Arhitekturni projekat

Tema: Algoritmi za kriptovanje i dekriptovanje

Student: Lazarević Nina

Broj indeksa: 15187

1. Cilj ovog dokumenta

Cilj ovog dokumenta je opis arhitekture rada aplikacije za zaštitu informacija.

2. Opseg dokumenta

Dokument se odnosi na aplikaciju za kriptovanje i dekriptovanje razlicitih algoritama čiji je zadatak da na sto efikasniji izvrse kriptovanje/dekriptovanje fajla koji odabere korisnik ili proizvodljnog teksta unetog sa tastature.

3. Predstavljanje dokumenta

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

4. Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. Aplikacija će biti implementirana kao desktop aplikacija u programskom jeziku C#
2. Aplikacija mora pružiti zadovoljavajuće performanse pri bilo kojoj operaciji.

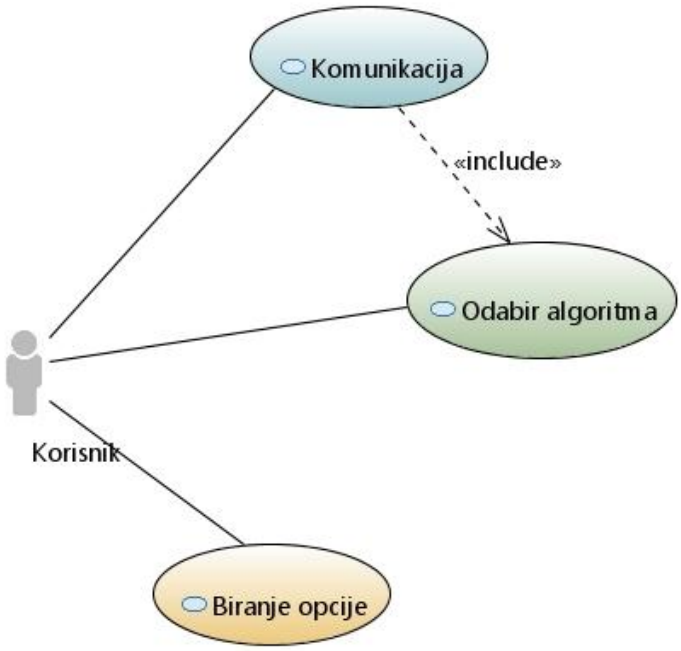
5. Pogled na slučajeve korišćenja

Slučajevi korišćenja su:

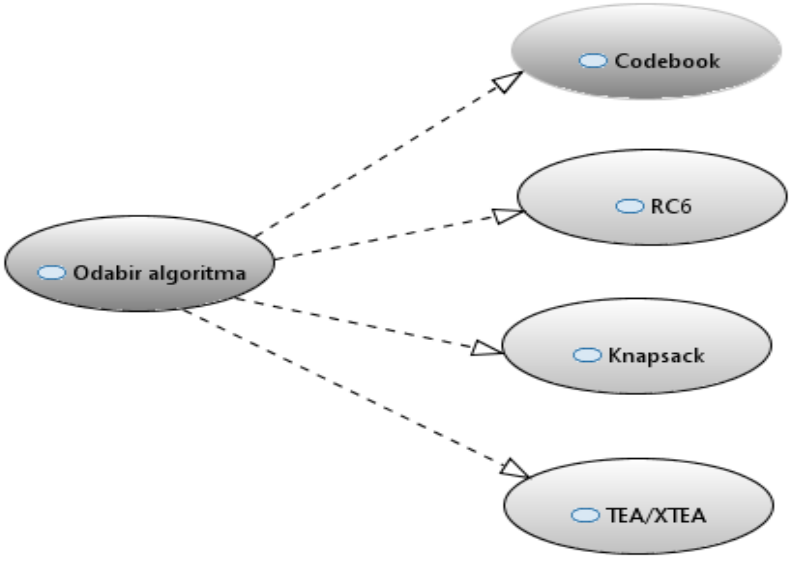
- Otvaranje i odabir zeljenog algoritma
- Biranje opcije za kriptovanje/dekriptovanje
- Unos teksta za kriptovanje preko tastature ili ucitavanje fajla
- Unos teksta za dekriptovanje preko tastature ili ucitavanje fajla
- Unos kljuca i fajla za kodiranje i fajla u kome ce se snimati kodirani podaci kod RC6 algoritma
- Unos kljuca i vektora iv kod TEA\XTEA algoritma
- Generisanje kljuceva i superrastuceg niza kod Knapsack algoritma

