Käyttöliittymäohjelmointi

**Randomizenator**

Oulun yliopisto

Tietojenkäsittelytieteiden laitos

Harjoitustyöselostus

Jussi Jokio, Niko Kupari

Pvm

Tiivistelmä

Randomizenator on salasanan generointi ohjelma, jolla voit luoda turvallisen salasanan annetuilla parametreilla. Ohjelma sopii yksittäisille käyttäjille tai ison organisaation käyttöön ja ohjelma on tehokas kaiken tasoisille käyttäjille.

Randomizenator käyttää Javan omaa random-luokkaa arpoessaan salasanaa merkki kerrallaan. Ohjelma ei kuitenkaan käytä suoraan random-olion tuottamia merkkejä, joten turvallisuusriskit ovat pienemmät. Ohjelma tuottaa yksittäisiä salasanoja hyvin nopeasti ja eri parametrien muuttaminen on helppoa. Ohjelmalla saadaan arvottua myös esimerkiksi pelkkiä numeroita ja se tuottaa erimittaisia salasanoja aina 20 merkkiin asti. Ohjelmalla voit halutessasi tallentaa käyttäjänimi-salasana pareja tekstitiedostoon (.txt).

Randomizenator on mahtava apu niin työssä, koulussa kuin kotonakin.

Sisällys

[Tiivistelmä 2](#_Toc469438153)

[1. Vaatimusmäärittely 4](#_Toc469438154)

[1.1 Toiminnallisuuden määrittely 4](#_Toc469438155)

[1.2 Käyttäjäryhmien identifiointi 4](#_Toc469438156)

[1.2.1 Esimerkkikäyttäjän #1 kuvaus 4](#_Toc469438157)

[1.2.2 Esimerkkikäyttäjän #2 kuvaus 4](#_Toc469438158)

[1.2.3 Esimerkkikäyttäjän #3 kuvaus 5](#_Toc469438159)

[1.3 Käytön kontekstin määrittely 5](#_Toc469438160)

[1.3.1 Fyysinen konteksti 5](#_Toc469438161)

[1.3.2 Sosiaalinen konteksti 5](#_Toc469438162)

[1.3.3 Organisatorinen konteksti 5](#_Toc469438163)

[1.3.4 Toiminnallinen konteksti 5](#_Toc469438164)

[1.3.5 Tekninen konteksti 6](#_Toc469438165)

[2. Käyttöliittymäkonseptit 7](#_Toc469438166)

[2.1.1 Konsepti #1 7](#_Toc469438167)

[2.1.2 Konsepti #2A & #2B 8](#_Toc469438168)

[2.1.3 Konsepti #3 9](#_Toc469438169)

[3. Käyttöliittymän prototypointi 11](#_Toc469438170)

[4. Käyttöliittymän evaluointi 14](#_Toc469438171)

[4.1 Heuristinen evaluointi 14](#_Toc469438172)

[4.2 Skenaariopohjainen evaluointi 14](#_Toc469438173)

[4.3 Käyttäjätestaus 15](#_Toc469438174)

[4.4 Vertaisarvioinnit 16](#_Toc469438175)

[4.5 Evaluointien löydökset ja vaikutukset 17](#_Toc469438176)

[5. Viimeistelty suunnitelma käyttöliittymästä 18](#_Toc469438177)

[6. Käyttöohje 22](#_Toc469438178)

[7. Poikkeamat suunnitelmasta 23](#_Toc469438179)

[8. Liitteet 24](#_Toc469438180)

[9. Harjoitustyön tekijöiden palautustiedot 26](#_Toc469438181)

# Vaatimusmäärittely

Randomizenator on suunnattu helpottamaan salasanojen tuottamista. Tavoitteena on rakentaa salasana-konstruktori joka auttaa käyttäjää suojaamaan tiedostoja tai tilejä.

## Toiminnallisuuden määrittely

Randomizenator generoi käyttäjälle salasanan käyttäjän haluamilla rajoituksilla, joihin lukeutuu esim. isot tai pienet kirjaimet, numerot ja merkkien määrä. Ohjelmaa voidaan myös hyödyntää muihin satunnaisia merkkejä vaativiin tilanteisiin.

Ohjelman toimintaan vaaditaan luotettava satunnaisgeneraattori, joka pystyy luomaan erilaisia merkkijonoja annetuilla parametreilla. Käyttäjän pitää kyetä vaikuttamaan salasanan parametreihin haluamallaan tavalla. Suurin sallittu merkkimäärä on rajoitettu 20 merkkiin.

## Käyttäjäryhmien identifiointi

Randomizenator-ohjelmalla on erittäin laaja käyttäjäryhmä ja -ikähaarukka. Ohjelman käyttö ei vaadi aiempaa tietotaitoa tai erityistä rajapintaa.

Randomizenator –ohjelmaa voi käyttää peruskäyttäjän suojaukseen, mutta myös vaativimpiin merkkiyhdistelmiin. Erilaiset käyttäjät voivat toteuttaa salasanansa eri parametrien mukaan.

Tärkein ohjelman käyttäjäryhmä on sellaiset käyttäjät, jotka tarvitsevat korkean suojauksen salasanan useita kertoja vuodessa. Myös henkilöt jotka eivät osaa muodostaa turvallisia salasanoja itse, käyttävät Randimizenatoria.

Esimerkkikäyttäjien kuvauksia (mitä enemmän asiaa/käyttäjä, sitä parempi)

### Esimerkkikäyttäjän #1 kuvaus

Teemu Teekkari on tietotekniikan opiskelija yliopistossa ja on tietoinen cyber-turvallisuuden vaatimuksista. Hän tarvitsee turvallisen salasanan lähetettävään tiedostoon, joka sisältää arkaluontoista dataa. Teemu tarvitsee nopeasti yhden kertakäyttöisen salasanan tiedostoa varten. Koska tiedosto on erittäin salainen, Teemu tarvitsee pitkän salasanan, jossa on isoja kirjaimia, numeroita ja vähintään 10 merkkiä. Teemu tietää, että ilkeämielinen taho on kuunnellut hänen puheluitaan, joten hän ei voi käyttää hänen lempisalasanaa ”TeemunSalis”.

### Esimerkkikäyttäjän #2 kuvaus

Marja-liisa Kukkanen on yli 50-vuotias rouva Petäjävedeltä, Hän haluaa liittyä ystävänsä innoittamana keskustelufoorumille ”kukkikset.fi”. Marja-liisan täytyy tehdä sivustolle käyttäjäprofiili ja hän on kuullut siskontytöltään Teemulta kuinka vaarallista internetissä voi olla. Marja-liisa ei osaa valita itselleen turvallista salasanaa joka sisältää numeroita.

### Esimerkkikäyttäjän #3 kuvaus

Matti Tepponen on ihan tavallinen henkilö, ihan tavallisessa työssä, ihan tavallisena maanantaina. Matti kirjautuu firmansa koneelle ja huomaa, että hänen on vaihdettava salasana. Matin firmassa on käytäntönä, että salasana tulee vaihtaa kuukauden välein ja matti on käynyt läpi jo kaikki perheen lemmikit. Firman käytäntönä salasanan tulee olla vähintään 8 merkkiä, sisältää isoja ja pieniä kirjaimia, numeroita ja erityismerkkejä eikä mikään vanhoista salasanoista käy.

* + 1. Esimerkkikäyttäjän #4 kuvaus

Seppo Turvallinen on tietoturvallisuuden testaaja, joka testaa profiilien turvallisuutta. Hän luo omalle serverille käyttäjiä ja testaa niiden turvallisuutta, jotta hän saa tietoon millaiset salasana vaatimukset se tarvitsee. Seppo luo päivittäin kymmeniä profiileja erilaisilla salasanoilla testejä varten.

## Käytön kontekstin määrittely

Randomizenator on tarkoitettu yleisimmille tietokoneen käyttöjärjestelmille kuten Windows ja Linux. Ohjelmaa voidaan käyttää niin yksityishenkilöiden salasanojen muodostamiseen, kuin eri organisaatioiden käyttöön. Ohjelmalla on käyttöä jokaisessa tilanteessa, missä tarvitaan salasanaa tai jopa ainustaan satunnaismerkkejä.

### Fyysinen konteksti

Ohjelmaa käytetään tietokoneen käyttöjärjestelmän päällä, joka tukee Java. Tietokoneen ei tarvitse olla yhteydessä nettiin sovellusta käytettäessä, joten sitä voidaan käyttää melkein missä vain. Fyysinen ympäristön melu, hämäryys tai valoisuus eivät vaikuta ohjelman käyttöön. Salasanaa generoidessa muita henkilöitä ei saisi olla takanasi, sillä he voivat nähdä salasanan.

