

# Dokumentasjon: Reseptregister

*for prosjektoppgaven i programtutvikling 2013/2014*

```
s Doctor extends Person implements Serializable{
static final long serialVersionUID = 42L;
private boolean licenseA;
private boolean licenseB;
private boolean licenseC;
private String password;
private String employeeAt;

public Doctor(String firstNamePara, String lastNamePara, String personNrPara, String passwordPara, String telephoneNrPara, boolean l
        boolean licenseBPara, boolean licenseCPara, String employeeAtPara){
    super(firstNamePara, lastNamePara, personNrPara, telephoneNrPara);
    licenseA = licenseAPara;
    licenseB = licenseBPara;
    licenseC = licenseCPara;
    password = passwordPara;
    employeeAt = employeeAtPara;
}

public String getWorkplace(){
    return employeeAt;
}

public void setHashedPass(String passwordPara){
    password = passwordPara;
}
```

Sondre Husevold (S198755) Og Magnus Tønsager (S198761)

Mai 2014

# Innholdsfortegnelse:

Introduksjon: Reseptregisteret og hva det gjør	3
Funksjoner til programmet:	3
Hvordan systemet er bygget opp	4
Personsøk	7
Lister og de forskjellige registerne	7
Hovedregister klassen	8
Personer som objekter	9
Lagring, lesing og datahåndtering	9
Reseptskrivning	9
AdminGUI	10
TModel	11
StatisticTableModel	11
StartGUI	11
Obfuscering og passord	12
PharmacyGUI	13
Prescription	13
PrescriptionReg	13
PanelStack	14
Driver	14
Videre utvikling	14

## Introduksjon: Reseptregisteret og hva det gjør

Reseptregistret er et elektronisk reseptsystem laget i Java for leger og apotek. Det er ment som et universalt program for alle parter inkludert administrator for å holde styr på pasienter, leger og deres resepter.

Legene har mulighet til å skrive resepter til pasienter, så lenge de har lisens til legemiddelet. De kan opprette, modifisere og søke etter pasienter. Etter at en resept har blitt skrevet blir den lagret slik at apoteket kan hente den opp senere.

Apoteket har muligheter til å se pasienter og skrive ut resepten som har blitt satt på en pasient. Apoteket har mulighet til å søke igjennom resepter og pasienter, men ikke opprette noen av delene.

Administrator har rettigheter til å se både på reseptene, legene og pasientene. De har full rettighet til å endre pasienter, leger og apotekene, inkludert lisensene til legene. De kan ikke opprette eller slette resepter da dette er opp til legene. Administrator har mulighet til å se resepter skrevet ut til pasienter, hvilken lege som skrev den og kan vise en oversikt over antall resepter som er blitt skrevet ut per måned det siste året utifra en rekke kriterier.

Dette programmet var laget i håp om å få fjernet fysiske papirresepter både for sikkerhet, brukervennlighet og for å gjøre hverdagen til lege, pasient og apotek mye lettere.

### Funksjoner til programmet:

- Har et system som lagrer pasienter, leger, resepter og holder styr på de.
- Kan lagre resepter på pasienter slik at man kan ta ut resepten på alle apotek som bruker programmet.
- Et avansert søkesystem for lett navigering og finning av gamle eller nye resepter.
- Et oversiktlig og brukervennlig brukergrensesnitt laget spesifikt for apotekene, legene og administratorene.
- Har en godt strukturert reseptdatabase for lett tilgang til samme resepter uansett jobb.
- Bruker et sikkert innloggingssystem for å ta vare på leger og apoteks passord.
- Gir leger kun mulighet til å skrive ut resepter med reseptgrupper de har lisens for.
- Fungerer på alle operativsystemer som støtter Java, noe som gjør at det så å si er ingen systemkrav.