5.1 Dijagrami slučajeva korišćenja

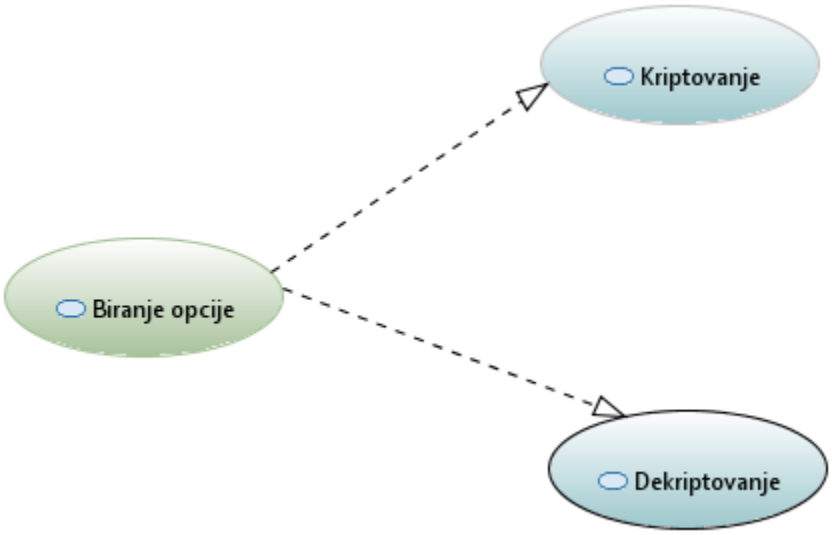
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja aplikacije MMS prikazan je na sledećoj slici:



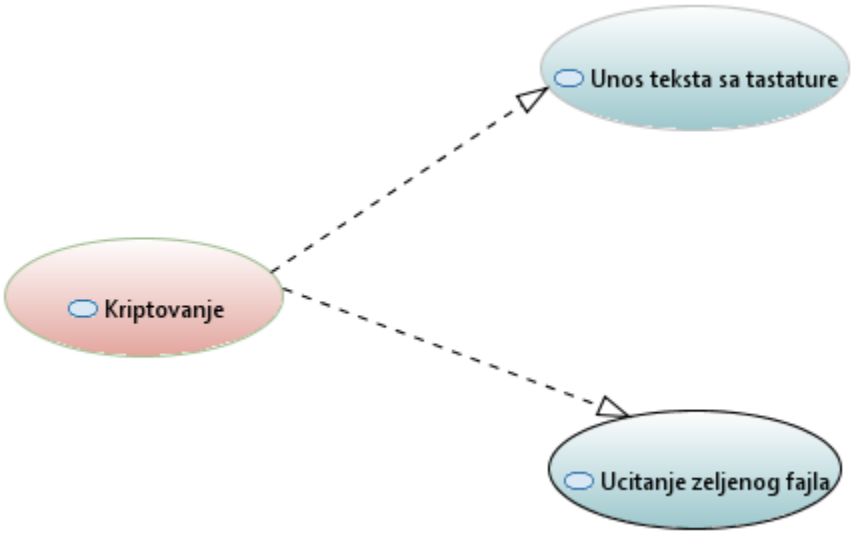
Detaljni UML dijagram za slučaj korišćenja *biranje algoritma* je prikazan na sledećoj slici:



