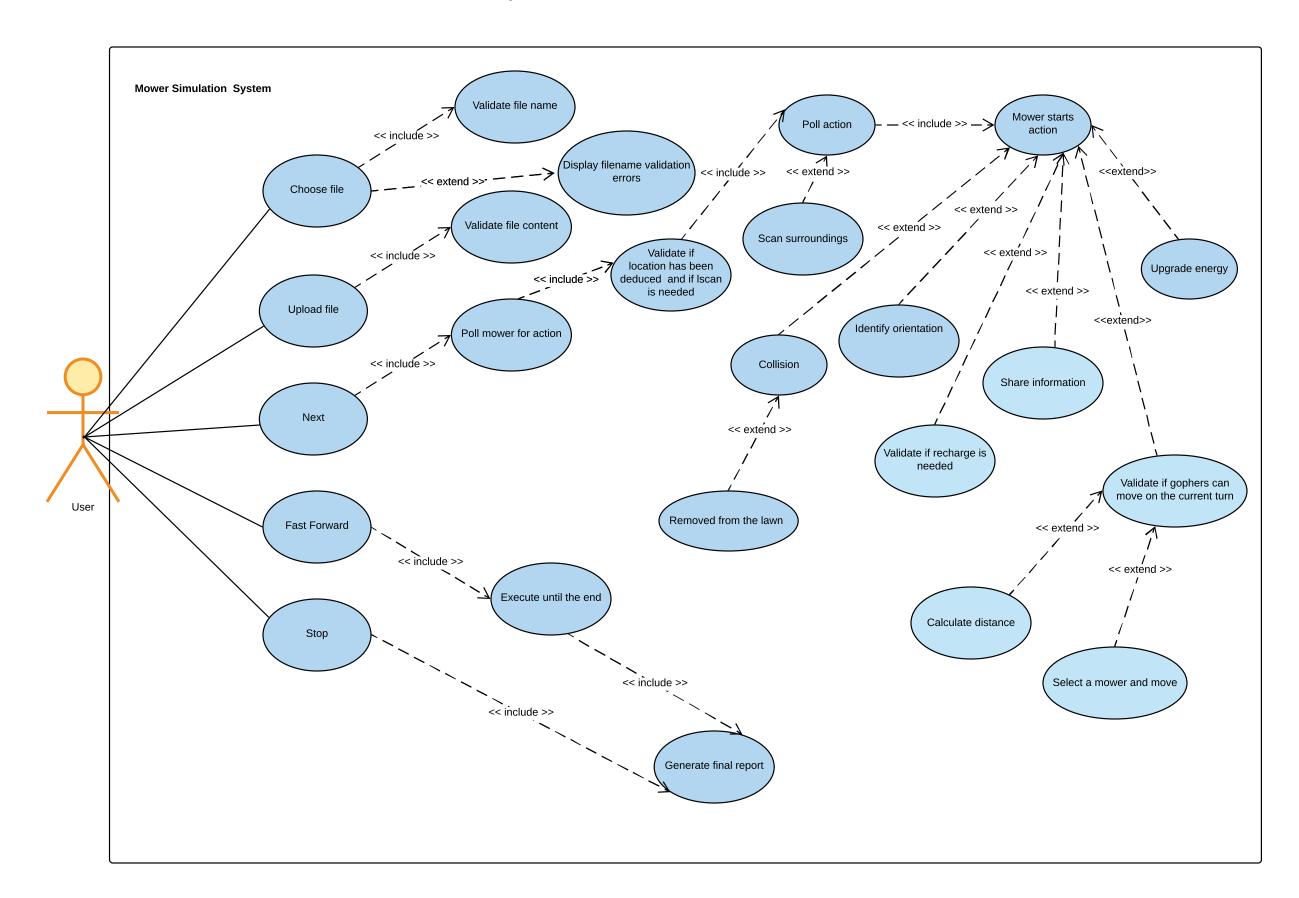
OsMowSis Class Model (uml 2.0) Group A7-10 **Sequence Diagram 2:** CS6310 Assignment 7 Next $\underline{\mathsf{MowerFrontEnd}}$ <u>OsMowSisAPI</u> $\underline{\mathsf{OsMowSisService}}$ request request Due to space limit, continue with following getNextTurnStatus diagram System Status $\underline{\mathsf{OsMowSisService}}$ <u>MowerSystem</u> <u>MowerStrategy</u> Mower x <u>SharedData</u> <u>GopherMovingStrategy</u> Gopher y <u>Lawn</u> getNextTurnStatus Alternative [Game over] generateSystemStatus pollGopherForAction gopherAction getMowerPos Alternatiye setCurrentLocation [Gover's turn] Calculate the current gopher index updateForMo\(\formalfor\)erCrash Alternative [Find the active mover 1 step away, then crsh the mower] updateForGopherMove getMower setCurrentState(CRASH) updateForMowerCrash [E|se, calculate the direction to closet mower then move to it] updateForGopherMove getActionMessage getGopherActionResult pollMowerForAction pollMowerForAction getPosInLawn [Mower's turn] Calculate the current active mover index Alternative [If already know abolute position in lawn] registerMower [Else, it may use steer and lcan to get (x, y) step by step] Iscan setLScanResult Alternative [If need cscan based on sharedData and history infor] setCScanResult Iscan [Els**∉**] Steer/move based on strategy getCurrentPos Alternative [Val|dation of mower action is successful] updateForMower [Valldation fails] -setCurrentState(CRASH)-

-getMowerActionResult—►

getActionMessage

OsMowSis Use Case Diagram

Group A7-10 CS6310 Assignment 7



OsMowSis Deployment Diagram Group A7-10 CS6310 Assignment 7

