Prezentarea tehnologiilor folosite:

1. Java

Java este un limbaj de programare de nivel înalt complet orientat pe obiecte – elimină complet stilul de programare procedural pe baza caruia sunt construite o serie de platforme(seturi de librării) destinate implementării de aplicații pentru toate segmentele industriei software.

Am ales acest limbaj deoarece oferă un avantaj destul de mare dat de execuția extrem de rapidă și totodată dispune de o interfaţă de programare a aplicaţiilor foarte bogată, precum şi de un ecosistem cu sursă deschisă extrem de ajutător.  Java vine, aşadar, la pachet cu tot felul de unelte care ne ajută indiferent de produsul software pe care vrem să-l dezvoltăm.

Două dintre elementele principale utilizate în aplicația noastră este JavaFX, o platformă de programare bazată pe Java care ne-a pus la dispoziție o mulțime de elemente folositoare în construcția interfeței cu utilizatorul și Gradle care este un utilitar ce ne-a permis gestiunea avansată a construirii aplicației și ne-a oferit suport pentru configurarea și descărcarea automata a bibliotecilor și a altor dependențe de care am avut nevoie.

2. Spring Remoting

Tehnologia Java Remote Method Invocation (RMI) permite unui obiect

dintr-o mașină virtuală Java să apeleze metodele unui obiect dintr-o altă

mașina virtuală Java.

RMI furnizează mecanismul prin care serverul și clientul comunică și își transmit informații.

Frameworkul Spring suportă diferite modele de apeluri la distanță printre care și Java RMI. Serviciile (obiectele remote) sunt configurate folosind

bean-uri Spring (obiecte create folosind containerul Spring).

Se folosește un “proxy factory bean” care permite injectarea dependențelor

către obiecte remote ca și cum ar fi obiecte locale.

• Clientul apelează metodele de la proxy ca și cum acesta ar furniza

funcționalitatea cerută.

• Proxy-ul comunică cu obiectul remote în numele clientului. Se ocupă de

toate detaliile conexiunii și a apelului metodelor la distanță.

Aplicația curentă a fost implementată pe modelul client – server, astfel am ales să folosit Java RMI folosind Spring pentru pentru că oferă o modalitate mult mai elegantă de crea și a folosi obiectele remote RMI.