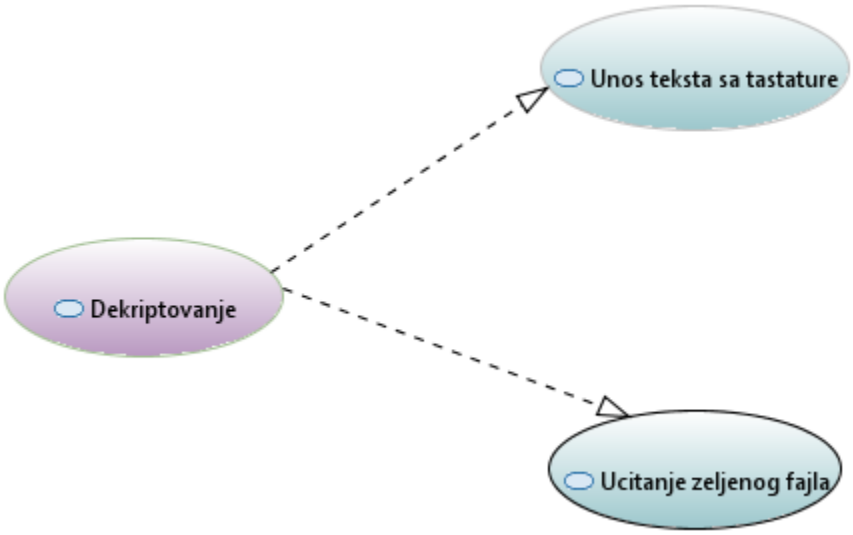
Detaljni UML dijagram za slučaj korišćenja *biranje opcije za kriptovanje ili dekriptovanje* je prikazan na sledećoj slici:



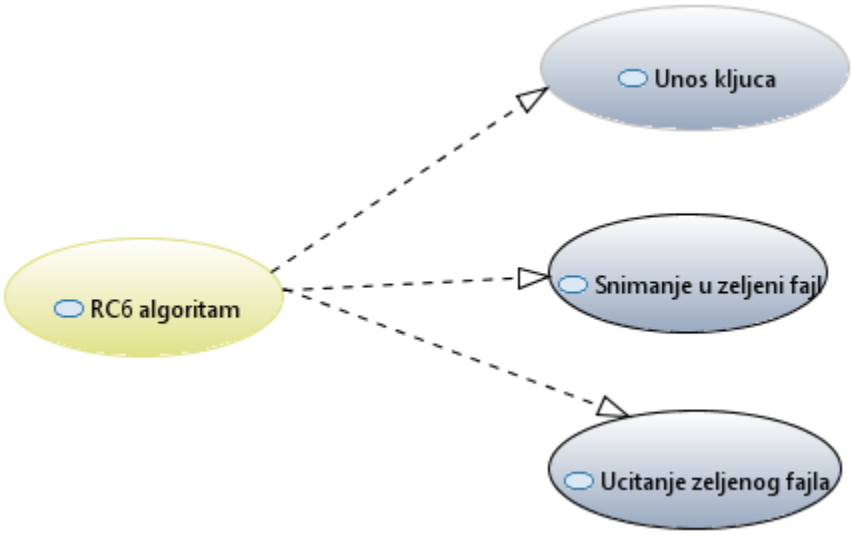
Detaljni UML dijagram za slučaj korišćenja *kriptovanja* je prikazan na sledećoj slici:



