# Funksjonelle krav:

**ParkX.Innlogging**

01. En bruker skal identifiseres i systemet ved bruk av e-post adresse.

02. Brukeren skal ha mulighet til å resette passordet sitt gjennom en lenke de får tilsendt på epost, ved å trykke på glemt passord knapp i innlogingsmenyen.

03. En ny bruker skal kunne registrere seg inn i systemet ved å fylle ut et registreringsskjema.

**ParkX.Personvern**

01. Brukere skal kunne fjerne seg fra hele tjenesten om de ønsker det, ved å tryke på   
 slett bruker knappen i brukerinstillinger.

02. Administrator skal kunne utestenge brukere fra tjenesten, ved å bruke menyen som bare administratorer har tilgang til.

**ParkX.Tilbakemeldingssystem**

01 Brukere skal kunne skrive klager, ved å bruke eget skjema tilegnet klager.

02. Brukere skal kunne se klager, ved å se klagehistorikken i en egen klagemeny.

02.01 Brukere skal se klager de skriver, ved at klagene de får opp i klagehistorikken har en «skrevet» tagg ved siden av tittelen.

02.02 Brukere skal se klager de mottar, ved at klagene de får opp i klagehistorikken har en rød «motatt» tagg ved siden av tittelen.

03. Brukere skal kunne rapportere andre brukere ved brudd av reglementet, ved hjelp av klagesystemet.

03.01. Brudd på reglementet skal skrives i klage ved å oppgi brukernavnet til gjerningspersonen, tidspunktet og en kort beskrivelse i et eget skjema.

04. Utleiere skal kunne blokkere visse brukere fra å leie plassen igjen (Viktig hvis utleieren ikke var fornøyd med kunden)

05.Brukere skal kunne gi hverandre rating, ved å bruke ratingsystemet etter enhver leie

05.01. Leier skal kunne gi parkeringsplassen den utleide en terningkast basert på hvor fornøyd den var med plassen.

05.02. Utleier skal kunne gi leieren et terningkast basert på hvor fornøyd utleieren var med den som leide plassen.

**ParkX.Leie-plassen**

01. Leiere skal kunne leie en parkeringsplass, ved å få opp en liste over plasser  
//Implementert  
//Testes i test\_ControllerIntegration.py -> test\_returnsListFromRepositoryProperly()  
//Testes også i //test\_controllerSendsRequestToChangeParkingPlaceStatusAndSavesStartDateCorrectly()

02. Leiere skal kunne reservere parkeringsplass frem i tid. (Dette vil være hensiktsmessig for de som planlegger fremtidige besøk)

03. Leiere skal kunne utvide parkeringstid på en reservasjon, i ledig tidsrom.

04. Leieren skal kunne sette seg på venteliste dersom ønsket parkeringsplass er reservert.

04.01 Leieren skal få en mulighet til å få varsel når parkeringsplassen blir ledig igjen.

05. En leier skal kunne starte en tilgjengelig parkering med en gang, og betale med en gang den stanses  
//Implementert  
//Testes i test\_ControllerIntegration.py ->   
//test\_controllerReturnsCalculatedPriceForParkingBasedOnTimePassedSinceParkingStartCor-//rectly()

05.01. En leier skal kunne stoppe en parkering de leier ved å trykke på «stopp parkering» knappen ved siden av parkeringen under «mine parkeringer» menyen.  
//Implementert  
//Testes i test\_ControllerIntegration.py ->  
//test\_controllerSendsRequestToChangeParkingPlaceStatusAndSavesStartDateCorre-//ctly()

06. Utleier skal kunne tilby langtidsleie, ved å markere parkeringsplassen som «til månedlig leie».

07. En leier skal kunne se detaljene på en parkeringsplass før den leier plassen  
//Implementert  
//Testes i test\_ControllerIntegration.py -> test\_controllerGetsSpecificParkingPlaceWithId()

**ParkX.Utleie-plassen**

01. Utleier skal kunne legge til en ny parkeringsplass, med detaljer angående miljøet og stedet som plassen befinner seg i, ved å fylle ut et «registrer ny parkeringsplass» skjema i hovedmenyen til utleier.  
//Implementert  
//Testes i test\_ControllerIntegration.py  
// -> test\_receivesDictionaryFromUserSavesItInRepositoryAndCreatesTheObjectProperly()

02. Utleier skal kunne slette egne plasser, ved å trykke på «fjern tilbud» knappen som befinner seg ved siden av enhver parkeringsplass i «min profil» visningsmenyen.

02.01. Utleier skal kunne slette egne plasser individuelt, ved å bare trykke på fjern tilbud knappen på en enkel parkeringsplass.  
//Implementert  
//Testes i test\_ControllerIntegration.py -> //test\_receivesIdFromUserAndDeletesCorrectParkingPlaceObject()

02.02 Utleier skal kunne slette egne plasser i mengder, ved å velge flere plasser for sletting samtidig.

03. Utleier skal kunne endre detaljene på en parkeringsplass, ved å trykke på endre detaljer knappen som befinner seg i den detaljerte visningen av en parkeringsplass.  
//Implementert  
//Testes i test\_ControllerIntegration.py -> //test\_controllerChangesParkingPlaceAttributesProperly()

04. Utleier skal kunne midlertidig deaktivere plassen sin for utleie (Dette er nødvendig dersom utleieren blir nødt til å bruke plassen selv)

05.Utleier skal kunne få en oversikt over alle plassen som den har lagt ut til utleie, ved å gå inn på dens profil-visning.