### Sosiaalinen konteksti

Ohjelman käyttöön vaaditaan vain yksi käyttäjä, sillä ohjelmalla generoidaan ainoastaan yksi salasana kerrallaan. Poikkeuksina siihen on, jos ryhmälle tai joukolle yhteinen käyttäjä, jolle generoidaan uusi salasana.

### Organisatorinen konteksti

Ohjelmaa voidaan käyttää yksittäiseen tiliin, mutta skaalaaminen on helppoa. Sovellusta voi käyttää niin yksittäiset henkilöt kuin myös yritykset.

### Toiminnallinen konteksti

Käyttäjä luo käyttäjä tililleen haluamillaan parametreilla salasanan, joten salasanojen tuottaminen täytyy olla sujuvaa ja nopeaa. Eri parametrien muuttaminen täytyy onnistua helposti. Käyttäjä voi myös tallentaa salasanansa halutessaan. Ohjelmaa tarvitsee käyttää vain luodessaan uuden salasanan.

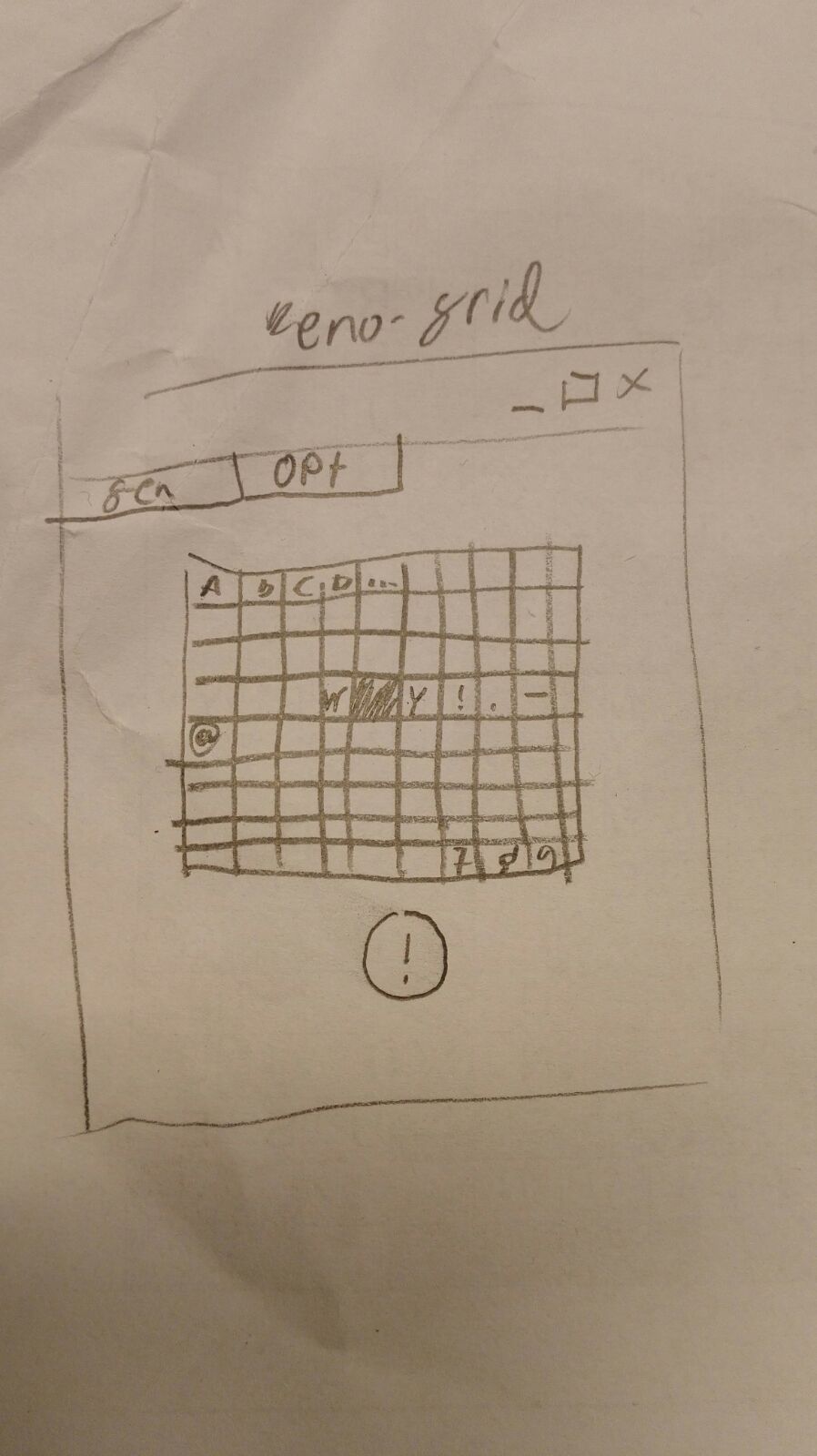
### Tekninen konteksti

Tarkoituksena on luoda erilaisia ja eri parametreilla luotuja salasanoja tarpeen mukaan. Vaaditaan vaikeasti purettava salaus ja mahdollisimman vähän korrelointia salasanojen välillä. Ohjelma ei vaadi koneen resursseilta paljoa, joten sitä voidaan käyttää vanhoillakin koneilla, mutta koneen pitää tukea Javaa. Salasanan tallentaminen vaatii tallennusoikeuden koneelta ja tallennustilaa noin 1MB.

# Käyttöliittymäkonseptit

Prototypoinnin perusteella valittiin #4, jota lähdettiin muokkaamaan valmiimpaan suuntaan (ks. prototyyppi). Syy miksi valittiin #4, oli konseptin selkeys verrattuna muihin konsepteihin. #4 oli myös visuaalisesti näyttävämpi kuin muut konseptit. Iso osa päätöksentekoa oli myös Help-painikkeen lisääminen käyttöliittymään, koska muista se puuttui.