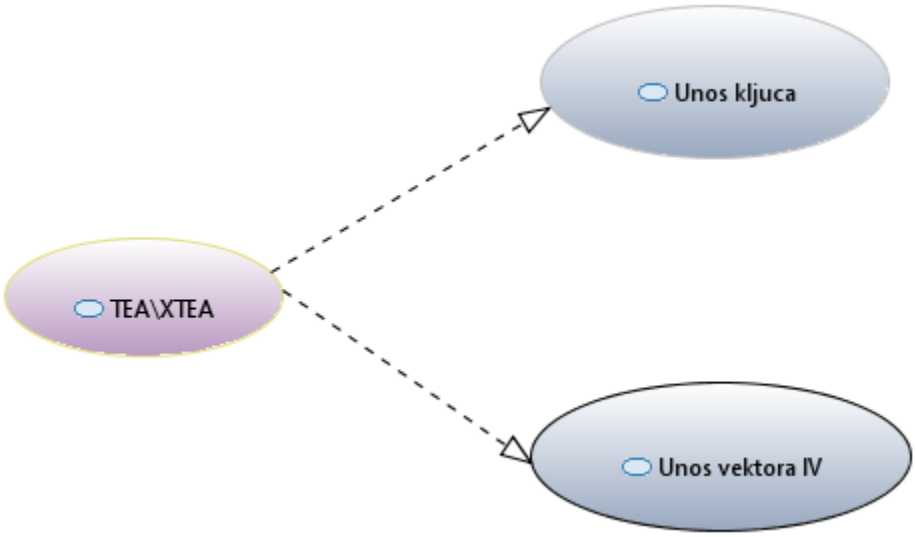
Detaljni UML dijagram za slučaj korišćenja *dekriptovanja* je prikazan na sledećoj slici:



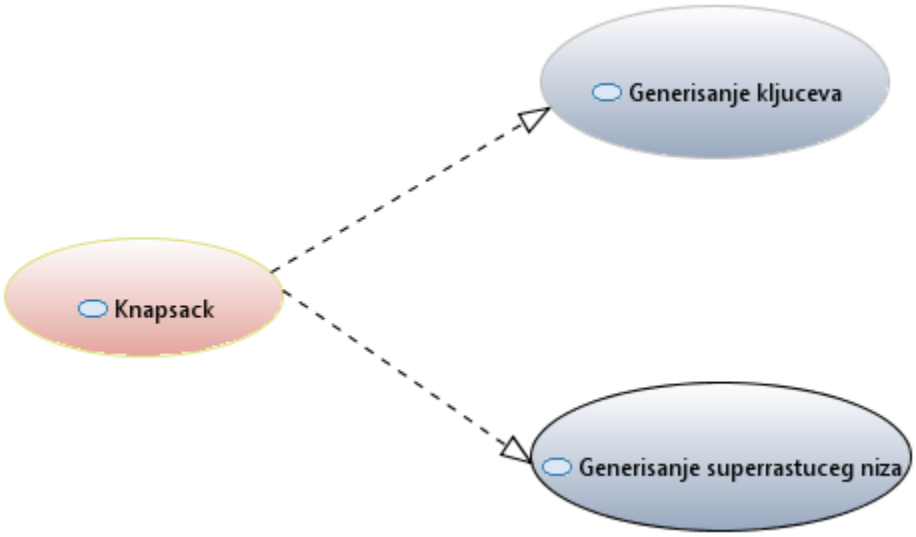
Detaljni UML dijagram za slučaj korišćenja *unos kljuca i fajla za kodiranje i fajla u kome ce se snimati kodirani podaci kod RC6 algoritma* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *unos kljuca i vektora IV kod TEA/XTEA algoritma* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj koriscenja *generisanje kljuceva i superrastuceg niza kod Knapsack algoritma* je prikazan na sledećoj slici:



