# NAVPAL FLOORPLAN CREATOR

5/4/2013

## Server Summary

A brief overview of the server structured and what features are implemented.

# NavPal Floorplan Creator

SERVER SUMMARY

## OVERVIEW

The server is built in NodeJS with Express Server, Passport for authentication and Mongoose to interface with the MongoDB server. The server serves the authoring tool, maintain the authoring data, and offers a public API for published buildings.

## CODE

The server is structured in a model and controller architecture where almost every Mongo Document has a model and controller. And the routes can be found in server.js.

## USER

The User object:

```
{
   registeredTimestamp: Date,
   lastLoginTimestamp: Date,
   buildingRefs: Array of building ids and building names,
   imageRefs: Array of Strings
}
```

Each user has a corresponding User object. The username (email) and password are managed by Passport. The buildingRefs and imageRefs are used to determine which buildings and images the user can access.

## BUILDING

The Building object:

```
{
   _creatorId: String,
   userBuildingId: String,
   userBuildingName: String,
   graph: Object,
   authoData: Object
}
```

The creator id is the node object id for the building's owner. The userBuildingId and userBuildingName are used on the client side to access buildings. The authoData is the data uploaded by the client side and isn't checked by the server. The graph is the BuildingGraph object constructed on the client side and isn't checked either.

## REFERENCES

NodeJS: http://nodejs.org/

Express Server: http://expressjs.com/api.html

Passport: http://passportjs.org/guide/

Mongoose: http://mongoosejs.com/

Mongo: installation instructions can be found in the README

Examples: http://kosbie.net/cmu/fall-12/15-237/ , http://kosbie.net/cmu/fall-12/15-237/handouts/notes-server-side-part1.html, http://kosbie.net/cmu/fall-12/15-237/handouts/notes-server-side-part2.html