YRKESHÖGSKOLAN
NOVIA

GDPR–Implementering i ett bokningssystem

Oscar Friman

Examensarbete för ingenjörs (YH)–examen

Utbildningsprogrammet för informationsteknik

Vasa 2018

EXAMENSARBETE

Författare:	Oscar Friman
Utbildning och ort:	Informationsteknik, Vasa
Handledare:	Susanne Österholm

Titel: GDPR–Implementering i ett bokningssystem

Datum: 30.09.2018 Sidantal: 23

Abstrakt

Detta examensarbete har utförts på begäran av mjukvaruföretaget Hogia Ferry Systems. Företaget utvecklar mjukvara för hanteringen av data om passagerare samt frakt inom sjötrafik runt hela världen. Arbetet gick ut på att identifiera samt implementera ändringar till hanteringen av persondata i bokningssystemet BOOKIT, för att göra det möjligt för användarna att följa EU:s nya dataskyddsförordning, även känd som GDPR.

Resultatet av arbetet blev en konfigurerbar implementering av funktionalitet för pseudonymisering samt radering av persondata, hantering av medgivande, loggning och rapportering av personuppgiftsbehandling. Med denna funktionalitet bör kunderna ha möjlighet att följa GDPR.

Språk: svenska Nyckelord: GDPR, Hogia, Visual Basic, BOOKIT

OPINNÄYTETYÖ

Tekijä:	Oscar Friman
Koulutus ja paikkakunta:	Tietotekniikka, Vaasa
Ohjaaja:	Susanne Österholm

Nimike: GDPR:n toteutus varausjärjestelmässä

Päivämäärä: 30.9.2018 Sivumäärä: 23

Tiivistelmä

Tämä opinnäytetyö on tehty Hogia Ferry Systemin pyynnöstä. Hogia Ferry Systems tekee ohjelmia merenkulun matkustajatietojen ja rahtitietojen käsittelyä varten ympäri maailmaa. Työn päätehtävään kuuluivat henkilötietojen käsittelyn muutosten tunnistaminen ja toteuttamien varausjärjestelmä BOOKIT:issa. Tämä mahdollistaisi käyttäjiä toimimaan EU:n uuden tietosuoja-asetuksen mukaisesti. Tietosuoja-asetus on myös tunnettu nimikkeellä GDPR.

Työn tuloksena on joustava toimivuus henkilötietojen pseudonymisoinnissa ja poistamisessa, suostumuksen käsittelyssä sekä henkilötietojen käsittelyssä niitä kirjattaessa lokiin. Tämän toiminnon myötä asiakkailla on mahdollisuus noudattaa GDPR:n määräyksiä.

Kieli: ruotsi Avainsanat: GDPR, Hogia, Visual Basic, BOOKIT

BACHELOR'S THESIS

Author:	Oscar Friman
Degree Programme:	Information Technology, Vasa
Supervisor(s):	Susanne Österholm

Title: GDPR–Implementation in a Booking Software

Date: September 30, 2018 Number of pages: 23

Abstract

This thesis was done on behalf of Hogia Ferry Systems. Hogia Ferry Systems is a software company developing solutions for freight- and passenger shipping companies around the world. The thesis entailed identification and implementation of changes to the processing of personal information in the booking software BOOKIT. This was done to allow the users of BOOKIT to follow EU's new regulation regarding personal data, also known as GDPR.

The thesis resulted in a highly configurable implementation of functionality allowing for pseudonymisation and deletion of personal information, handling of consent, logging of personal data processing and a report of personal data processing. This new functionality should give the customers the ability to follow GDPR.

Language: Swedish Key words: GDPR, Hogia, Visual Basic, BOOKIT

Innehållsförteckning

1 Inledning

I detta kapitel presenteras uppdragsgivaren Hogia Ferry Systems, mjukvarukoncernen Hogia samt uppdraget som arbetet baseras på.

1.1 Uppdragsgivare

1.1.1 Hogia

Hogia är en mjukvarukoncern som grundades år 1980 av Bert-Inge Hogsved. Hogia består idag av 27 företag inriktade på mjukvara inom ekonomi-, personaladministrativa- samt transportsystem. Hogia sysselsätter idag ca 600 personer i Sverige, Finland, England samt Norge, och har en omsättning på ca 500 miljoner svenska kronor (Hogia, 2018).