6.Pogled na logičku arhitekturu sistema

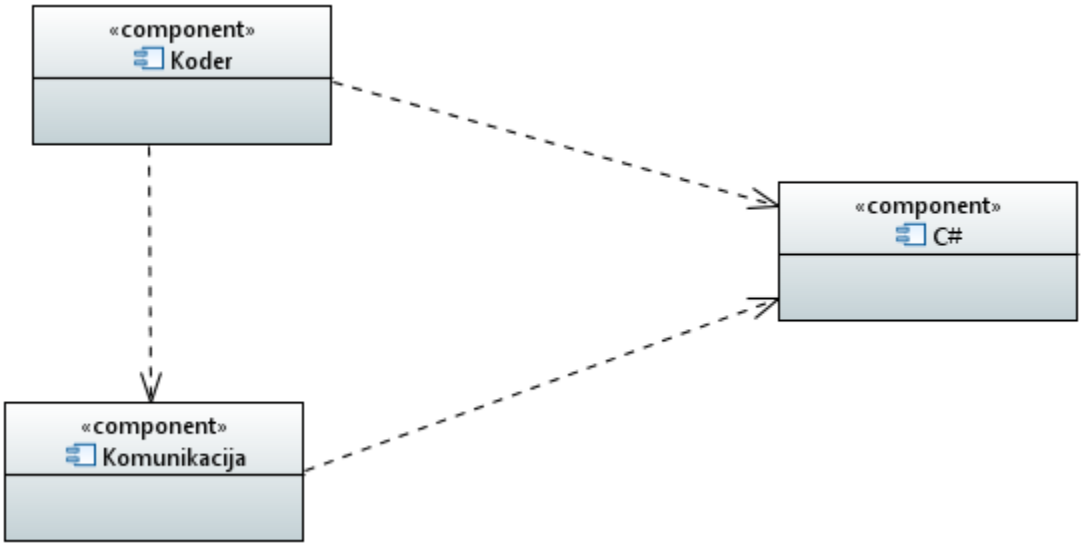
U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva

korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

Logički pogled na aplikaciju kriptovanja i dekriptovanja podataka obuhvata 2 glavne klase: *Koder i Komunikaciju*.

Klasa Koder sadrži glavni prozor i komponente za kriptovanje i dekriptovanje podataka biranjem zeljenog algoritma, odnosno grafički interfejs preko kog korisnici sistema komuniciraju sa aplikacijom.

Klasa *Komunikacija* zadužena je za realizaciju funkcionalnosti koje se tice komunikacije. Komunikaciju realizovati preko soketa.



6.1.1 Koder

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržane sve klase koje predstavljaju poglede na podatke.