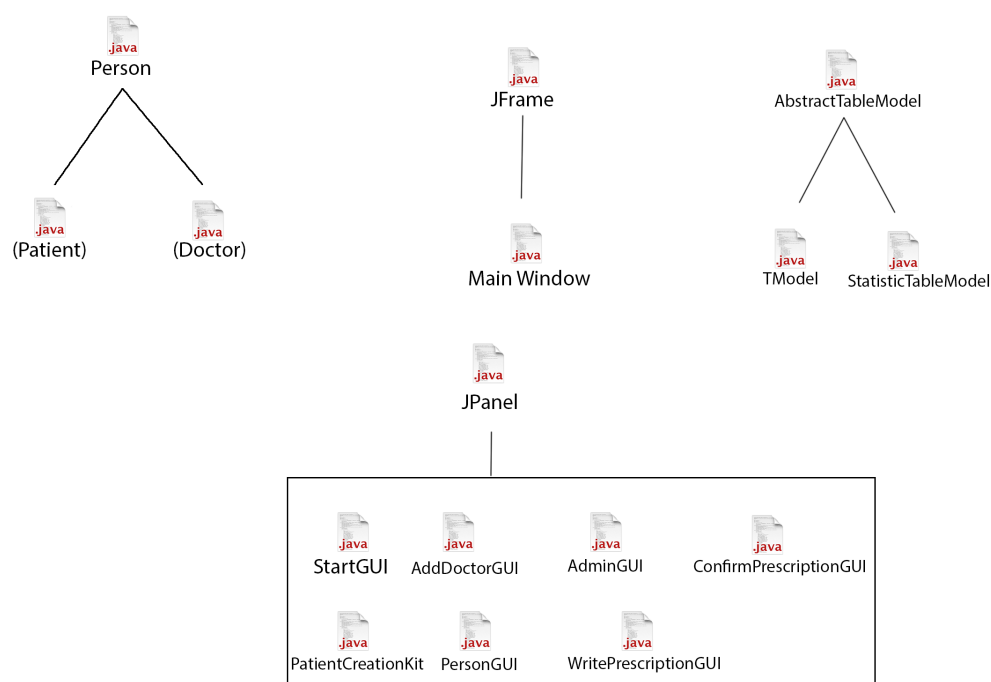
# Hvordan systemet er bygget opp

Systemet er bygget opp ved at man har en sentrert klasse kalt “Register” som tar hånd om mye av det programmet trenger å gjøre, og tar imot andre klassers informasjon. Ved å la den være sentrum i universet blir det lett å ta seg frem til andre klassers kode og metoder, noe som vil gjøre det lettere å utvide programmet senere.