1.1.2 Hogia Ferry Systems

Det finländska företaget Oy Consy Ab grundades år 1988 av Stefan Engelholm och då påbörjades även utvecklingen av bokningsmjukvaran BOOKIT. År 2000 köptes Consy upp av Hogia och fick namnet Hogia Ferry Systems (**HFS**). Hogia Ferry Systems har ett 20-tal anställda, varav nio arbetar med utveckling. HFS är fokuserat på mjukvarulösningar för sjötrafik och deras huvudprodukt är BOOKIT. BOOKIT används idag av ca 20 kunder runtom i världen för hantering av data om passagerare samt frakt inom sjötrafik (Hogia Ferry Systems, 2018).

1.2 Uppdraget

Syftet med examensarbetet var att planera och implementera funktionalitet i mjukvaran BOOKIT som skulle göra det möjligt för HFS-kunder, användarna av BOOKIT, att följa den nya dataskyddsförordningen till den grad de ville. Eftersom BOOKIT är en väl etablerad programvara var det viktigt att försöka hitta lösningar som inte påverkade arbetsflödet för användaren i onödan. Den nya dataskyddsförordningen var öppen för olika tolkningar, så det var även viktigt att inte tvinga HFS-tolkning av förordningen på deras kunder, utan istället göra implementering mycket konfigurerbar, så att kunderna kan tolka förordningen som de vill och följa den därefter.

2 Dataskydd

I detta kapitel presenteras i huvudsak Europeiska Unionens dataskyddsförordning General Data Protection Regulation. Utöver detta handlar kapitlet även om tidigare dataskyddslagstiftning i Finland.

2.1 General Data Protection Regulation

General Data Protection Regulatation eller GDPR är en ny dataskyddsförordning som godkändes av EU-parlamentet den 14 april 2016 och är lagligt bindande från och med den 25 maj 2018. Förordningen är designad och skapad för att harmonisera dataskyddslagarna inom Europa och skydda EU medborgares persondata. (Official Journal of the European Union, 2016, L119)

Förordningen gäller alla företag samt organisationer inom den Europeiska Unionen som hanterar personlig information, samt alla företag och organisationer som hanterar personlig information om EU medborgare oberoende av var företaget eller organisationen är belägen. (Official Journal of the European Union, 2016, L119, 2)

Företag som faller under den nya dataskyddsförordningen och inte följer den kan bli bestraffade med böter upp till 20 miljoner Euro eller 4% av sin årliga globala omsättning. (Official Journal of the European Union, 2016, L119, 83)

De grundläggande reglerna för den nya dataskyddsförordningen är:

Medgivande måste ges för att personlig information skall få sparas samt behandlas. Företag får inte längre gömma medgivande i långa oläsbara villkor och juristspråk. Medgivande måste bes om i klartext och måste tydligt stå ut från annan information. Medgivande måste även vara lika lätt att ta bort som att ge. (Official Journal of the European Union, 2016, L119, 6)

Rätt att bli glömd berättigar individer att kräva att behandlingen av deras persondata upphör samt kräva radering av deras persondata (Official Journal of the European Union, 2016, L119, 17).

Rätt till åtkomst berättigar individer åtkoms till sin persondata samt information om varför och vilken typ av persondata som behandlas (Official Journal of the European Union, 2016, L119, 15).

Dataportabilitet, persondata måste utan extra kostnad kunna ges åt individen i ett elektroniskt maskinläsbart format (Official Journal of the European Union, 2016, L119, 20).

Privacy by design and by default, System skall från början utvecklas med säker behandling av personlig information i fokus (Official Journal of the European Union, 2016, L119, 25).

Dataintrångsnotifiering, då ett dataintrång sker eller dataläcka upptäcks bör de berörda parterna meddelas snarast möjligt och senast inom 72 timmar (Official Journal of the European Union, 2016, L119, 33).

2.2 Tidigare dataskyddslagstiftning i Finland

Personregisterlagen 471/1987, trädde i kraft den första januari 1988. Lagens syfte var att skydda individen då personuppgifter hanteras. Lagen gällde inte hanteringen av personuppgifter i privat syfte.