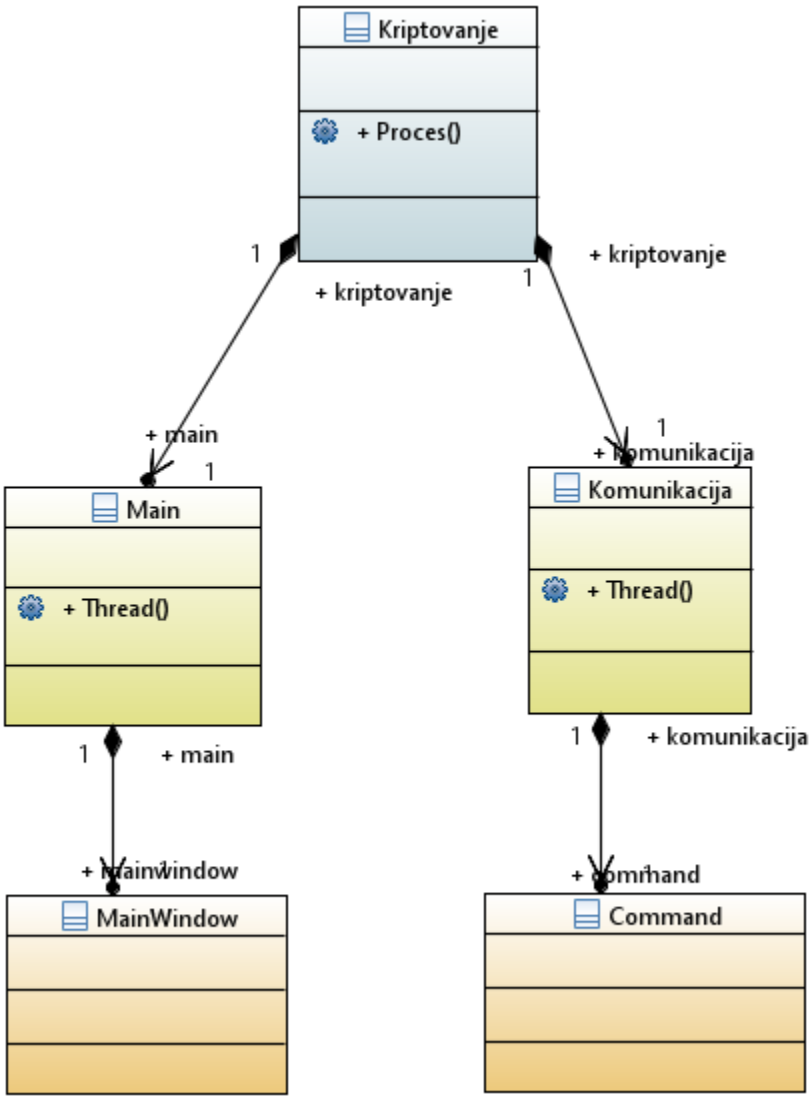
6.1.2 Komunikacija

Ovaj sloj zadužen je za realizaciju funkcionalnosti koje se tice komunikacije. Komunikaciju realizovati preko soketa.

6.1.3 C#, Windows form aplikacija

7.Pogled na procese

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.



7.1.1 MMS

Aplikacija koja poseduje glavnu nit *Main* zaduženu za *update*-ovanje GUI-a implementiranog klasom *MainWindow* i pomoćnu nit *Komunikacija* koja se koristi za operacije koje se izvršavaju duže vreme kako ne bi došlo do blokiranja grafičkog interfejsa.

8.Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja aplikacije za kriptovanje i dekriptovanje podataka.

8.1 Korisnički računar

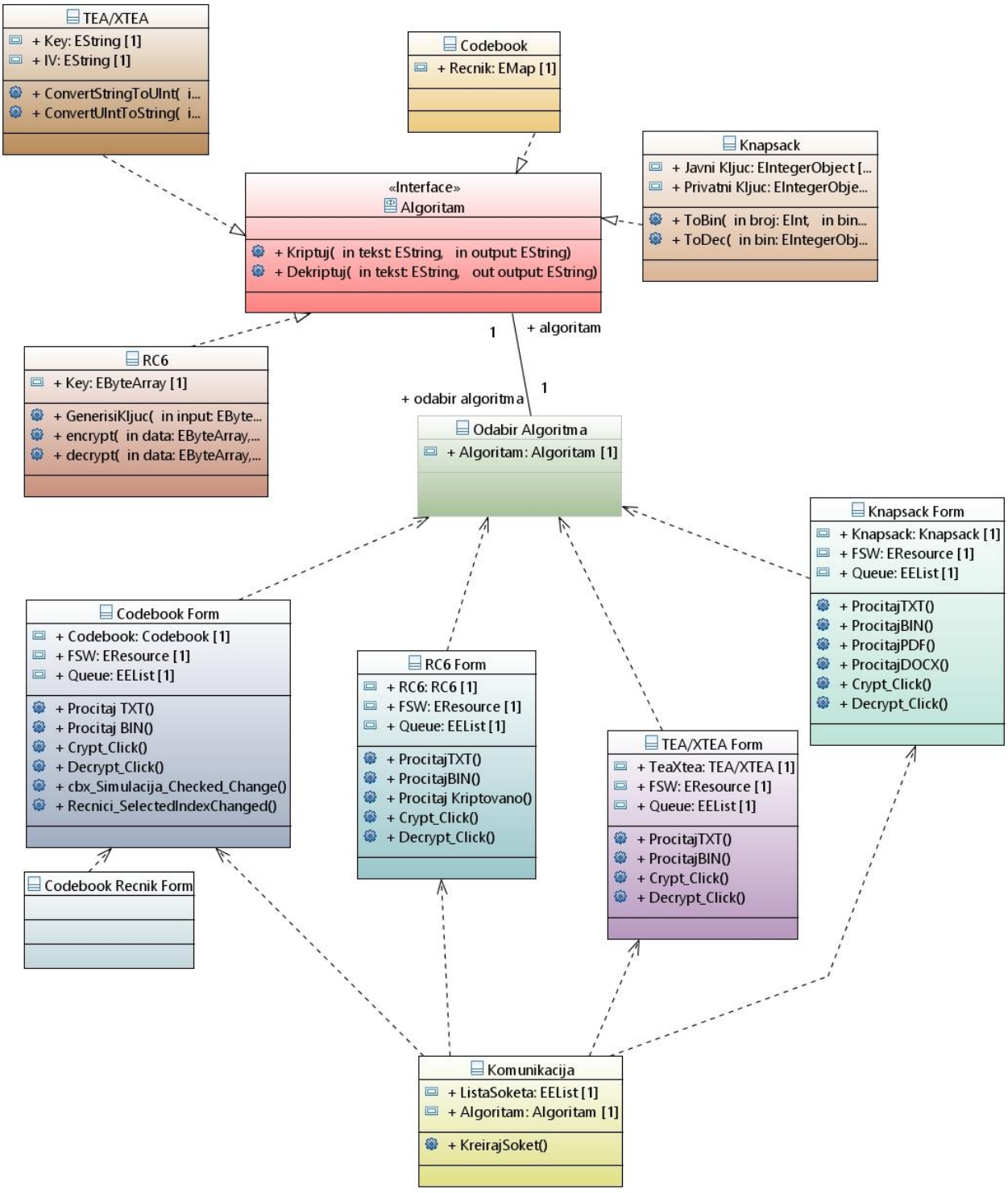
Aplikacija za kriptovanje i dekriptovanje podataka se izvršava na korisničkom računaru sa svim potrebnim bibliotekama.