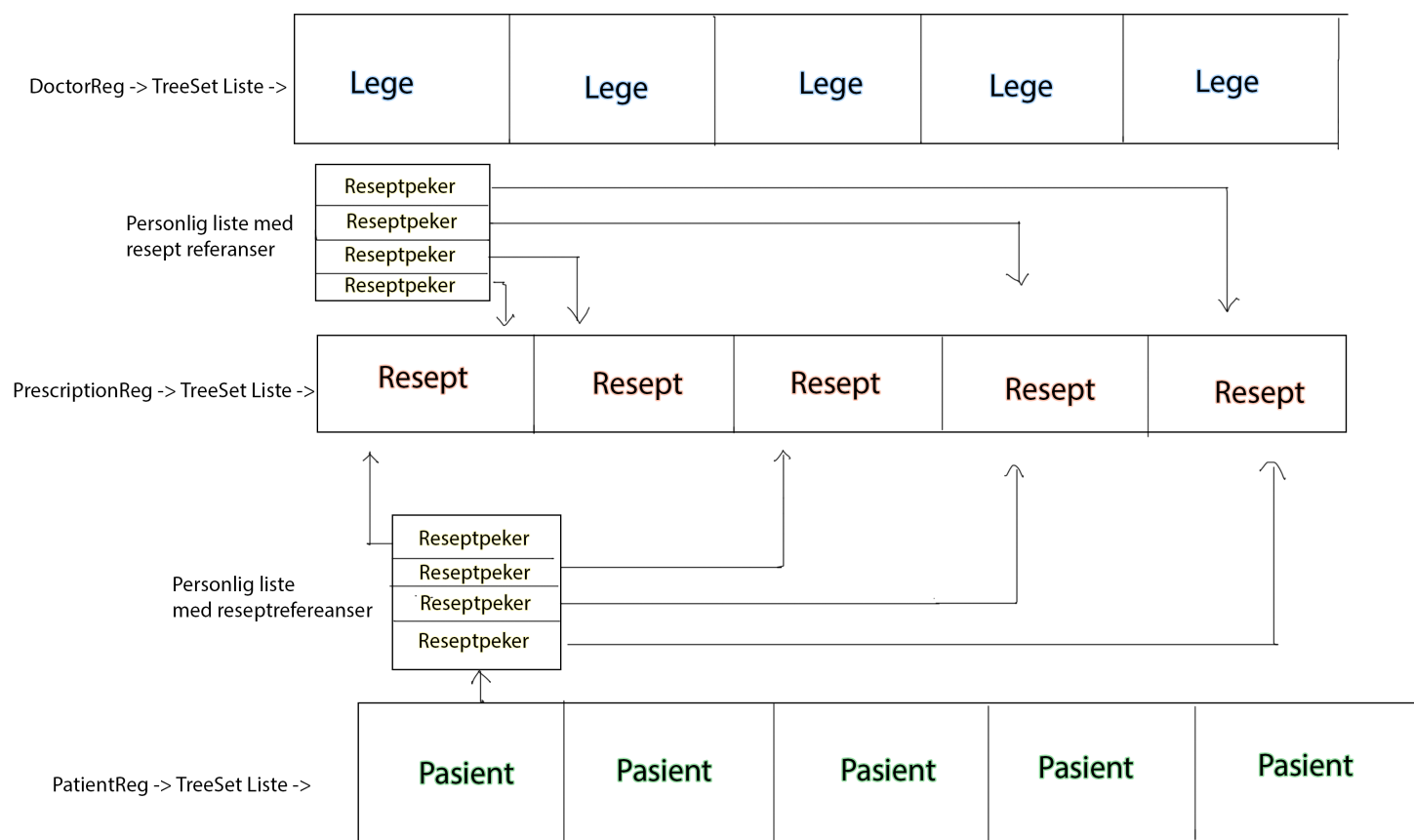
Øverst i hierarkiet finner man main metoden som oppretter et datamanager som leser av filene i “data” mappen eller lager en ny slik mappe med filer dersom disse ikke eksisterer. Deretter oppretter det en referanse til et register. Registeret er systemet i hele programmet og er et sentrum. Datamanager leser fra fil eller lager en ny fil og sender med det opprettede registeret videre og starter innloggingsvinduet med MainWindow (herved kalt mainframe) som er grunnsteinen i brukergrensesnittet.

Herifra kan det gå mange veier, enten om personen logger seg inn som lege, administrator eller apotek. Bytting av paneler blir gjort av mainframen og den tar også hånd om vinduplassering.

## Klassehierarki - Hvem extender hva







## Personsøk

Dersom de logger seg inn som en lege vil mainframen bytte panel til personsøkpanelet hvor man søker etter pasienter. Her er det fem felter satt av til forskjellige søkekriterier som legen kan søke seg til, det er også satt av knapper til å lage pasienter eller endre pasienter i tillegg til søk. I senter av hele personsøk panelet ligger en tabell som oppdaterer seg basert på søkekriteriene.

Søkekriteriene blir fikset via hovedregisterets pasientregisterklasse og filtrert før det blir sendt til personsøkevinduet tabell og tabellen vil oppdateres med den nye informasjonen. Derifra er det mulig å velge pasienter i tabellen eller eventuelt endre pasienten. Legen får presentert informasjonen om pasienten på en fin måte hvor all informasjon er tilgjengelig. De kan også stokke om på kolonnene om de så vil.