### Konsepti #1

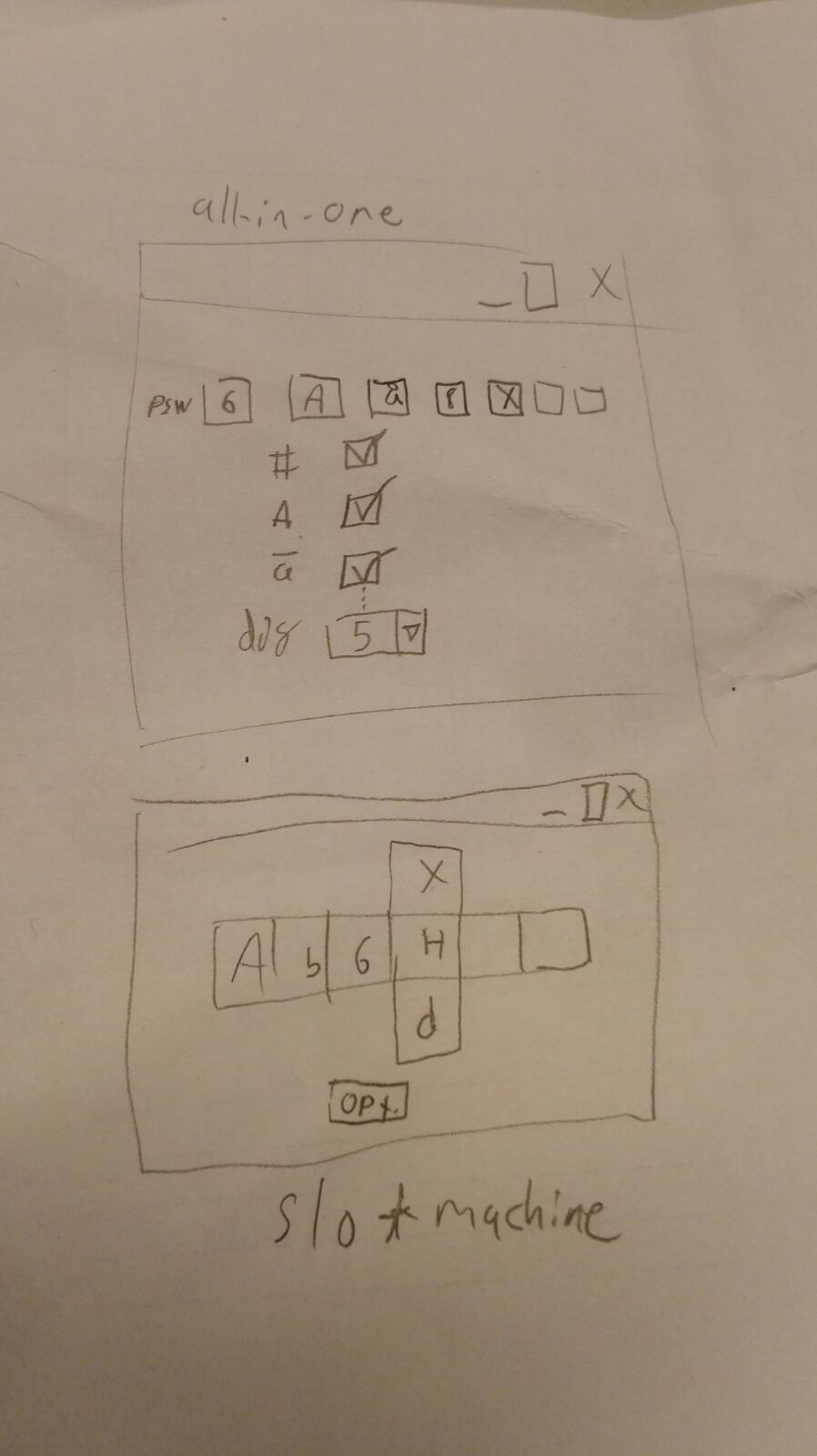


Ajatuksena antaa salasana merkki kerrallaan kuten Kenossa. ! -näppäin arpoo seuraavan merkin.

Ei kauhean selkeä ja hidas suurilla merkkimäärillä, esim. 15 merkin salasana vaatii 15 klikkausta. Edelliset merkit eivät ole järjestyksessä vaan pelkästään tummennettuina, joten vain yksi merkki voi ilmestyä vain kerran. Voidaan arpoa salasana merkki kerrallaan ja käyttäjä päättää haluaako hän käyttää arvottua merkkiä.

Tämä konsepti sopii erityisesti tyypin 2 esimerkkikäyttäjille kuten Marja-liisa Kukkaselle, koska käyttöliittymä on hyvin yhdennäköinen esimerkiksi kenon arvontaruudukon kanssa. Toisaalta konsepti on aivan liian hidas tyypin 4 käyttäjille kuten Seppo Turvallinen.

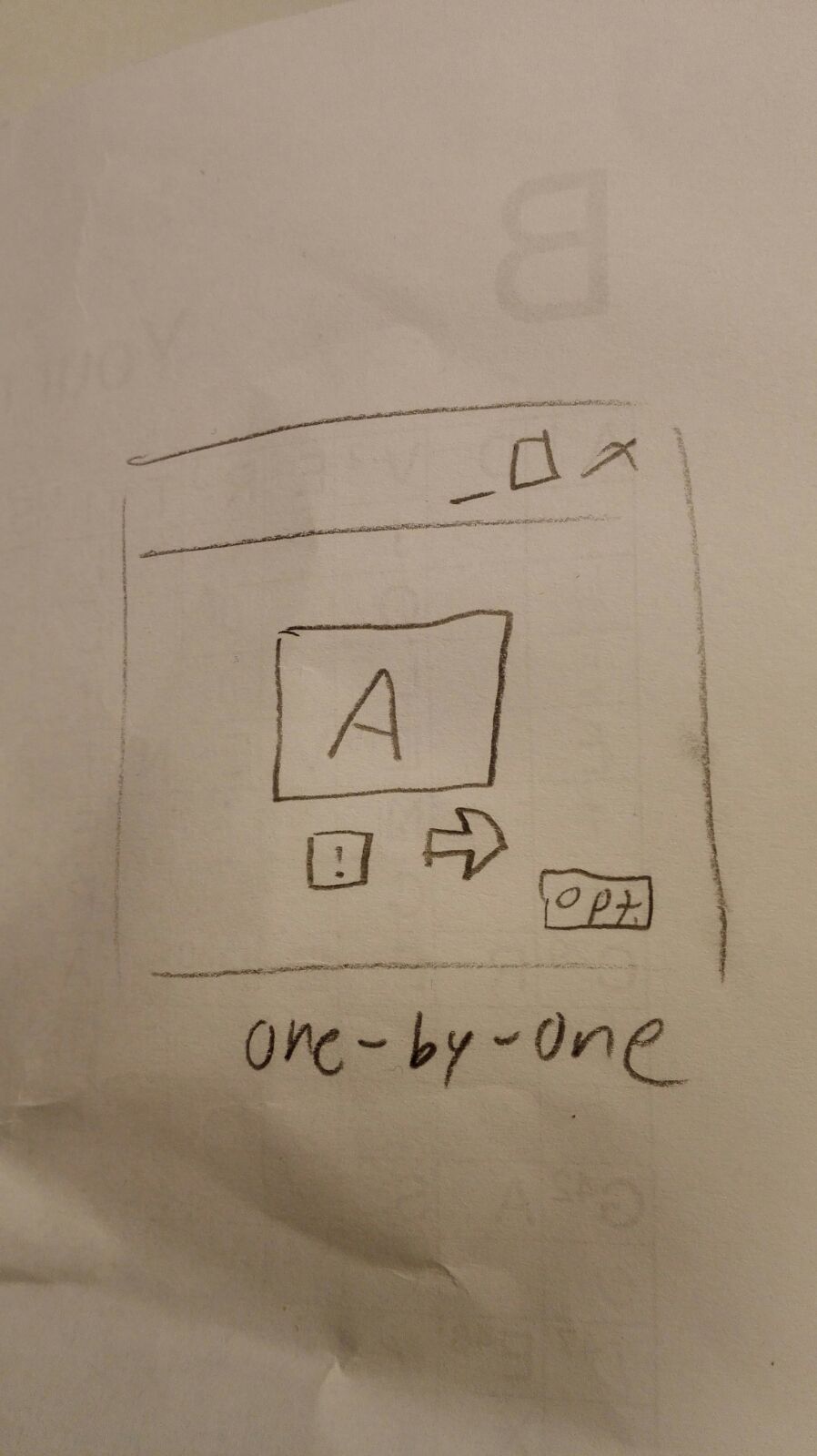
### Konsepti #2A & #2B



A: Kaikki toiminnot samassa ikkunassa, luo ahtauden tunnetta. Lisäksi salasanana generointi painike puuttuu. Tämä konsepti hylättiin välittömästi ahtauden ja epäjärjestyksen vuoksi. Lyhyille salasanoille jää ylimääräisiä ruutuja ja salasanan kopioiminen leikepöydälle ei onnistu kivuttomasti.

B: Prototyyppi ei ole kovin selkeä käyttäjälle ja vaatii perehtymistä ennen käyttöä. Lisäksi salasanana generointi painike puuttuu. Kuten konsepti #1 tämä sopii myös tyypin 2 käyttäjille, koska sillä on vastaavuus uhkapelimaailmassa, mutta tyypin 4 käyttäjille tämä konsepti on aivan liian hidas.

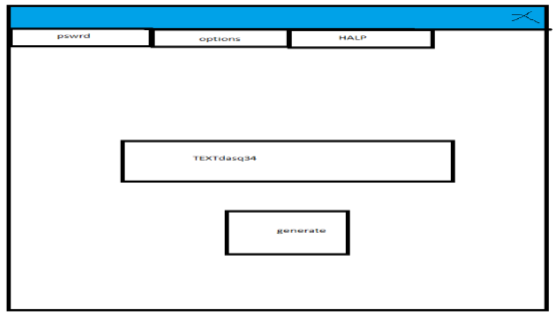
### Konsepti #3

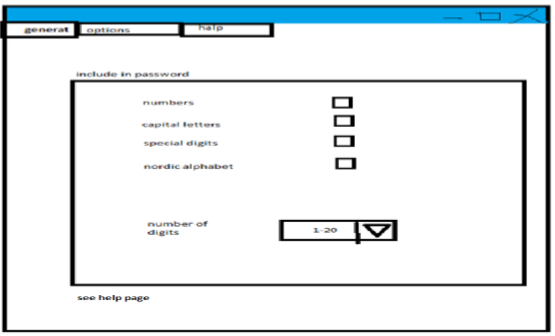


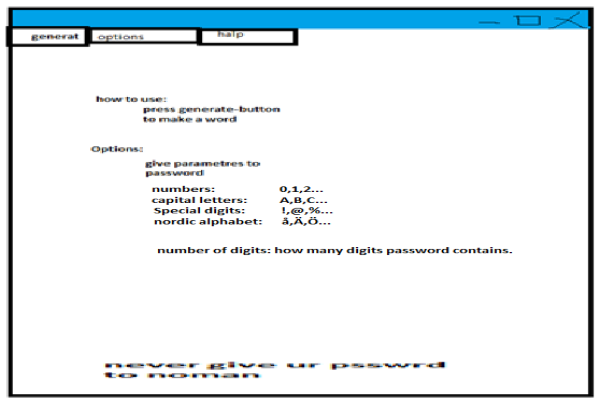
Käyttäjä arpoo salasanan merkki kerrallaan.

