**Calendar**

**data structures**

// ======= Application data structures ========

// ======= Application data structures ========

// ======= Application data structures ========

// ======= grid data =======

const times = [];

const dates = [];

const rooms = [];

const data = [times, dates, rooms, sessions];

// ======= session data =======

{

session\_id:\_,

room\_id:\_

session\_title:\_

session\_start:\_

room\_name:\_

}

// ======= sequence data =======

{

seq:\_,

abs\_id:\_

abs\_title:\_

}

// ======= cellData =======

{

R\_C: {

id:\_, // behavior

addr:\_,

cellComp:\_,

cellType:\_, // swap data

className:\_,

sessionData:\_,

x:\_, // position data

y:\_,

w:\_,

h:\_

}

}

// ======= cellDataObj =======

{

R\_C: { cellData },

R\_C: { cellData },

R\_C: { cellData }

...

}

// ======= moveData =======

{

cellType:\_,

className:\_,

sessionData:\_

}

// ======= swapCells =======

{

startCellId: "R\_C"

targetCellId: "R\_C"

}

// ======= Application Algorithms ========

// ======= Application Algorithms ========

// ======= Application Algorithms ========

====== makeGridComps() =======

dataIn: { dates:\_ rooms:\_ times:\_ }

dataOut: { roomCells:\_ emptyCells:\_ sessionCells:\_ }

====== buildCellDataObj() =======

dataIn: cellIdsArray // [ "2\_1", "2\_2", "2\_3" ... ]

cellData // id:\_, addr:\_, cellComp:\_,

cellType:\_, className:\_, sessionData:\_,

x:, y:\_, w:\_, h:\_

====== swapStartTarget() =======

dataIn: startCellId // R\_C

targetCellId // R\_C

moveData\_start // cellType:\_, className:\_, sessionData:\_

moveData\_target // cellType:\_, className:\_, sessionData:\_

====== getEmptyHi() =======

loop target to top (stop on first empty)

====== getEmptyLo() =======

loop target to bottom (stop on first empty)

====== shiftCellData() =======

dataIn: upORdown // if up => getEmptyHi(), if down => getEmptyLo(),

targetCellId // 2\_1

emptyHi // getEmptyHi() => none

emptyLo // getEmptyLo() => 2\_6

shiftCellsArray // up: []

down: [2-1, 3-1, 4-1, 5-1]

shiftData // cellType:\_, className:\_, sessionData:\_ // (source: nextCell)

process: hiData => loData

dataOut: moveData => shiftCellsArray cells

====== updateCellComponents() =======

styles

text