# SpotCafe prototype

2017-08-16

Since most of the features doesn’t have detailed requirements, we are assuming that the most important goal is to have some functional prototype as soon as possible, which implements the core features like enable/disable access to client computers, basic configuration, operator and administrator interfaces. That’s why the estimates will be based on simple versions of the functionality, which means non-shell version of client application, not so “flexible” user interfaces and settings, no “fancy” features like “in-game performance” or “remote desktop control”, no external systems integration, no “cafeteria” etc. After the evaluation of the prototype, all these could evolve to better versions.

Because of the lack of domain data shape, the database will be designed to support the prototype only - information collected during the prototype evaluation could greatly differ compared to what is needed in the real implementation, making data transfer between the two versions impossible. This might require post-prototype data to “start over”. The most knowledge about data and relations in advance – the better.

The prototype and estimates are based on a “non-shell” version of the client application. “Securing” the client computers could be achieved to some extent by using non-administrator Windows user – this will prevent clients of altering important system files and settings. Further restrictions (like hiding C: drive, disabling task manager, command prompt, registry editor etc.) can be applied by using Group Policies. Please refer to Group Policy Editor Tool gpedit.msc and what is possible with it (especially in “User Configuration” section). Each machine could have a disk image that will be applied if something goes wrong. Another option is to use software that virtualizes access to C: disk and drops any changes made when the machine is restarted - this is a very good solution to prevent changes to your system drive which will keep your Windows unbreakable and unchangeable but could require big amounts of RAM if there are a big files copied on the protected drive like big game updates etc. You should not keep your games and their saves on system drive anyway.

# System components

1. Database server running MS SQL Server. MongoDB is also an option but whether it is suited for the application requires more detailed requirements of the data and its relations. The type of the database must be decided at very early stages of application development.
2. Application server running NodeJS - <https://nodejs.org/> . This server will apply system logic and will provide data to client machines as well as operator and admin interfaces.
3. Windows service on the client machines – this is needed only to provide administrative capabilities to the client application. Requires .NET Framework 4.5
4. Client application on the client machines – enables and disables access to client machines based on the information from the application server. Requires .NET Framework 4.5
5. 3D client application – user interface for the clients. Shows available applications, user profile etc.
6. Operator and Admin web applications – web sites for controlling the machines, changing the settings, showing reports etc.

# Prototype features

Most of the features require implementations in multiple system components so they are not grouped by component. Estimate is made based on time needed to implement the feature in all components. There will be an initial time for setting up the environment and software skeleton of the system which would be 24 hours. The time that could be dedicated for implementation per day varies between 3 and 12 hours so feature estimated for 8 hours could be delivered somewhere between 1st and 3rd day after it is started. The amount of features needed for the prototype are subject to change – if some of the must have features are missing, they will be added when requirements are created. Please comment on the description of existing features.

|  |  |  |
| --- | --- | --- |
| Name | Description | Estimate (hours) |
| Server auto-discovery | The goal is effortless client application configuration – you install client application and “forget”. On start-up client application will try to find the server by sending broadcast UDP packets to local network fragment x.x.x.255. Application server will respond to these packets. Client application will receive server response and will use the information for further communication (which includes loading configuration etc.). Requires clients and server to be in the same network fragment (“x.x.x” part of the IP addresses of clients and server to be the same). This feature will also be used from the server in order to add computers to its list with supported machines if the requesting machine is not already in that list. This could later be expanded to also auto-update client applications making maintenance of the client machines close to zero. | 16 |
| Enable/disable access to client machine | Server will control client application whether the machine should be in “enabled” or “disabled” state. Implementation could be simply to show 3D application in “log-in” mode and monitor it to keep it “on-top” of all other applications. When the user enters his credentials (or the operator enables the computer from its web application), the server will send information to “enable” the machine. Then the client application will communicate with 3D application that it should be in “logged in” state. | 24 |
| Log in from client machine | Username and password credentials only. Still needs to be decided how credentials will be shared between client application and 3D application. | 16 |
| Apply client machine restrictions | After server discovery, the client could receive restrictions from the server which must be applied to the machine. They could be simple .reg file merged by executing Windows’ reg.exe utility with appropriate command line arguments since most (if not all) of the (group policy) restrictions use the Windows registry. Generation of the .reg file content is not part of the estimation. An admin page is needed in order to upload .reg content to server. | 16 |
| Create accounts | A single page in admin interface for editing information about the account like username, password and access group (role). Access groups (roles) with their permissions will be in separate page. That way single set of permissions grouped in access group would be assigned easy to employees with same roles. | 24 |
| Roles (access permission groups) | A page where a new roles could be edited. Each role will enable certain permissions creating permission sets that can later be applied to different type of users – admins, operators, clients etc. For simplicity permissions could be of type “No access”, “Read only” and “Full access” for each entity. For example the interface for the entity “Users” could look like a table row with 4 columns:   |  |  |  |  | | --- | --- | --- | --- | | Users | (o) No access | ( ) Read only | ( ) Full access |   With default selection “No access”. Administrator would select one of the three possibilities. This feature requires showing/hiding/disabling specified buttons/links on the web application interface if specified permission is not granted as well as permission checks on the server operations. Please note that for some of the entities / parts of the application logical permission could be only two – “No access” and “Read only” like reports – they are an information that is calculated and are only “viewed” - one cannot have “Full access” to something that can be only “visible” or “invisible”. Please comment on that idea of “three types of access” permissions. | 40 |
| Computer groups | A page where groups can be created and edited and computers will be added/removed from them. Also will allow editing hourly price for the group. | 16 |
| Additional settings | Various settings like “Shutdown/Restart/Log out after X minutes of idle time”. | 16 |
| Remote file explorer | Client application will be able to provide information to the server about drives, folders and files on its hard disk. This functionality is needed for “Application management” feature. | 32 |
| Application management | A page where application profiles will be edited. Each profile will contain one or more applications. The application executable file will be selected from the client machines using “Remote file explorer” feature. Later application profiles will be assigned to computer groups. When user logs in using 3D application, server will send it list with applications that are allowed. | 24 |
| Reporting | A page where some simple reports could be shown. Prototype will support only table-based reports. Supported reports will be “Totals per day”, “Totals per employee” and “Totals per client”. | 32 |
| Pricing | Prices per hour for user group or computer group. Single currency only. Setup of negative balances per group. |  |
| Operator views | A page with list view of the state of client computers, latest transactions, recent log (what is this?) | 40 |
| Remote manager | A page that will show started applications on selected client computer with the ability to “kill” them. Will also support user log off, shut down, restart. | 24 |
| Other | All other necessary features to support the system like log-in / log-out pages for employees, configuration of the database connection etc. not mentioned above. Can’t really have an estimate on this. |  |