Ei kauhean selkeä ja hidas suurilla merkkimäärillä, esim. 15 merkin salasana vaatii 15 klikkausta. Edelliset merkit eivät ole näkyvissä. Voidaan arpoa salasana merkki kerrallaan ja käyttäjä päättää haluaako hän käyttää arvottua merkkiä. Yksi merkki voi esiintyä useamman kerran. Tämä konsepti hylättiin juuri sen verrattain hitaan toiminnan takia.

* + 1. Konsepti #4





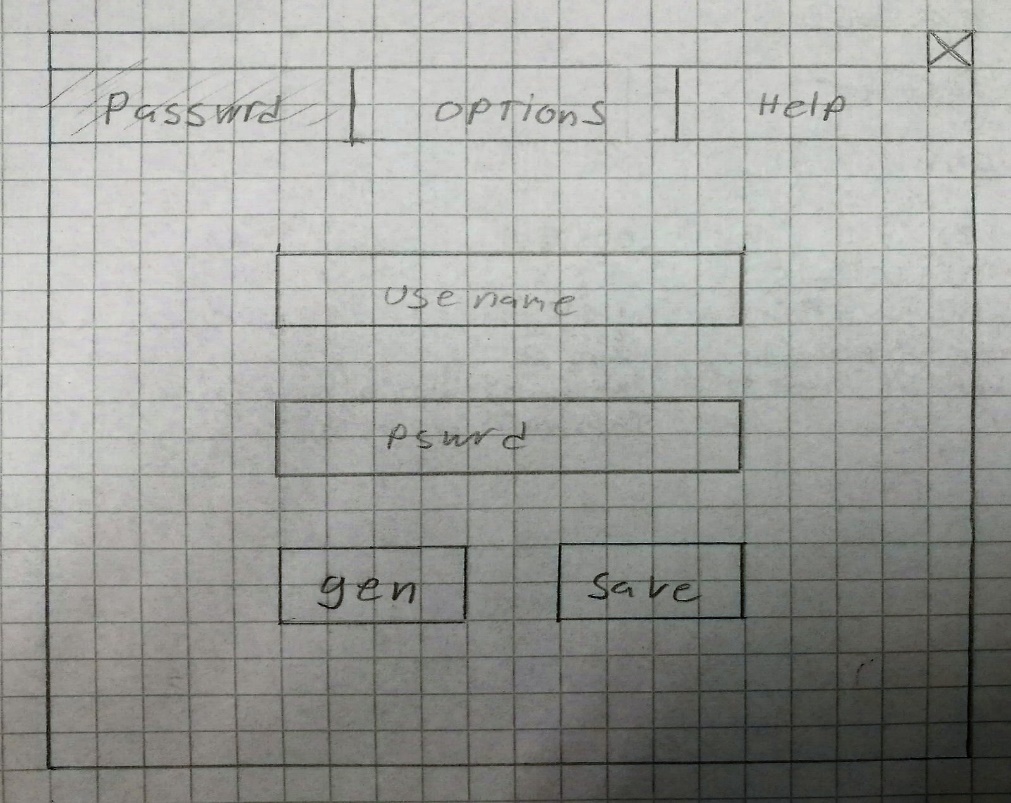


Koska ohjelmassa on erilaisia toimintoja verrattain vähän, voi prototyyppi #4 näyttää koko ruudulla hieman avonaiselta. Options-välilehteä piti selittää testikäyttäjälle, joten lisättiin selvennykset help-välilehteen. Generoitu salasana ei ole mikään oikea sana (”sala5@Na”), eikä siihen ole valikkoa.

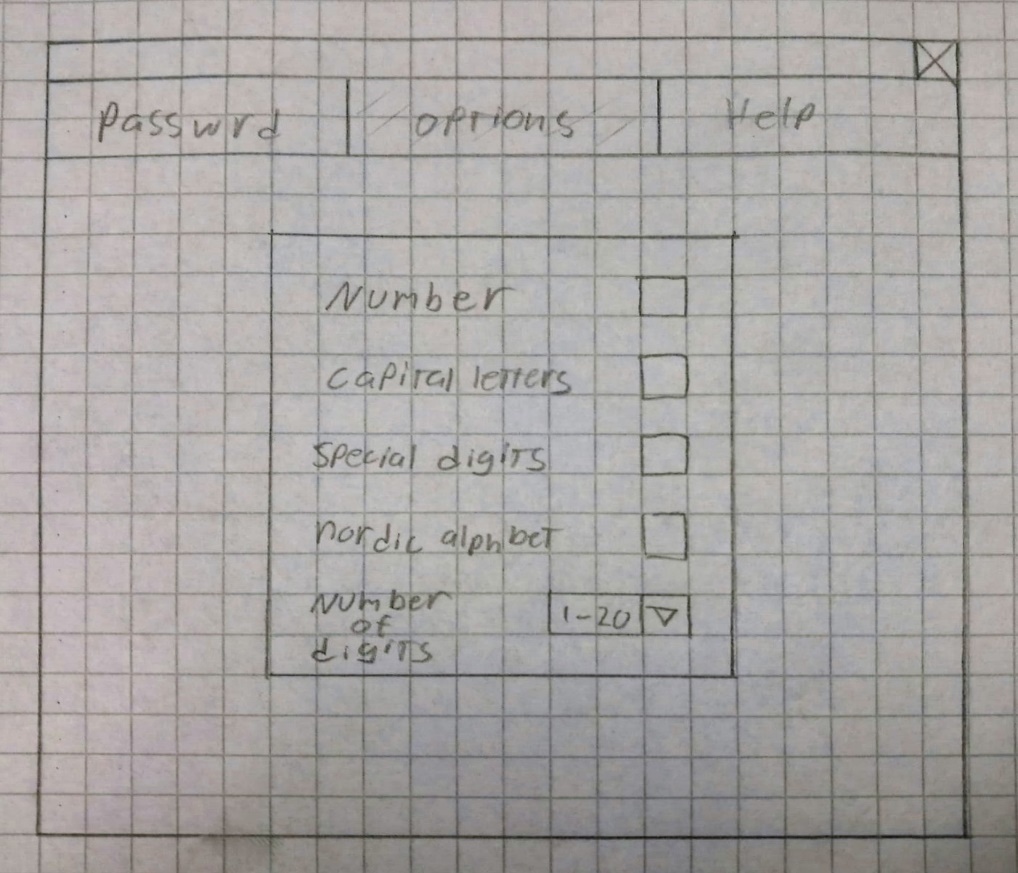
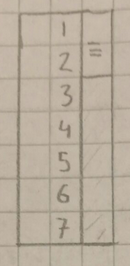
Tämä konsepti löysi kultaisen keskitien käytön nopeuden ja selkeyden välillä ja se sopii kaikille esimerkkikäyttäjille. Käyttö on sujuvaa ja asetuksien muuttaminen on selkeää ja tarpeen vaatiessa myös ohjeet ovat näkyvillä helposti.

Tämä konsepti valittiin prototypointiin.

