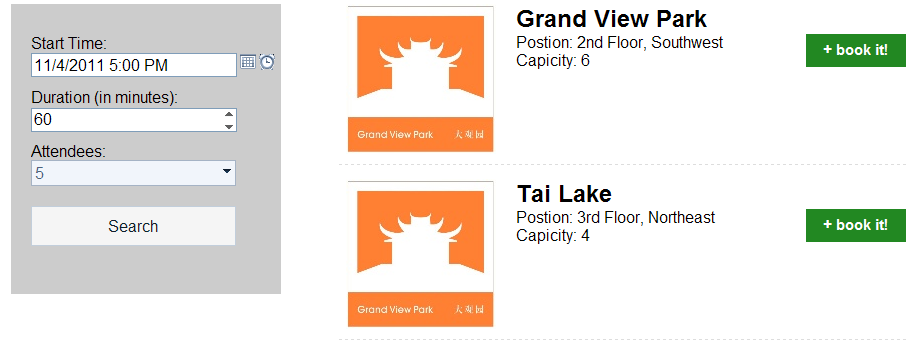


**Regular Search**

This is the most common way you would use to look for an available meeting room. Provide start time, duration and attendees then click “Search”, all available rooms meet for your criterion will be listed as this:



By a benefit of ASP.NET MVC actually you can directly play with URL to get the result you want.

For example, you want to find meeting rooms at 2012/1/1, 10:00:00 AM for 30 minutes with 8 people, just enter:

http://<serveraddress>/bookit/regular/2012-01-01-10-00/30/8

into your browser then results should be displayed.

The scheme is:

http://<serveraddress>/bookit/regular/{starttime}/{duration}/{attendee}

where {starttime} format = yyyy-MM-dd-HH-mm

**OneClick Search**

OneClick search provides you another way to look for a meeting room. It immediately lists all meeting rooms that **will be** available at **anytime** in next xxx minutes. See below:



Hope this feature could help you in such scenarios:

* You want to start a short meeting with a room that will be available as early as possible;
* There’s no room right now but you don’t mind to find one sometime later.

**Book it!**

Click  button after any available room then the browser should prompt you to Open/Save an iCalendar file. Click “Open” then your Outlook should open a new meeting request mail with start/end time you have chosen and the meeting room added as a resource.

Fill in more attendees, Subject, body text then click “Send”. Done!

\*Don’t worry if you see that “Location” text box only displays meeting room name instead of a long messy string when you add a room manually. They have no difference.

**Behind the scenes: Meeting Room Rank**

Perhaps most interesting feature is “Meeting Room Rank”.

If you provide your “Cubicle#” (i.e. seat number in your name plate, something like 2322, 1211 etc.), through upper-right corner on the UI,



Book!t will take the distance between you and the room into consideration when looking for a room.

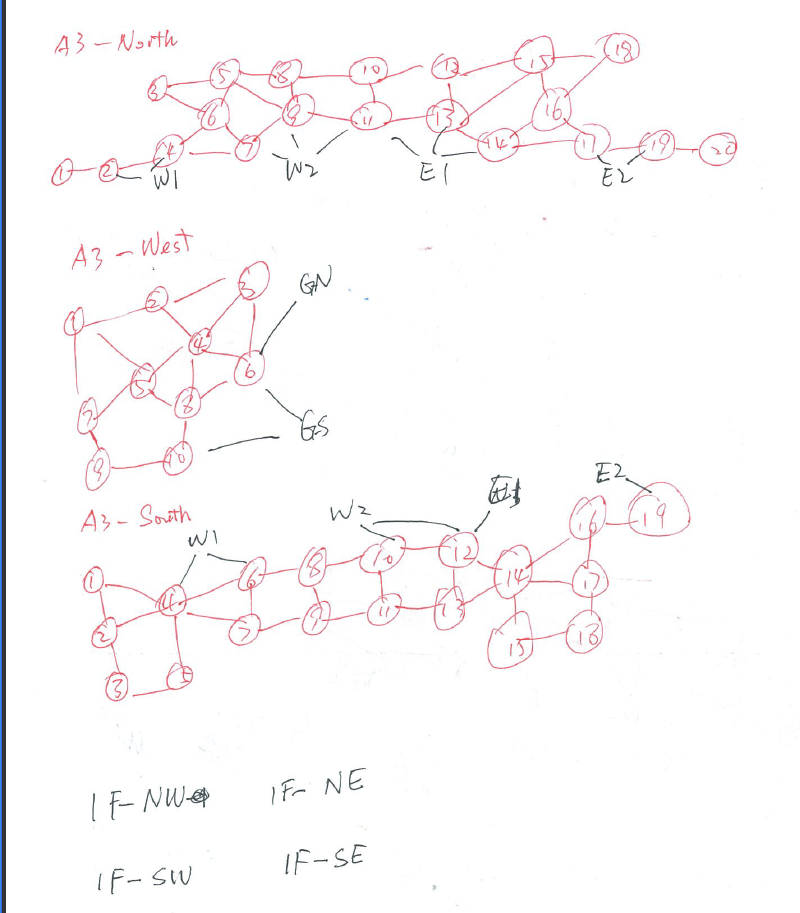
Another consideration is the “Attendee vs. Capacity”.

Actually Book!t will “rank” each result by the following formula:

**Score = distance + (attendee – capacity)2**

then sorts the results by the scores in ascending order.

To implement this feature we actually input all cubicle information and their relationships in ARCA building into the database. At runtime Book!t builds an undirected graph in memory and traverses this graph to find a shortest path between YOU and the meeting room, textbook Dijkstra shortest path algorithm is used here.



To see it really works you can try with following cases (Regular Search on weekend is recommended)

Cubicle# = 2322 (2F EZ near Wang Fu Jing) with attendees = 5.

Cubicle# = 3003 (3F NZ, near Po Yang Lake) with attendees = 5.

Cubicle# = 2322 (2F EZ near Wang Fu Jing) with attendees = 10+.

**Future Development Plan**

* Fix bugs; make it more stable and robust.
* Add more “eye candies” to the UI.
* I consider adding “Who occupied the room” feature into it so that use will have a chance to negotiate with the booker if he/she can let you use the room.