**Data Plan**

**Client - Server interactions**

Landing page:

GET to server:

* Get homepage

Response:

* Statistically send page by server

Log in/sign up:

POST to server with AJAX, send JSON object:

* Information from forms or APIs for logging in

Response:

* User profile object that matches the login, or a failure code/message if no matches. Create user session.
* Take user to user homepage and client loads the following from user object:
  + Fill the side panel with boards the user has
  + Fill the homepage with projects the current board has
  + Fill the homepage with tasks the project has
  + Notification of board invitations from managers.
  + Notification of join requests from employees
  + Notification of day off request.

Accept board invitation:

GET to server with AJAX

Process:

* Adds user to the board’s employee objects array. Adds board to the user’s board objects array

Response:

* User JSON object.
* Client populates side panel with the newly accepted board

Join board:

POST to server with AJAX, send JSON object:

* Board link: string
* User object

Process:

* Adds user object to the board’s join requests array.

Response:

* If no match of link on server, send error code. Otherwise, send Success code.
* Client shows message if join request was sent or not.

Add member:

POST to server with AJAX, send JSON:

* A user’s email: string
* The board’s link

Process:

* Adds board to the user’s board\_invitations array

Response:

* If no match of email on server, send error code. Otherwise, send Success code.
* Client shows message if successfully sent or not.

Add task tag:

POST to server with AJAX, send JSON:

* Task tag name: string
* Board link: string

Process:

* Add new task tag object to board’s task tag objects array

Response:

* Task tag JSON object of the board
* Client populates task window with the new task tag

Bulk add task:

POST to server with AJAX, send JSON object:

* Project name: string
* Task name:
* Tag: string
* Person(s) responsible: array of strings
* Priority: string
* Dates: array of dd:mm:yyyy formatted strings
* Start time: 24 hour format string
* end time: 24 hour format string

Process:

* Add new project object to the board’s project objects array across the selected dates.

Response:

* Board JSON object where the tasks are added.
* Client populates pages of relevant dates with the new project along with the new task. If project already exists, add task to that project.

Change availability:

POST to server with AJAX, send JSON object:

* Day: string
* Start time: 24 hour format string
* end time: 24 hour format string

Process:

* In user’s availability objects array, change the values of the availability object that corresponds with the board

Response:

* Check against user master availability if the requested project availability don’t clash. If no send the user’s new availability object. If clashing, send error code.
* Client displays the updated values if successful. Shows error if not.

Submit day off:

POST to server with AJAX, send JSON object:

* Date: dd:mm:yyyy formatted string
* User email: string

Process:

* Add new day-off object to the board’s day off requests objects array.

Response:

* Success code.
* Client populates div with the “submission pending” and shows message notifying request was sent successfully.

Change task tag performance ability:

POST to server with AJAX, send JSON object:

* Task tag name: string
* Ability: string

Response:

* User’s task tag performance JSON object.
* Client updates the task tag to show the correct ability.

Add board:

POST to server with AJAX, send string:

* Board name: string

Process:

* Add a board object to the user’s board objects array

Response:

* The new board JSON object
* Client populates the homepage with the details of the new board. These are expected to be empty.

Add project:

POST to server with AJAX, send JSON object:

* Project name: string
* Date: dd:mm:yyyy formatted string

Process:

* Add new project object to the board’s project objects array

Response:

* The new project JSON object.
* Client populates board with the new project table

Add task:

POST to server with AJAX, send JSON object:

* Date: dd:mm:yyyy formatted string
* Project name: string
* Task name: string
* Tag: string
* Person(s) responsible: array of strings
* Priority: string
* Start time: 24 hour format string
* End time: 24 hour format string

Process:

* Add new task object to the project’s task objects array.

Response:

* Project JSON object of the new task.
* Client populates project table with new task row

Profile page:

GET to server:

* Get profile page

Response:

* User JSON object.
* Client populates the page using details from user object

Update profile:

POST to server with AJAX, send JSON object:

* String of the fields that user wants to change

Process:

* Update user object properties

Response:

* JSON of user’s personal info

Change password:

POST to server with AJAX, send JSON object:

* Encrypted object of new password

Process:

* Update user’s password.

Response:

* JSON of user’s personal info
* Client shows success message.

About us, contact us page:

GET to server

* Get about us/contact us

Response:

* About us or contact us HTML

Logout:

POST to server with AJAX, send JSON object:

* String indicating user wants to logout

Response:

* End user session

**Data to be stored on the server**

User

{

    name: 'Carlos',

    email: 'a1731835@adelaide.edu.au',

    password: encrypted,

    birthday: '01-10-1998',

    phone: 11111111111,

    career\_background: 'student',

ID: 12432,

    board\_invitations =[ADSA],  //board objects

    boards = [WDC], //board objects

    availability =[wdc\_availability], //availability objects. availability[0] applies to boards[0] and so on

    task\_tags = [wdc\_tt],  //task\_tag objects

    task\_tag\_performances = [wdc\_ttp],   //task\_tag\_performance objects. task\_tag\_performances[0] applies to task\_tags[0] and so on.

}

board

{

    name: 'WDC',

    link: 'www.carlosproject.com/4w45ewt2bd',

    manager: Carlos //user object

    employees: [Hunter, Tien], //user object

    join\_requests: [Lim],    //user objects

    tasks\_tag: WDC\_tasks, //task\_tag objects

    //different for each date, still have to come up with a calendar system to

//differentiate projects below for each day

    projects=[Milestone\_1], //project objects

}

project

{

    name:'Milestone 1',

    tasks = [Research, Design, Data\_plan]   //task objects

}

task

{

    name: 'Research',

    tag: 'Documentation',    //a value from task\_tag object

    person\_responsible: [Hunter, Tien], //user objects

    priority = {

        high: Boolean,

        medium: Boolean,

        low: Boolean

    },

    status = {

        complete: Boolean,

        in\_progress: Boolean,

        stuck: Boolean

    },

    time: task\_duration //duration object

}

task\_tag

{

    tasks\_tags: ['Documentation', 'Accounting', 'Programming', 'Calculation']

}

task\_tag\_performance

{

    task\_tag\_performances: ['Good', 'Average', 'Needs Improvement', 'Good']

}

availability

{

    Sunday: [sunday\_morning, sunday\_night],     // duration objects

    monday: [monday\_morning, monday\_night, monday\_afternooon],

    tuesday: [tuesday\_morning, tuesday\_night],

    wednesday: [wednesday\_morning, wednesday\_night],

    thursday: [ thursday\_morning,  thursday\_night],

    friday: [friday\_morning, friday\_night],

    saturday: [ saturday\_morning,  saturday\_night],

}

duration

{

    start: '8:00',

    end: '13:00'

}