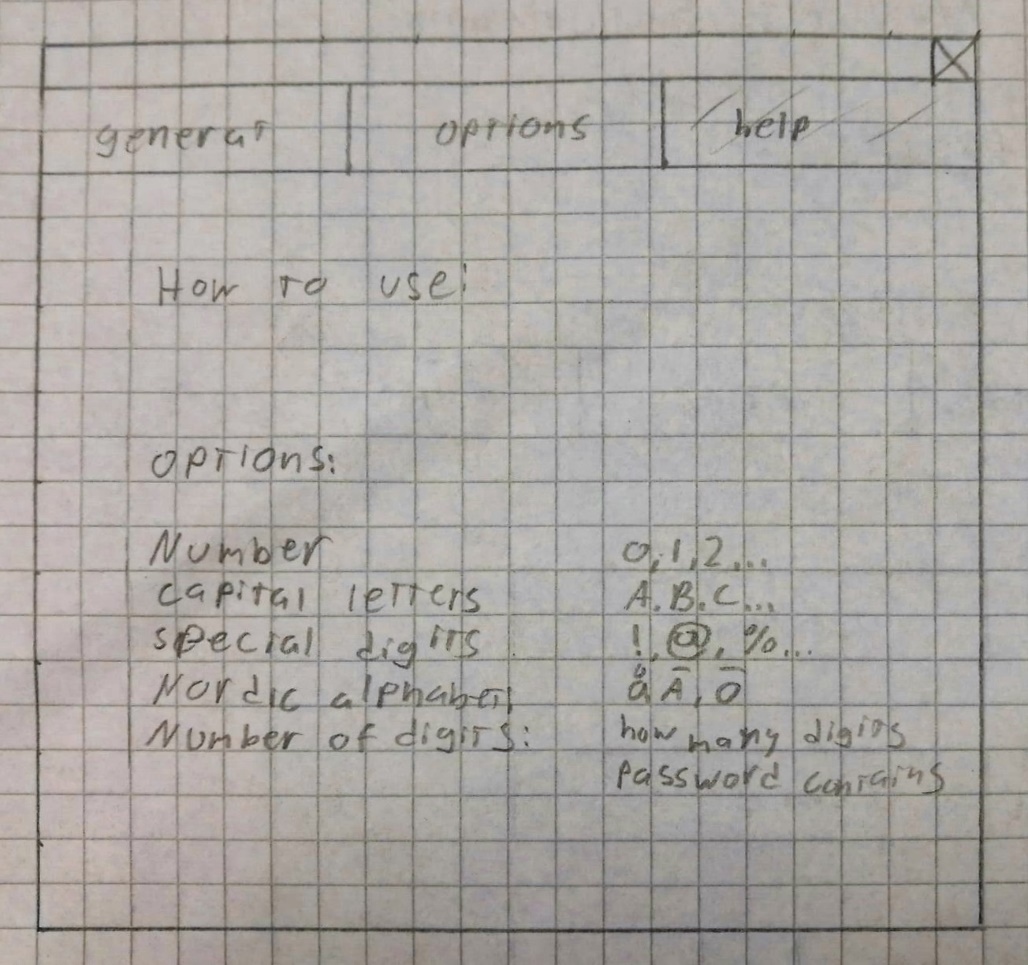
# Käyttöliittymän prototypointi



**Aloitusruutu:** Käyttäjän ensimmäinen näkymä ja salasanan generointialue, mahdollisuutena myös käyttäjänimi-salasana parin tallentaminen. Mahdollista tallentaa myös ilman käyttäjänimeä.

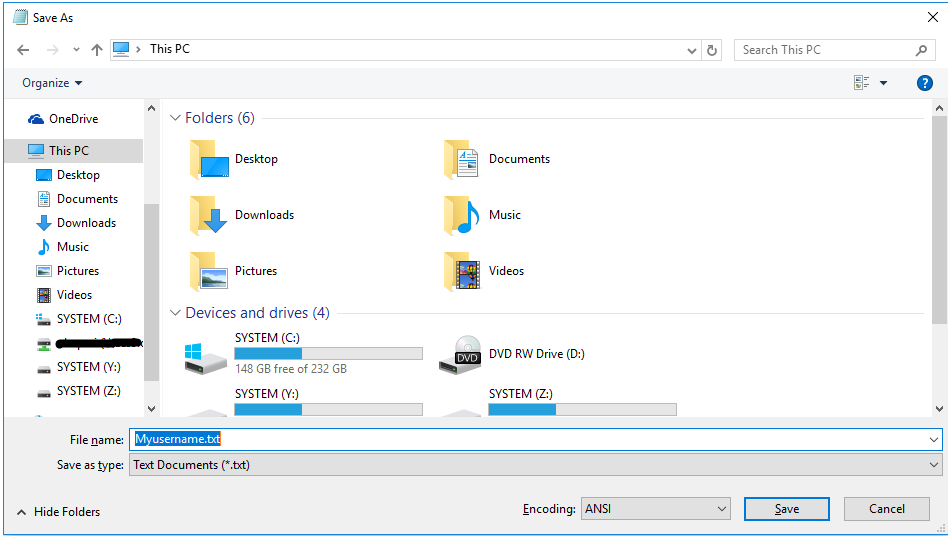
 

**Optionsruutu**: Tässä valitaan salasanan parametrit. millaisia merkkejä salasana sisältää ja kuinka monta merkkiä. Merkkien määrä valikossa käytetään liukuvalikkoa. Oletuksena käytetään vain pienet kirjaimet ja 8 merkkiä.

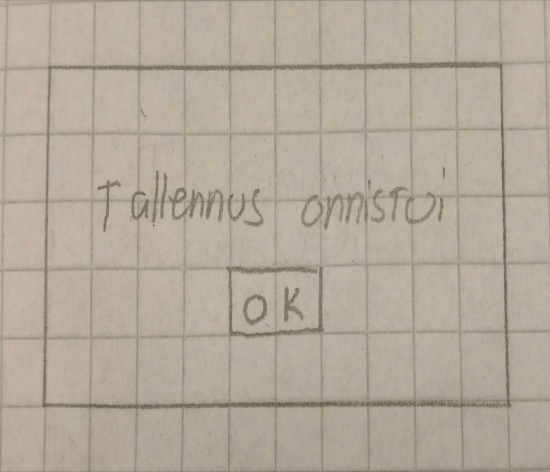
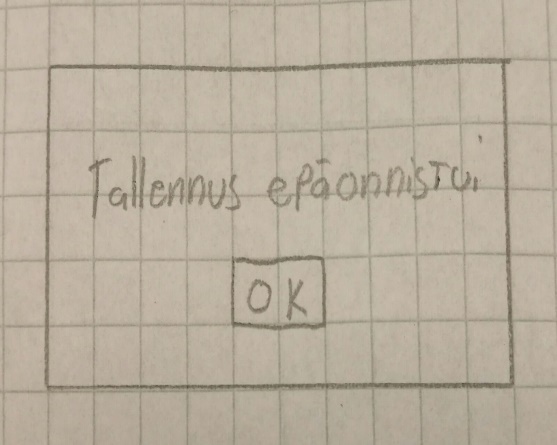


**Helpruutu:** Annetaan käyttäjälle esimerkkejä erilaisista merkeistä ja kerrotaan lyhyesti ohjelman toiminta.

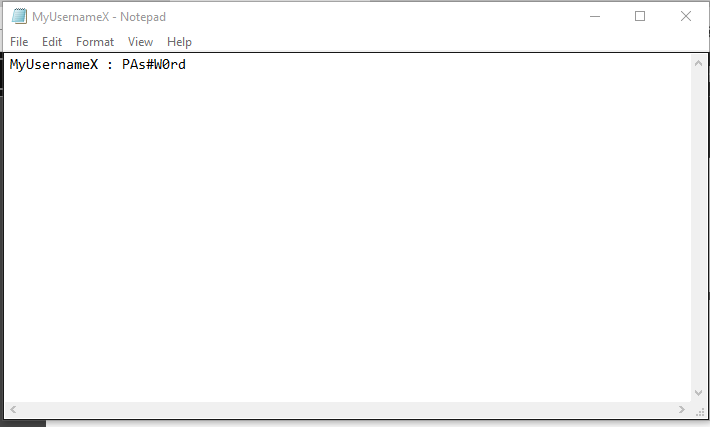
**”** How to use: Generate your password with “Gen”-button. Use “options” to give parametrs to password. Username is optional”



**Save mode 1:** Käytetään Windowsin omaa tallennus ikkunaa. Tämä on oletus tallennustila.

**Save mode 2:** Kerrotaan käyttäjälle onnistuiko salasanan tallennus vai ei. Tiedosto tallennetaan samaan kansioon kuin ohjelma. Tallennusta ei voi suorittaa, jos käyttäjä ei ole generoinut salasanaa, käyttäjällä on vain luku oikeudet koneelle tai saman niminen tiedosto on jo olemassa.



**Tallennettu tiedosto:** Tallennetaan käyttäjänimi-salasanapari erilliseen tekstitiedostoon jonka oletusnimenä käytetään käyttäjänimeä. Jos ei ole annettu käyttäjänimeä käytetään nimeä ’username.txt’. Tallennetaan muodossa ’käyttäjänimi : salasana’.

# Käyttöliittymän evaluointi

Evaluointi tehtiin Nielsenin heuristisilla säännöillä. Löydetyt ongelmat on korjattu prototyyppiin ja kirjattu alle.