Personnummer	Fornavn	Etternavn	Telefon	Adresse
16087602828	Agnes	Bolstøy	27519271	Høystredslia 9E, 0205 Fruktb...
01110349232	Andreas	Granfjær	91289582	Bjerkeveien 11B, 0093 Jernv...
27098905212	Arne Bendik	Olsen	75232729	Fingerøyen 2E, 0668 Hol
08039185432	Bjarne	Betjent	12918282	Sesam Stasjon
26029908281	Charlotte	Johannessen	24462812	Høystadvika 2A, 0958 Bukktj...
03068319196	Dina Henriette	Frøyen	92858128	Godtstabukta 9B, 0221 Inger...
31038802582	Fatima Kjersti	Grønnli	92582181	Østringen 5D, 0502 Ytterlia
31118585721	Finn	Bjørk	92581852	Gyllenøy 16E, 0822 Bullsrud
24039485271	Frank Sølve	Ølbu	92581825	Mattestrøket 9E, 0095 Dilland
12058285711	Frøya	Strand	92582811	Barbaraveien 21C, 0592 Tøy...
30059405828	Gard	Grunnstad	25825711	Relvik 12B, 0214 Innhøy
02047602102	Heidi	Høybrekken	27566291	Idrettsbukta 6B, 0291 Yngen
30050792721	Illya	Kiritsuguvitch	92851781	Einsbergveien 7A, 0292 Skog...
15079668554	Janne Hanne	Bannesrøy	82588211	Ullensfryd 17B, 0611 Kallbak...
20069985211	Kirino	Kjåsaka	52174322	Akesvingen 12F, 0622 Bjørne...
25020885721	Lene Helle Aurora Andrea	Bukstad Karlensvik	25882967	Langveien 6B, 0096 Drøyen
11109885271	Martine	Hansen	92581821	Høystadsvingen 6C, 0852 Ull...
11126894282	Martine	Hansen	29582178	Introvertveien 15A, 0598 Fjæ...
01048502576	Max	Mekker	91281252	Sesam Stasjon
22079502851	Muhammad Abdi	Arganahsti	91281214	Gressveien 7E, 0122 Ilsjø
05092205732	Ragnhild Dagros	Røyen	85221632	Østre Granvik Aldershjem, 00...
11053492527	Reidun	Jaktsbo	85782185	Nattsrud 11A, 0352 Jansborg
15048295282	Sivert Emil	Tyland	28517201	Fagerbu 31E, 0776 Ekstrali
22129728578	Stian	Bakken	29581925	Jernveien 14E, 0258 Bakkens...
13058285711	Trine Henriette	Ungardfjell	92588211	Bukstavstedet 15E, 0799 Fin...
27128895282	Trude	Estmann	92582571	Barnehjørnet 2E, 0995 Gangf...
12033592852	Ystrad Riolkonov	Haitrokoströv	25718125	Ytre Østland 3C, 0721 Inløv

## Lister og de forskjellige registerne

Resepter, doktor og pasientregisterne er objekter som blir opprettet av hovedregisteret og blir lest opp når datamanager leser registeret. Fra disse tre klassene har man tilgang til TreeSet lister som inneholder disse personene. Disse tar også hånd om manipulering av disse listene som å legge til nye personer eller resepter, fjerner personer, gir antallet objekter som ligger listen, kan skrive ut alle

personene til en tekst og kan returnere iteratorere for å gå igjennom lista. Disse har også metoder som søker på personene i lista og returnerer et nytt pasient eller doktorregister av samme type med referanser til personene som ble filtrert. Dette gjør det lett å utvide senere ved at man alltid får objektene det er snakk om og kan iterere, manipulere eller søke i disse filtrerte listene for å søke enda dypere.

Det er også laget en kollator for personene og reseptene. Personene blir sortert med tanke på navn og blir dobbeltsjekket dersom personnummeret eksisterer med noen som allerede har et likt personnummer. Dermed slipper man eventuelle feil ved at noen kan legge inn samme person flere ganger. Personer kan likevel hete det samme så lenge de ikke har samme personnummer som blir en unik identifiseringfaktor for personene. Reseptene blir organisert etter reseptnummer som øker for hver resept som blir skrevet ut.

Det er små forskjeller på disse klassene hvorav reseptene ikke har mulighet til å fjerne resepter fra listen. Dette er gjort med hensikt, da resepter ikke skal kunne bli fjernet av noen, inkludert administrator og blir alltid lagret i databasen for sikkerhet og historikk.

### Hovedregister klassen

Programmets “Register” klasse er som sagt programmets hovedklasse. Klassen tar seg av all kommunikasjon mellom GUI klassene og listeklassene, de fleste søke og sorteringsmetodene ligger her.

Øverst i klassen finner du get-metoder for å hente ut klassens datafelter. Under ligger add metodene for å legge inn nye leger, pasienter og resepter. Videre inneholder klassen et filtreringssystem som kan brukes for søk i de forskjellige registrene. Disse fungerer slik at de tar imot en liste og et søkekriterium. Dersom den innkommende listen er null, bruker den listen som ligger i klassens eget datafelt. Den oppretter så en ny tom liste og legger alle elementene som stemmer med søkekriteriet til i den nye listen. Listene skal være av typen PrescriptionReg, DoctorReg eller PatientReg og det finnes tilsvarende metoder for søk i alle tre listetyper.



