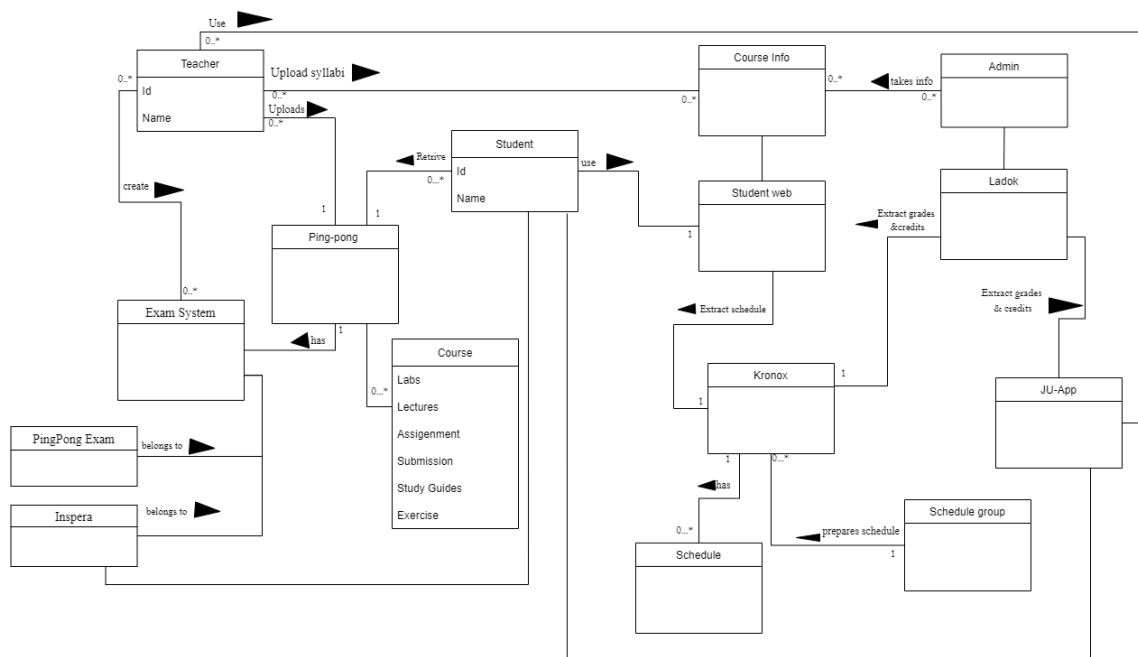Domain Model

1. Noun analysis: - student, admin, teacher, course, student web, kronox, course info, JU-App, ladok, submission, labs, lectures, assignment, exercise, study grades, Ping-Pong, Inspera, Ping-Pong-exam, exam system, schedule, course info, schedule group and course syllabi

2. Classes and attributes: -
   a. Classes → student, admin, teacher, Ping-Pong, course, student web, kronox, ladok, course info, Ju-app, digital exams, schedule
   b. Attributes → Submission, labs, lectures, assignment, exercise, study guides,

3. Domain Model

4.
   4. Ping-Pong is the most important part that supports the process.

Part B

Design model

Board
xsize: int
ysize: int
+ getXSize(): int
+ getYSize(): int
+ containSquare():boolean

ChessGame
board: Board
nextTeamToMove: Team
+ getNextTeamToMove():Team
+ movePieceTo() : boolean
+ changeTurn() : void
+ getPieceInSquare(): Piece
+ isSquareFree() : boolean

ChessView
-chess: Chess
-selectedPiece:Piece
+ mousePressed(): void
- getSquareSize():int
- boardXToViewX():int
- boardXToViewY(): int
- viewXToBoardX():int
- viewYToBoardY():int
+ paint(): void

Piece
team: Team
chess : Chess
xPosition: int
yPosition: int
#pieceStrategy: PieceStrategy
+ turnInto(): void
+hasPosition(): boolean
+shouldTurnIntoOtherType(): boolean
+getLetter(): char

Move
dx: int
dy: int
+ getDestination: Position
+ getDx(): int
+ getDy(): int

PieceStrategy
chess:Chess
piece:Piece
+ getLetter():char
+ shouldTurnIntoOtherType():boolean

Position
- x: int
- y: int
+ getX(): int
+ getY(): int
+ equals(): boolean

**<<enum>>
Team**
+White
+ Black

BishopStrategy
moveDirections: Move
+ getLetter(): char

KingStrategy
- moves: Move
+ getLetter(): char

KnightStrategy
- moves: Move
+ getLetter(): char

PawnStrategy
+ isPawnOnStartPosition(): boolean
+ shouldTurnIntoOtherType(): boolean
+ getLetter(): char

QueenStrategy
+ rockStrategy: RockStrategy
+ bishopStrategy: BishopStrategy
+ getLetter(): char

RockStrategy
- moveDirections: Move
+ getLetter(): char