Prototyyppi vastasi alkuperäisiin vaatimusmäärittelyihin hyvin ja joissain tilanteissa jopa ylitti ne. Prototyyppiin ei tarvitse tehdä toiminnallisia muutoksia, vaan ainoastaan visuaalisia parannuksia tai järjestelyjä.

## Heuristinen evaluointi

Käytettiin Jakob Nielsenin kymmentä sääntöä heuristisessa arvioinnissa. (<https://www.nngroup.com/articles/ten-usability-heuristics/>) Löydetyt ongelma kohdat, kuten virhe ilmoitukset, on kirjattu tähän.

* + Aloitus ruutu: Aloitus ruutu tarvitsee selkeyttä, jotta käyttäjä ymmärtää mihin ruutuun voidaan kirjoittaa. Tällä hetkellä ”Username” ja ”Password” valikot näyttävät samalta, vaikka ne toimivat eri tavalla. Tarkoituksena on laittaa ”Username”-laatikkoon watermark kuvaamaan käyttäjänimen kirjoittamista ”Insert username here”. Aloitus ruudun alalaitaan lisätään teksti ”Never give your password to anyone!”.
  + Options ruutu: Asetukset ovat selkeästi sijoiteltu, mutta hieman puutteelliset. Asetusvalikosta puuttuu myös ”OK”-painike parametrien käyttöönottoa. varten Asetuksista puuttuu pienet kirjaimet ja vakioasetuksien näkyminen. Oletus asetukset pitää näkyä valikossa, kun ohjelma käynnistetään (vain pienet kirjaimet, 8 merkkiä). Prototyypistä ei selviä, että käytetään merkkien määrän kuvaamiseen liukuvalikkoa (korjattu prototyyppiin). Jos käyttäjä on ilkeämielinen ja poistaa kaikki parametrit generaattorista, Täytyy ohjelman antaa virheilmoitus.
  + Help ruutu: Informaatio on jäsennetty selkeäksi ja käyttäjää ei kiusata liialla tekstillä. ”How to use:” otsikkoon lisätään tieto, ettei käyttäjänimi ole pakollinen tieto.
  + Save Mode I: arvioinnin jälkeen päädyttiin käyttämään kyseistä tallennus muotoa.
  + Ilmoitukset: Virheilmoitus tapahtuu, kun jokin seuraavista toteutuu; salasanan parametrit puuttuvat, tallennus ei onnistu, poistutaan ohjelmasta. Virheilmoitukset sisältävät syyn virheilmoituksen ilmestymiseen.

## Skenaariopohjainen evaluointi

Käyttäjän tehtävänä on:

* + Tuottaa neljä (4) yhdeksänmerkkistä salasanaa, joissa kaikissa on käytetty isoja ja pieniä kirjaimia.
    - * Käyttäjällä meni tähän tehtävään 9 klikkausta, joka on erittäin nopeaa.
  + Luoda salasana, joka koostuu 5 merkistä, joissa on mukana myös numeroita, pohjoismainen aakkosto ja isot kirjaimia.
    - * Käyttäjällä meni tähän 8 klikkausta.
  + Tallentaa tuotettu käyttäjänimi-salasana pari, jossa on vain pieniä kirjaimia ja kahdeksan merkkiä.
    - * Käyttäjällä meni tähän neljä klikkausta, mutta tähän ei otettu huomioon käyttäjänimen pituutta, eikä muutettu oletustallennus polkua. Käyttäjän ei tarvinnut vaihtaa välilehteä.

## Käyttäjätestaus

Suoritimme kahta erilaista käyttäjätestausta. Ensimmäinen testaus oli epämuodollinen testaus, jonka suoritti eräs vertaisryhmän edustaja, ei kuitenkaan kehittäjä. Testaus suoritettiin projektin alkuvaiheessa, kuitenkin siten, että lähes kaikki toiminnallisuus oli jo valmista. Ensimmäinen testaus keskittyi lähinnä toiminnallisuuteen ja siitä ei ole yhtä kattavaa dokumentaatiota kuin toisesta testauksesta. Testauksessa löydettyjä ongelmia olivat esim. ”Nordic alphabets” luokan muokkaaminen yksikäsitteiseksi, siten, että se sisältää isot ja pienet kirjaimet eikä kyseistä ominaisuutta voi muokata. Muutos oli helpompi toteuttaa ja vähensi käyttäjien virheilmoituksia huomattavasti.

Toinen käyttäjätestaus suoritettiin projektin loppuvaiheessa ja oli paljon formaalimpi kuin ensimmäinen testaus. Testauksen tulokset ovat liitteenä 1 ja 2. Testaajille annettiin testauslomake ja Randomizenator ohjelma käytettäväksi. Liite 1 on esimerkkikäyttäjän 3 (Matti Tepponen) tyypin testaaja ja liite 2 taas edustaa käyttäjätyypin 4 (Seppo Turvallinen) testaajan arviota.

Testauksista selvinneet ongelmakohdat olivat muun muassa: käyttäjänimi laatikon muokkaaminen informatiivisemmaksi ja tiedoston tallennukseen lisättiin .txt pääte oletuksena. Myös ”Help”-välilehteä muokattiin käyttäjäystävällisemmäksi.

## Vertaisarvioinnit

*”Mielenkiintoinen ohjelma!*

*Mielestäni käyttäjä saa hyvin tietoa siitä, millä välilehdellä kulloinkin ollaan.*

*Itselle jäi hieman epäselväksi, että onko aloitusruudussa (sivulla 10) mahdollista siis generoida sekä käyttäjänimi että salasana, vaiko pelkästään generoida salasana, sitten kirjoittaa siihen itse käyttäjänimi ja tämän jälkeen tallentaa kombinaatio?*

*Options-ruudussa (sivulla 11) on tällä hetkellä "number of digits" 1-20. Mikäli ymmärsin oikein, niin tuolla asetuksella saa salasanan jonka pituus on 1-20 merkkiä? Eli pitäisikö tuossa olla tarkempi valinta, kuinka monta merkkiä haluaa? Ja jos oletuksena on vain pienet kirjaimet ja kahdeksan merkkiä, niin voisiko ko. näkymässä olla myös vaihtoehto "small letters" ja siinä automaattisesti rasti paikallaan ja "number of digits"issä valittuna 8? Kun entä jos joku haluaa salasanan, jossa ei ole yhtään pientä kirjainta ja tasan 9 merkkiä :)*

*Help-ruutu on mielestäni looginen ja siitä saa hyvin selvän, mitä esimerkiksi erikoismerkeillä tarkoitetaan.*

*Suhteellisen hyvin ainakin itseäni ohjelma neuvoo liikkumaan eteenpäin, että ei tarvitse vain oman muistin varassa sitä käyttää. Mitä nyt tosiaan heti alussa mietin tuota käyttäjänimen antamista/pakollisuutta/generointia. Design vaikuttaa hyvältä, ei ole liikaa tilpehööriä tai muuta ylimääräistä höskää häiritsemässä käyttöä.*

*Hyvältä vaikuttaa! ”*

*– Miia Laitinen - Saturday, 1 October 2016, 3:33 PM*

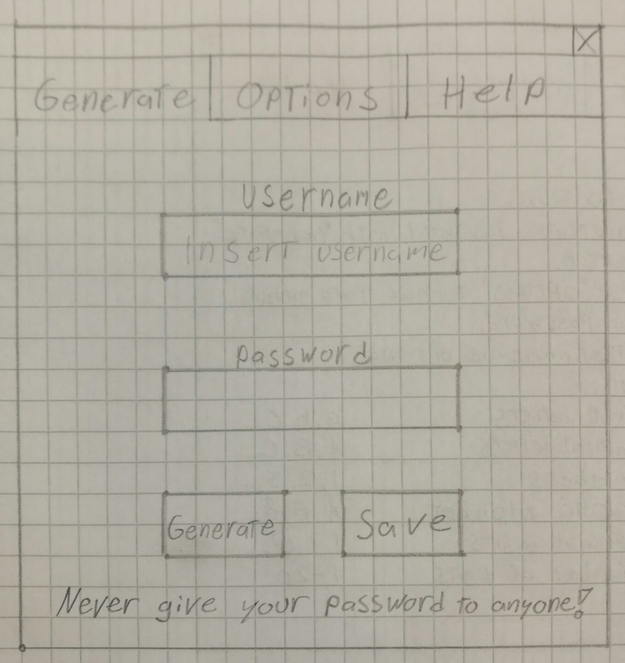
*”Moi!  
  