06. Utleier skal kunne se hvem som leier plassen nå, ved å gå inn i utleie-historikk menyen.

06.01 Utleier skal kunne se hvem som har leid plassen i fortiden, ved å gå inn i utleie-historikk menyen

**ParkX.Søking**

01. Leier skal kunne søke på en plass i et ønsket geografisk område, ved å bruke filter by eller avstand fra nåværende posisjon

02. Leier skal kunne søke på en plass i et ønsket miljø, ved å bruke checkboksene ved siden av søketreffene. (Et miljø er altså om parkeringsplassen befinner seg inne/ute, eller om det er med lader, lys o.s.v)

**ParkX.Historikk**

01. Leier skal kunne få en oversikt over aktive og tidligere parkeringer, ved å gå inn i historikk menyen.

02. Brukere skal kunne se eldre betalinger, ved å gå inn i betalingshistorikk i historikk menyen.

**ParkX.Administrator**

01. En Administrator skal ha alle rettigheter som andre brukere har.

02. En Administrator skal ha alle nødvendige rettigheter til å opprettholde orden og rettferdighet i systemet, ved å logge inn som en spesiell bruker.

**ParkX.Økonomi**

01. Brukere skal legge inn kort for automatisk trekk, ved å legge til kortet i min profil menyen.

02. Brukere skal kunne trekkes månedlig for langtidsleie, ved å trekkes automatisk fra kortet.

02.01 Brukere skal kunne få tilsendt faktura, dersom de ikke har lagt til kortet sitt for automatisk trekk.

03. Utleiere skal få månedlige utbetalinger i form av totalsum, samlet opp gjennom hele måneden, ved at systemet sender ut pengene en dag i måneden.

04. Brukere skal kunne få en liste med alle gjenstående betalinger som de har å betale ned, ved å gå inn på min profil vinduet.  
//Implementert  
//Testes i ControllerIntegration.py -> test\_can\_get\_list\_with\_all\_payments()

05. Brukere skal kunne betale gjenstående betalinger, ved å gå inn på min profil vinduet og trykke på «betal alle gjenstående parkeringer»  
//Implementert  
//Testes i ControllerIntegration.py -> //test\_can\_empty\_all\_payments\_from\_list\_if\_acceptedPaymentDetails\_is\_true()  
//+ ->test\_can\_not\_empty\_all\_payments\_from\_list\_if\_acceptedPaymentDetails\_is\_false()

**ParkX.Validering**

1. Brukere skal få tilbakemelding dersom de fyller ut et skjema feil.
2. Brukere skal få tilbakemelding dersom et skjema de fyller ut inneholder minst et tomt felt.

//Implementert  
//Testes i ControllerIntegration.py -> //controllerRaisesValueExceptionIfInputFieldsWhereIntExpectedIncludesLetters()

1. Brukere skal få tilbakemelding dersom et skjema de fyller ut inneholder bokstaver steder hvor bare tall forventes.  
   //Implementert  
   //Testes i ControllerIntegration.py ->   
   //test\_controllerRaisesUserWarningIfInputFieldIsEmpty()

02. Brukere skal få tilbakemelding dersom feil oppstår under betaling.  
//Implementert  
//Testes i ControllerIntegration.py -> //test\_can\_get\_accepted\_payment\_details\_state\_from\_Payment()

# Ikke-Funksjonelle krav:

**ParkX.Data**

1. Systemet skal samle data, ved å sende statistikk til eieren av systemet, en gang i måneden.
2. Systemet skal lagre data i MYSQL.

**ParkX.System**

1. Systemet skal skrives i Python 3.7.
2. Systemet skal kodes på engelsk, men brukergrensesnittet skal skrives på Norsk.
3. Systemet skal kunne kjøres på datamaskin, nett og android, ved å bruke kivy rammeverket.
4. Systemet skal max bruke 3 sekunder per scene switch.
5. Systemet skal ikke bruke mer enn 1GB RAM minne.
6. Systemet skal kreve at en bruker identifiseres seg før den får lov til å bruke systemet.

**ParkX.Sikkerhet**

1. Systemet skal kryptere alt dataen, som sendes ut av systemet.
2. Systemet skal ta vare på personvern, ved å følge GDPR regelverket.

**ParkX.Tilgjengelighet**

1. Den nettbaserte delen av systemet skal være universelt utformet og tilgjengeligjort etter WCAG-2.0 standarden.
2. Systemet skal vær enkelt å bruke, slik at brukere trenger hjelp med systemet gjennomsnittlig en gang per 6 måneder.

**ParkX.Driftsikkerhet**

1. Systemet skal ha en feilfrekvens(ROCOF) på mindre enn 1/100 000, ved hjelp av gode feilbehandlinger.
2. Systemet skal være tilgjengelig(AVAIL) 99.5% av tiden.
3. Systemet skal takle feil, slik at den ikke bruker mer enn 5 minutter på å restarte og bli brukbart igjen.