9.Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju aplikacije MMS ovaj odeljak sadrži model domena i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

9.1 Model domena

Model domena za koji se aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, kao i veze koje se mogu identifikovati između njih.


TEA/XTEA
+ Key: EString [1]
+ IV: EString [1]
+ ConvertStringToUInt(i...
+ ConvertUIntToString(i...
Codebook
+ Recnik: EMap [1]
Knapsack
+ Javni Kljuc: EIntegerObject [...
+ Privatni Kljuc: EIntegerObje...
+ ToBin(in broj: EInt, in bin...
+ ToDec(in bin: EIntegerObj...
«Interface»
Algoritam
+ Kriptuj(in tekst: EString, in output: EString)
+ Dekriptuj(in tekst: EString, out output: EString)
1
+ algoritam
1
+ odabir algoritma
RC6
+ Key: EByteArray [1]
+ GenerisiKljuc(in input: EByte...
+ encrypt(in data: EByteArray,...
+ decrypt(in data: EByteArray,...
Odabir Algoritma
+ Algoritam: Algoritam [1]
Knapsack Form
+ Knapsack: Knapsack [1]
+ FSW: EResource [1]
+ Queue: EEList [1]
+ ProcitajTXT()
+ ProcitajBIN()
+ ProcitajPDF()
+ ProcitajDOCX()
+ Crypt_Click()
+ Decrypt_Click()
Codebook Form
+ Codebook: Codebook [1]
+ FSW: EResource [1]
+ Queue: EEList [1]
+ Procitaj TXT()
+ Procitaj BIN()
+ Crypt_Click()
+ Decrypt_Click()
+ cbx_Simulacija_Checked_Change()
+ Recnici_SelectedIndexChanged()
RC6 Form
+ RC6: RC6 [1]
+ FSW: EResource [1]
+ Queue: EEList [1]
+ ProcitajTXT()
+ ProcitajBIN()
+ Procitaj Kriptovano()
+ Crypt_Click()
+ Decrypt_Click()
TEA/XTEA Form
+ TeaXtea: TEA/XTEA [1]
+ FSW: EResource [1]
+ Queue: EEList [1]
+ ProcitajTXT()
+ ProcitajBIN()
+ Crypt_Click()
+ Decrypt_Click()
Codebook Recnik Form
Komunikacija
+ ListaSoketa: EEList [1]
+ Algoritam: Algoritam [1]
+ KreirajSoket()

9.2 Sekvencijalni dijagram