## Personer som objekter

Personene er laget som en abstrakt klasse hvor person holder på ting som er felles for både leger og pasienter som dermed inkluderer ting som fornavn, etternavn, personnummer og telefonnummer. De har også en referanseliste som inneholder pekere til reseptene de er tilknyttet, dermed forhindrer vi dobbellagring ved at det kun er pekere som peker til et felles reseptobjekt. Det blir også utrolig lett å finne igjen hvilke resepter som har blitt gitt til hvem. Leger og pasienter blir dermed utvidede klasser utifra personklassen. Pasienter har kun ett ekstra felt utover person som vil være deres hjemmeadresse.

Leger har i tillegg til personfeltene tre lisenser, A, B og C som kan slås av og på dersom dette er nødvendig. Dette forhindrer legene i å kunne registrere resepter som ikke de har rettigheter til å skrive ut. De har også et adresse felt for hvor de jobber.

## Lagring, lesing og datahåndtering

Datamanager er en egen klasse som tar hånd om all form for lagring og lesing. Denne leser registeret som inneholder alle de tre andre delregisterne: Pasientregisteret, doktorregisteret og reseptregisteret. Det er satt inn unntakstilstander her dersom ikke mappen eksisterer eller filene, dersom informasjonen er korrupt, eller andre feil som kan oppstå ved lagring eller lesing av data.

## Reseptskrivning

Når en lege skal skrive en resept, søker han eller hun etter pasienten i personsøk vinduet, velger pasienten, og trykker på “skriv resept” knappen hvor man blir kastet videre til reseptskrivingsvinduet. Når dette skjer blir både pasienten og legen sendt videre som objekter til reseptskrivingsvinduet. Dette vinduet leser av legens bevilgninger og gjør klar pasient og lege til reseptlagring. Det tar imot alle tekstene som blir skrevet inn i feltene og sender de videre som parametere videre til bekreftelsesvinduet sammen med legen og pasienten.

Når man kommer til bekreftelsesvinduet vil resepten bli vist svart på hvitt i tekstformat. Herifra kan legen velge å ikke sende resepten, eller sende den. Dersom den blir sendt blir den opprettet som et nytt reseptobjekt ved bruk av parameterne som ble sendt med fra resepskrivingsvinduet og lagret som referanse i reseptregisteret, doktors referanseliste og pasientens referanseliste.

## AdminGUI

AdminGUI klassen er et JPanel med diverse søke og redigerings metoder. Panelet er ment for administratoren, og har metoder for å søke opp ønsket informasjon, vise statistikk og endre data i programmet. Klassen bruker Register sitt søkefilter for å finne ønsket informasjon om resepter, leger og pasienter. Klassen bruker TModel for å vise informasjon om resepter, leger og pasienter og StatisticTableModel for visning av statistikk.

Reseptregister

Her kan du finne og vise ulik info om resepter, leger og pasienter. Under søk etter kan du velge om du vil søke etter resept, lege eller pasient. Du bestemmer søkekriteriene ved å fylle ut feltene under, tomme felter blir ikke tatt med i søket

Søk etter:

☒ Resepter  
☐ Pasienter  
☐ Doktorer

Pasient navn/personnummer:

Doktor navn/personnummer:

Medikament navn:

Medikament kategori:

Forskrevet medikament i gruppe:

☒ Gruppe A ☒ Gruppe B ☒ Gruppe C

Søk

Legg til en ny doktor

Vis statistikk

Reseptnummer	Skrevet ut	Hentet	Pasient	Lege	Medisin navn	Medisin kategori	Medisin gruppe
--------------	------------	--------	---------	------	--------------	------------------	----------------

