**JSON files**

Some parameters are added “on the fly” and not initially initiated

**papers.json** {

* paperID :
* paperTitle :
* abstract :
* authors : [authorIds…]
* presenterID :
* pdfLink :
* review : [

{

- reviewerID

- evaluation

-contribution

-stengths

-weaknesses

},

{

- reviewerID

- evaluation

-contribution

-stengths

-weaknesses

}]

}

**schedule.json** {

-scheduleID

-date (not in session since all sessions in same schedule have same date, derived from conference)

-sessions :[sessions…]

Each session: {

-sessionID

-title

-location (from Locations.json)

-time

}

}

**Locations.json** {

-code

-building

-room

}

**conference-dates.json**{

-day (1,2,3…)

-date

-scheduleID

}

**users.json** {

-id

-firstName

-lastName

-email

-password

-role

}

**USE CASES and functions**

userType (global Variable) >> per default guest/visitor

login will change the userType variable & linked with if functions for each use case (include error case “sorry you’re not allowed”, or don’t show if not allowed user type)

**generic functions (to put in main js & exported in each js to be used)**

lookForUser

loadPage

Use Case 1 (Login)

export function checkUserType(userType){

//check if the actual user is eligible

// this function needs to be called for review, submission, organization of schedule

}

Use Case 2 (submit)

Assuming authors are already registered

Use case 5: (get conference schedule)

When selecting a date (dropdown from conference-dates.json)

* display the day schedule (including all sessions)