(offtopic: mahtavat esimerkkikäyttäjät! :D)  
Hyvä sovellusidea nykypäivään, kun painotetaan keksimään vahvoja salasanoja. Generaattorin avulla saa helposti itselleen monimutkaisen salasanan jota on hankala murtaa. Käyttöliittymä on selkeä ja sisältää oleelliset päävalikot (generaattori, asetukset ja ohje).  
  
Käyttäjänimi- ja salasanaloota päänäkymässä olisi hyvä erottaa näyttämään siten että ainoastaan käyttäjänimi-kenttään voi syöttää tekstiä. Protossa näyttää siltä että kumpaankin lootaan voisi syöttää tekstiä, kun ymmärtääkseni se on mahdollista vain usernamen kohdalla?  
  
Koska sovelluksessa on paljon tilaa, help-osion voisi jakaa muille näkymille? Voisiko help-osiosta siirtää esimerkiksi parametrien selitykset options-osioon, kun tilaakin on sovelluksessa melko paljon? Ei tarvitsisi hyppiä monen eri näkymän välillä.  How to use -osio voisi näkyä suoraan päänäkymässä. Tallennus epäonnistui -ikkunassa olisi myös hyvä näyttää miksi tallennus epäonnistui.  
  
Hyvä ja selkeä prototyyppi, pienillä lisäinformaatioilla tästä tulee hyvä salasanageneraattori.”*

*–Mikko Rouru - Saturday, 1 October 2016, 6:26 PM*

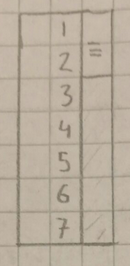
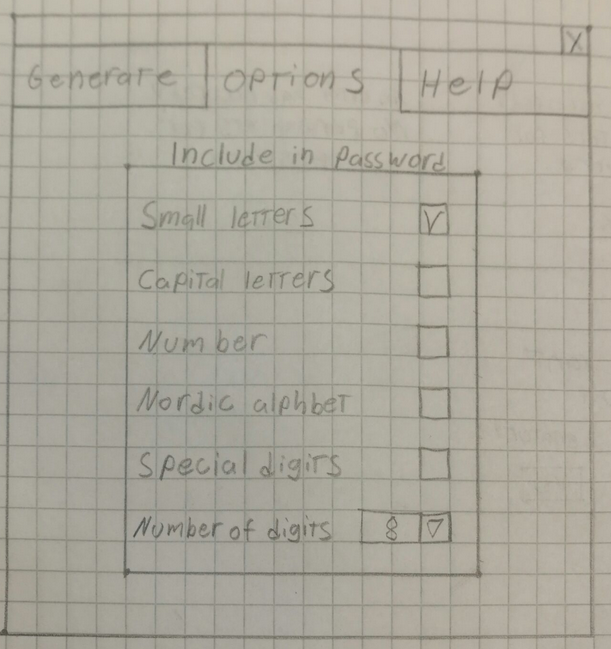
## Evaluointien löydökset ja vaikutukset

Käyttöliittymän arvioinnissa löydettiin ongelmakohtia, mutta mikään löydös ei ollut kriittinen ohjelman toiminnan kannalta. Tärkein löydös oli asetus sivulta puuttuva pienet kirjaimet valintaruutu, koska käyttäjä ei voinut poistaa niitä käytöstä. Tämä rajoitti ohjelman käyttöä, vain sellaisiin salasanoihin joissa oli aina mukana pieniä kirjaimia. Asetus sivulle ei lisätty tässä vaiheessa ”OK”-painiketta, koska asetetut parametrit astuvat voimaan aina kun käyttäjä painaa ”Generate”-painiketta. Myös virheilmoitusten ulkoasu korjattiin käyttäjäystävälliseksi, kun niihin lisättiin virheen syy. Käyttäjä ei aiemmin voinut tietää virheeseen johtanutta toimintaa. Ohjelmaan lisättiin oletusarvot, jotka selkeyttävät ja sujuvoittavat salasanojen generointia.

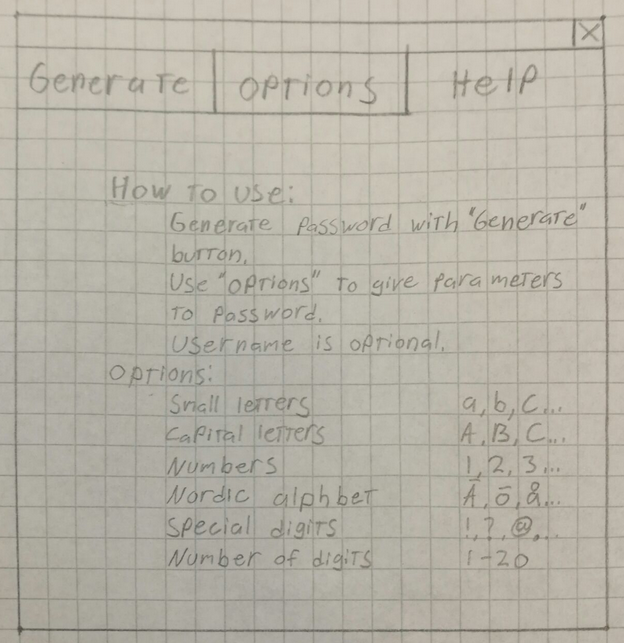
# Viimeistelty suunnitelma käyttöliittymästä