Vis resept

Tilbake Logg ut Fram

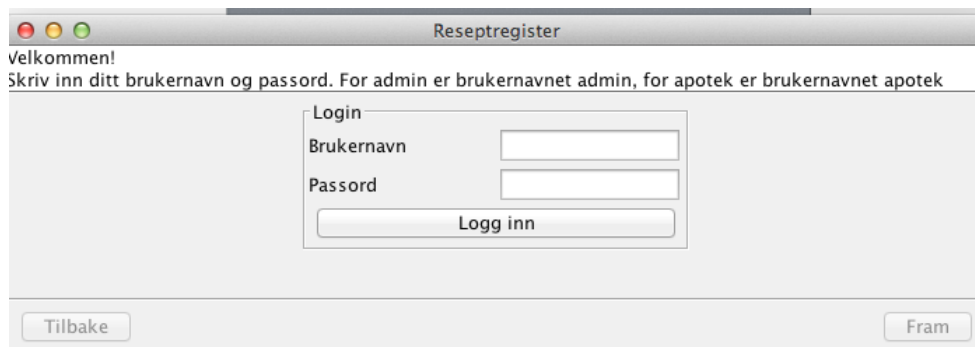
## TModel

Dette er modellen for den mest brukte tabellen i programmet. Den har tre konstruktører, en for å lage resepttabell, en for å lage lege tabell og en for å lage pasienttabell. Dataen til modellen blir opprettet i konstruktøren. Klassen er en utvidelse av AbstractTableModel og implementerer alle nødvendige og anbefalte metoder derfra. I tillegg har klassen metode for å sette en DefaultCellEditor for endring av String verdier og en metode for å lagre all data i modellen.

## StatisticTableModel

Dette er modellen programmet bruker til å vise statistikk over antall utskrevne resepter per medikament over et år. Den er en utvidelse av AbstractTableModel og implementerer alle nødvendige og anbefalte metoder derfra. Det er ikke mulig å endre dataene i denne modellen

## StartGUI

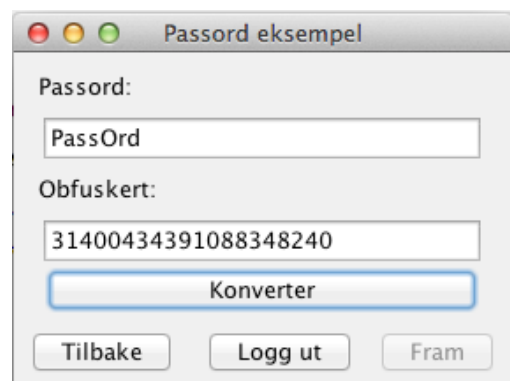


Denne klassen utvider JPanel med tekstfelter for brukernavn og passord og har en logg inn knapp. Klassen har metoder for å logge inn riktig bruker og inneholder brukernavn og passord for admin logg inn og apotek logg inn. Det er menning at programmet etter hvert bør utvides med egne brukerklasser for administrasjon og apotek, men vi har ikke hat tid til dette. Passord og brukernavn for administrasjon og apotek ligger hardkodet i StartGUI klassen, men passordene er kryptert via Obfuscation sin “kryptering”.

Det er ikke mulig å endre passord via brukergrensesnittet på nåværende tidspunkt, men det er lagt opp til at dette skal gjøres hvis det blir implementert en egen klasse for brukere. Passord til en lege ligger kryptert i legeobjektet og bestemmes når legen opprettes, vi har heller ikke implementert noen funksjon for å endre dette passordet etter opprettelsen.

## Obfuskering og passord

Ved bruk av en klasse kalt Obfuscation vil programmet kunne obfuskere String objekter til en lang rekke med tall. Dette blir gjort ved at klassen bruker metoden readPassword som tar imot et String objekt. Deretter blir denne lest igjennom av et scanner objekt og gjort om til bokstav objekter. Bokstavene blir vurdert om de er store eller små og vil derfra bli konvertert til en tallverdi basert på bokstavens unikode. Hver gang scanneren leser en bokstav vil et heltall kalt iteration bli satt opp et hakk for å kunne tukle med tallverdien til bokstaven ved bruk av modulo, deling, pluss og minus. Dermed vil alltid tallverdien til neste bokstav være forskjellig fra den forrige selv om det er samme bokstav som har blitt skrevet inn flere ganger på rad. Det vil bli gjort ekstra tukling med store bokstaver for å skille de fra små.



