

CA

```
/**      vytvori novy certifikat pre zadane meno a verejny kluc
 *      ako vystup vracia certifikat - meno,a verejny kluc zasifrovane sukromnym klucom authority
 * @param name meno pre ktore bude certifikat vystaveny
 * @param publicKey verejny kluc uzivatela ktory o certifikat ziada
 * @param output vystupny buffer obsahujuci certifikat
 * @return 0 ak v poriadku prejde inak error code
 */
int newCertificate(string name, string publicKey, string* output);
```

SERVER

```
/**      vytvorenie noveho uctu
 *      server overi dostupnost zvoleneho loginu, desifruje certifikat verejnym klucom authority,
 *      overi ci je certifikat vystaveny k zadanemu menu a vytvori novy ucet obsahujuci login, heslo,
      meno majitela a verejny kluc ziskany z certifikatu
 * @param login pozadovany login pre ucet
 * @param password pozadovane heslo pre ucet
 * @param name meno majitela uctu
 * @param certificate certifikat s klucom vystavenym na meno majitela uctu
 * @return 0 ak v poriadku prejde inak error code *
 */
int newRegistration(string login, string password, string name , string certificate);
```

```
/** autentizuje uzivatela podla zadaneho loginu a hesla
```

```
 * @param login login pre prihlasenie
 * @param password heslo pre prihlasenie
 * @return 0 ak v poriadku prejde inak error code
 */
```

```
int userAuthentication(string login, string password);
```

```
/** vrati zoznam uzivatelov
```

```
* @param login login pre prihlasenie
```

```
* @param users zoznam uzivatelov
```

```
* @return 0 ak v poriadku prejde inak error code
```

```
*/
```

```
int getUsersList(string login, List<string>* users);
```

```
/** umozni uzivatelom navzajom sa overit a vymenit kluc pre symetricku sifru
```

```
* @param requestingUserLogin ziadatel o spojenie
```

```
* @param targetUserLogin prijemca spojenia
```

```
* @return 0 ak v poriadku prejde inak error code
```

```
*/
```

```
int establishConnection(string requestingUserLogin, string targetUserLogin);
```

```
/** odpoji uzivatela od servra
```

```
* @param login uzivatel
```

```
* @return 0 ak v poriadku prejde inak error code
```

```
*/
```

```
int disconnectUser(string login)
```

```
CLIENT
```

```
/** umozni klientovi poziadat autoritu o certifikacu kluca
```

```
* @param name meno pre ktore bude certifikat vystaveny
```

```
* @param name meno pre ktore bude certifikat vystaveny
```

\* @param publicKey verejny kluc uzivatela ktory o certifikat ziada

\* @param output vystupny buffer obsahujuci certifikat

\* @return 0 ak v poriadku prejde inak error code

\*/

int requestKeyCertification(string name, string publicKey);

/\*\* umozni klientovi poziadat o vytvorenie uctu

\* @param login pozadovany login pre ucet

\* @param password pozadovane heslo pre ucet

\* @param name meno majitela uctu

\* @param certificate certifikat s klucom vystavenym na meno majitela uctu

\* @return 0 ak v poriadku prejde inak error code

\*/

int requestRegistration(string login, string password, string name , string certificate);

/\*\* umozni klientovi poziadat o prihlasenie

\* @param login login pre prihlasenie

\* @param password heslo pre prihlasenie

\* @return 0 ak v poriadku prejde inak error code

\*/

int requestAuthentication(string login, string password);

/\*\* umozni klientovi vyziadat si zoznam uzivatelov

\* @return 0 ak v poriadku prejde inak error code

\*/

int requestUsersList(string login);

```
/**      umozni clientovi poziadat o vytvorenie spojenia
 * @param login ziadatel
 * @param target ciel
 * @return 0 ak v poriadku prejde inak error code
 */
int requestConnection(string login, string target);
```

```
/**      umozni uzivatelovi odoslat data
 * @param login odosielatel
 * @param target prijemca
 * @param data prud s datami
 * @return 0 ak v poriadku prejde inak error code
 */
int sendData(string login, string target, stream data);
```

```
/**      umozni uzivatelovi odpojit sa od servra
 * @param login uzivatel
 * @return 0 ak v poriadku prejde inak error code
 */
int requestDisconnect(string login);
```