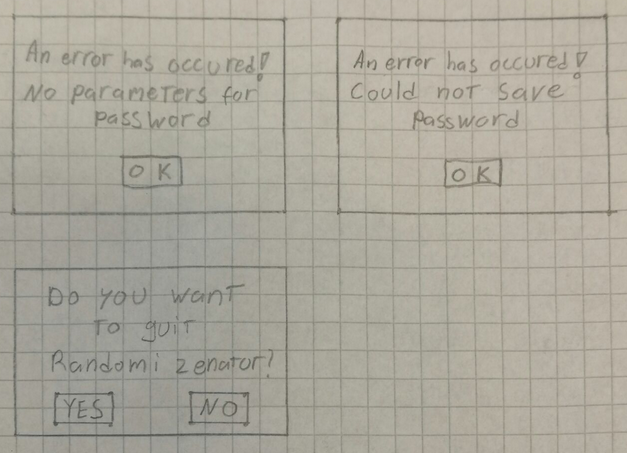
**Generate**



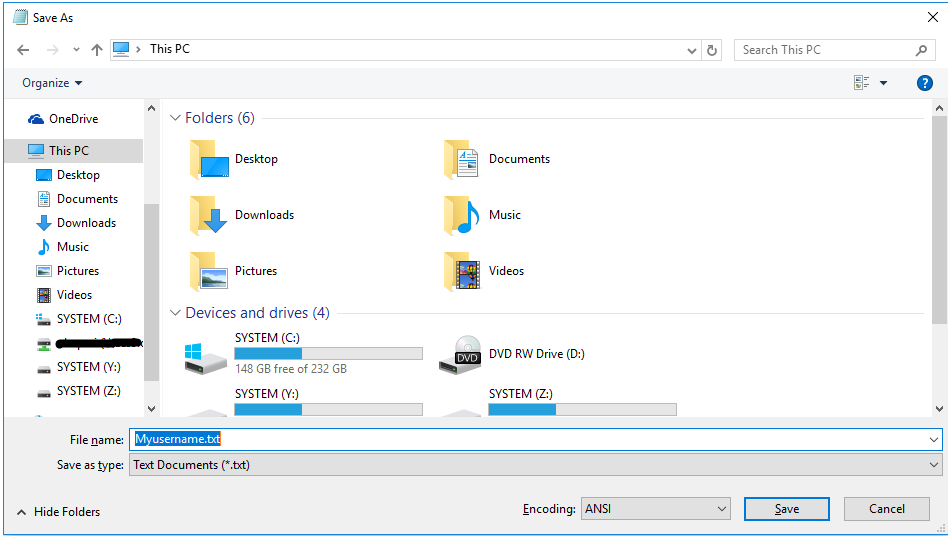
**Options**



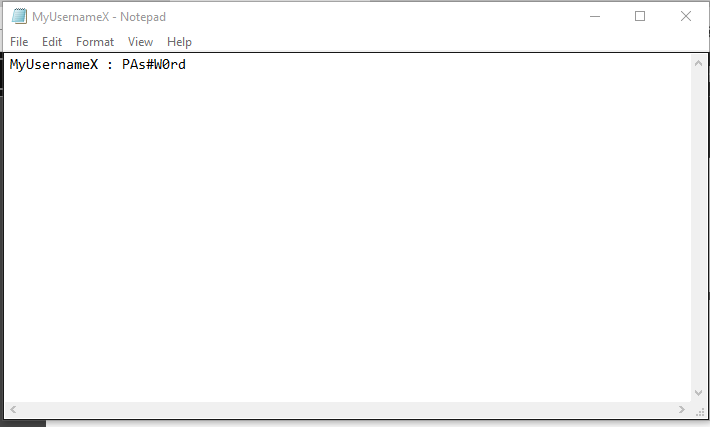
**Help**



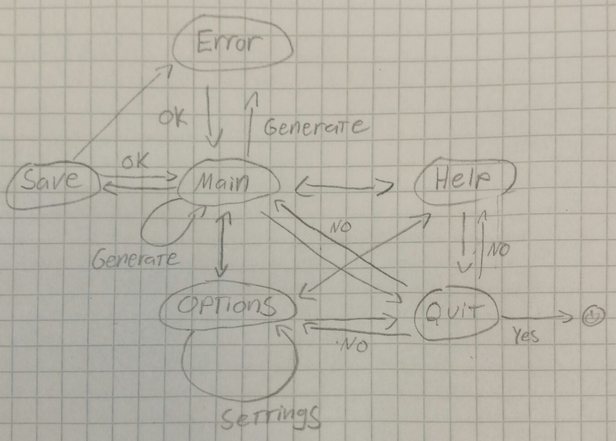
**Notifications**



**Save dialog**



**Saved file**



**Navigointi kartta**

# Käyttöohje

Randomizenator on helppo työkalu kaiken tasoisille käyttäjille.

Ohjelma käynnistyy ”Generate”-välilehteen johon käyttäjä voi lisätä haluamansa käyttäjänimen tai muuten salasanaa kuvaavan aputunnuksen, käyttäjänimen voi myös jättää tyhjäksi. Jos haluat vain nopean oletus salasanan (8 merkkiä ja vain pieniä kirjaimia), paina ainoastaan ”Generate”-nappia ohjelman käynnistyttyä ja olet valmis. Halutessasi voit tallentaa käyttäjänimi-salasana parin tekstitiedostoon(.txt) painamalla ”Save As”-painiketta. Ohjelma osaa liittää tiedoston perään itse tiedostomuodon, joten käyttäjä kirjoittaa vain valitun tiedoston nimen. Mikäli käyttäjä valitsee jo olemassa olevan tiedoston, tiedot tallennetaan tiedoston perään. Tiedostot tallennetaan muotoon:

'yourusername : yourpassword'

”Options”-välilehdeltä löydät salasanaan vaikuttavat parametrit, kuten merkkien määrän tai isot kirjaimet.

”Help”-välilehti sisältää samat tiedot kuin tämä, mutta lyhemmin.

Randomizenator koostuu kolmesta eri tiedostosta: Tehtava, Ikkuna ja Generator. Ohjelman suoritus käynnistyy, kun suoritetaan Tehtava.java tiedosto, mutta myös Ikkuna.java ja Generator.java täytyy olla mukana. Ohjelma on toteutettu Eclipse:llä joten on suositeltavaa suorittaa ohjelma siellä. Eclipse:llä voi kuitenkin luoda suoritettavan Randomizenator.jar tiedosto.

# Poikkeamat suunnitelmasta

Pitäydyimme suunnitelmassa Lähes täydellisesti. Ainoa ero suunnitelman ja toteutuksen välillä löytyy tallennusdialogista. Käytämme Javan tallennusdialogia, kun taas suunnitelmassa on käytetty Windowsin omaa tallennusdialogia. Päädyimme kuitenkin käyttämään Javan tallennusta, koska dialogit eivät eroa toiminnallisuudeltaan juuri lainkaan. Suunnitelmasta poikkeaminen ei siis aiheuttanut ohjelman toiminnallisuuteen minkäänlaista muutosta.

# Liitteet

Liite 1

**Randomizenator- käyttäjätestaus**

1. ”Generate”-välilehti
   1. Mitä katsoit ensimmäisenä?
      * + Katsoin ensimmäisenä yleisilmettä.
   2. Oliko nappien nimet ymmärrettäviä?
      * + Kyllä olivat.
   3. Oliko toiminnallisuudessa ongelmia?
      * + Ei, kaikki sujui mukavasti.
   4. Oliko ikkunan koko sopiva?
      * + Ikkunan koko oli mielestäni juuri sopiva.
   5. Luitko tekstit?
      * + En ole nyt varma mitä tekstejä tarkoitetaan. Luin kaikki välilehdellä olevat tekstit.
   6. Tarvitaanko pikanäppäimiä?
      * + Ei mielestäni.
2. ”Options”-välilehti
   1. Mitä mieltä olit sijoittelusta?
      * + Sijoittelu oli toimiva ja selkeä.
   2. Ymmärsitkö kaikkien vaihtoehtojen toiminnallisuuden?
      * + Englannin termit olivat vähän itselläni hakusessa, mutta help välilehdeltä löytyi apua niihin.
   3. Tarvittiinko pikanäppäimiä?
      * + Ei tai en henkilökohtaisesti käytä sellaisia yleensä.
3. ”Help”-välilehti
   1. Oliko teksti helppolukuista?
      * + Kyllä.
   2. Oliko parametreja avattu tarpeeksi?
      * + Oli. Parametrit olivat helposti ymmärrettävissä ja ”selitykset” selkeästi esillä.
4. Pop-upit
   1. Parameter error
      1. Luitko?

-En saanut kyseistä pop-uppia.

* + 1. Oliko tarpeeksi infoa?

-

* 1. Save error
     1. Luitko?

-Tallennus onnistui joka kerta.

* + 1. Oliko tarpeeksi infoa?

-

* 1. File saved
     1. Luitko?

-Vilkaisin.

* + 1. Oliko tarpeeksi infoa?

-Mielestäni oli tarpeeksi infoa. Tallennuspaikka oli selkeästi esillä.

1. Muut kommentit

-Mielestäni sovellus on erittäin hyödyllinen ja antaa apua pieneen arkiseen pulmaan. Asia, jota voisi vielä hioa on ”Generate” -välilehti. En itse ymmärtänyt, että mikä käyttäjänimi laatikkoon tulee laittaa. Voisiko siinä lukea, että ”Username you want to create a password/optional” tai jotain muuta vastaavaa. Kaikenkaikkiaan kuitenkin erittäin selkeä ja toimiva kokonaisuus!

Liite 2

**Randomizenator- käyttäjätestaus**

1. ”Generate”-välilehti
   1. Mitä katsoit ensimmäisenä?
      * + miten generate sivu on suuniteltu
   2. Oliko nappien nimet ymmärrettäviä?
      * + ymmärrettävät oli.
   3. Oliko toiminnallisuudessa ongelmia?
      * + Ei.
   4. Oliko ikkunan koko sopiva?
      * + Suhteellisen sopiva tälläiseen ohjelmaan..
   5. Luitko tekstit?
      * + Kyllä
   6. Tarvitaanko pikanäppäimiä?
      * + Ei.
2. ”Options”-välilehti
   1. Mitä mieltä olit sijoittelusta?
      * + Hyvin suunniteltu.
   2. Ymmärsitkö kaikkien vaihtoehtojen toiminnallisuuden?
      * + Kyllä.
   3. Tarvittiinko pikanäppäimiä?
      * + Ei.
3. ”Help”-välilehti
   1. Oliko teksti helppolukuista?
      * + sen lukeminen auttoi miten ohjelmaa toimii/käytetään.
   2. Oliko parametreja avattu tarpeeksi?
      * + Kyllä.
4. Pop-upit
   1. Parameter error
      1. Luitko?
         * Kyllä
      2. Oliko tarpeeksi infoa?
         * Kyllä
   2. Save error
      1. Luitko?
         * En
      2. Oliko tarpeeksi infoa?
         * En tiedä
   3. File saved
      1. Luitko?
         * Kyllä.
      2. Oliko tarpeeksi infoa?
         * Oli hyvä varmistus siitä missä se tiedosto on.
5. Muut kommentit
   * + - Kiva app.

# Harjoitustyön tekijöiden palautustiedot

Tekijän/tekijöiden tiedot:

|  |  |
| --- | --- |
| Nimi | Jussi Jokio  Niko Kupari |
| Hetu / op.numero | 2433950  2354080 |
| Tiedekunta ja laitos  (Aloitusvuosi) | Tietotekniikka 2014  Tietotekniikka 2014 |
| Sähköpostiosoite | [jussi.jokio@student.oulu.fi](mailto:jussi.jokio@student.oulu.fi)  [niko\_cu@hotmail.com](mailto:niko_cu@hotmail.com) |