Etter at alle bokstavene har blitt konvertert til en lang tallrekke vil programmet kaste den over til en reverseringsmetode. Her vil metoden splitte hele tallrekka hver gang den finner tallet 2 og 5 til en String array. Deretter vil den reversere hele tallrekka og sette inn tall der splittelsene foregikk basert på en iterator som er lengden på String arrayen og starter på siste splittelsen og går nedover til den har reversert og puttet inn tall i hele String objektet den fikk og returnerer denne. Deretter vil readPassword returnere det ferdige obfuskerede String objektet hvor det enten lagres i en lege, eller blir brukt av StartGUI til å sammenlikne passordet som blir skrevet inn med legen sitt passord.

## PharmacyGUI

Denne klassen utvider JPanel med funksjoner for søk etter og uthenting av resepter. Klassen bruker Register sitt søkefilter for å finne ønsket informasjon om resepter. Når brukeren har valgt en resept, har panelet funksjoner for å åpne resepten, dette viser brukeren all informasjon om resepten, og brukeren kan da registrere resepten som hentet. Tabellen i apotekvinduet viser ikke resepter som allerede er hentet.

## Prescription

Prescription definerer en resept og inneholder datafelter for all relevant data. Klassen har også datafelt for lege og pasient. Vi har valgt å lagre medikamentgruppe som en Integer. Disse er definert som static final konstanter og har public access.

## PrescriptionReg

PrescriptionReg inneholder en TreeSet<> over alle reseptene programmet har lagret. Metodene til klassen gjør operasjoner på denne listen. Klassen har også en metode for å konvertere reseptgruppe konstantene fra en Integer til tilhørende String og motsatt.

Det er definert en egen Comparator, PrescriptionCollator, for TreeSet listen. Denne sørger for at det ikke kan være to resepter med samme reseptnummer i listen. Den sorterer også reseptene etter reseptnummeret og siden dette øker med en per nye resept som opprettes, blir listen også sortert etter tidspunkt resepten ble opprettet.

## PanelStack

Denne klassen inneholder metoder for frem og tilbake navigering mellom tidligere paneler. Den brukes av MainWindow sine frem og tilbake knapper. Klassen består av to lister og to iteratorer, en for hver liste. Den ene listen lagrer de tidligere panelene i riktig rekkefølge og den andre lagrer paneler når klassens tilbake funksjon blir brukt. Altså den ene listen er for å navigere bakover, den andre fremover. Klassens MAX\_SIZE konstant avgjør hvor mange paneler den skal huske.

## Driver

Inneholder programmets main metode og oppretter et datamanager objekt for lesing av filer.

## Videre utvikling

Programmet innfrir alle ønskene til arbeidsgiver, men det er likevel ting vi følte burde være med utover dette. Dette kan være ytterligere funksjoner som kan bygges på i programmet for å gjøre det bedre og mer aktuelt for bruk i arbeidslivet.

Et databasesystem som holder rede på resepter, leger og pasienter via nettet eller på en annen måte slik at alle legene har alle pasientene tilgjengelig.

Et system som lagret eller hentet inn medisinene fra Internett for lesing av reseptgruppe og annen informasjon hadde vært nyttig. Dermed hadde man ikke trengt å skrive inn kategori eller reseptgruppe og kunne fått en mal på anvisningen. Dette hadde det gjort lettere for leger som hadde måtte skrevet inn den samme resepten mange ganger om dagen og hadde forhindret at apoteket må dobbeltsjekke at legen har autoritet til å faktisk skrive resepten de har skrevet ut.

Mer dynamiske søkefelt som kunne funnet pasienter og doktorer mens man skriver inn. Dette ville helst vært noe som liknet på Google sitt søkeprediksjon hvor de estimerer hva du har lyst til å søke på mens du skriver. Personsøk

vinduet har allerede implementert at man ikke trenger å skrive inn hele navnet perfekt for at den skal søke etter så dette hadde vært å tatt det til neste steg.

Kryptering av passord, personlig informasjon og annet burde bli gjort for sikkerhet. Dette var noe vi ønsket å gjøre, men ikke fikk tid til. Vi har heller ikke den nødvendige kompetansen til å lage dette selv eller tøyler til å importere det da det ikke var lov å bruke plugins.

Et litt mer unikt brukergrensesnitt hadde også vært bra, da det nåværende brukergrensesnittet føles litt tamt ut. Det skulle gjerne hatt litt mer farger og gått litt vekk fra Javas standard brukergrensesnitt farger og knapper.