(Personregisterlag, 1987, 1)

Lagen var mycket lik GDPR med paragrafer som t.ex.:

– **Angivande a personregisters ändamål**, lagringen och hanteringen av persondata kräver ändamål (Personregisterlag, 1987, 4).

– **Införande av personuppgifter i personregister**, för lagring samt hantering av en individs personuppgifter krävs medgivande ifall individen inte har anknytning till verksamheten. Personuppgifter får även sparas utan medgivande i samband med forskning, marknadsföring opinionsundersökningar m.m. (Personregisterlag, 1987, 5)

– **Förbud mot registrering av känsliga uppgifter**, känsliga uppgifter som ras, brottsliga gärningar, hälsotillstånd m.m. får inte lagras i personregister (Personregisterlag, 1987, 6).

– **Rätt till insyn**, individer har rätt att få veta vilka personuppgifter om honom har lagrats i ett personregister och varför de lagrats (Personregisterlag, 1987, 11).

– **Rättelse av fel**, individen har rätt att få felaktig, inkomplett, gammal eller onödig information raderad eller kompletterad på begäran med motivation (Personregisterlag, 1987, 15).

Personuppgiftslagen 523/1999, trädde i kraft den 1 juni 1999 och upphävde samtidigt personregisterlagen. Personuppgiftslagen var specifikare än den föregående personregisterlagen och påminner väldigt mycket om GDPR.

(Personuppgiftslag, 1999)

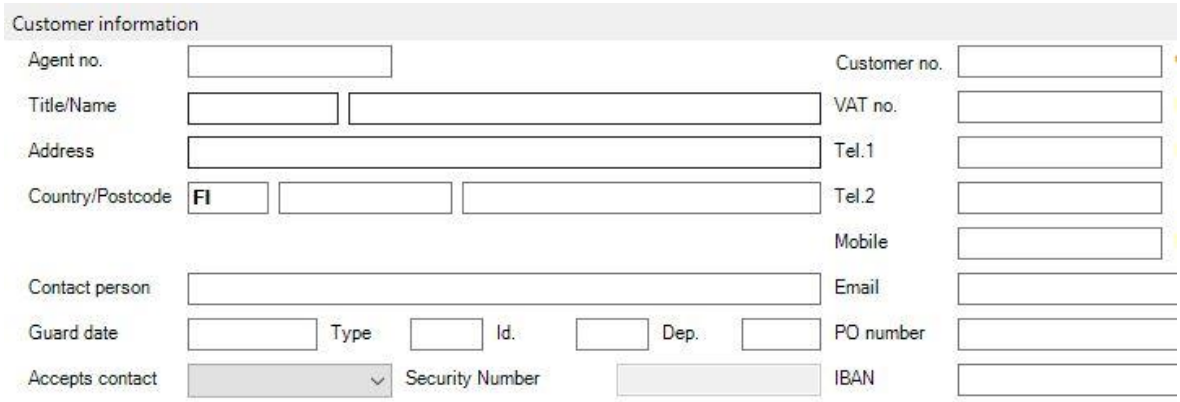
3 BOOKIT

Detta kapitel beskriver Hogia Ferry Systems bokningsprogramvara BOOKIT.

3.1 Programvaran

BOOKIT är ett mjukvarupaket utvecklat av Hogia Ferry Systems för bland annat hantering av bokningar inom sjöfart. BOOKIT är ca 30 år gammalt och består idag av ca fem miljoner rader kod. Paketet består av applikationer för reservationer, check-in, finans, rapportering, mobilapplikationer m.m. BOOKIT är idag till största delen en applikation skriven i programmeringsspråket Visual Basic.NET på .NET ramverket, med någon liten del skriven i C#. BOOKIT har även ett applikationsprogrammeringsgränssnitt eller API som kunder kan skapa t.ex. webbsidor och mobilapplikationer mot.

I BOOKIT behandlas och lagras relativt stora mängder persondata, eftersom att det oftast är lag på att veta exakt vem som är ombord på ett fartyg före det får avgå. I figur 1 kan man se ett exempel på personuppgiftsformulär i BOOKIT.



Figur 1. Personuppgiftsformulär i BOOKIT.

3.2 Infrastruktur

Nya versioner av BOOKIT släpps tre gånger per år och innehåller ny funktionalitet som arbetats på. Äldre versioner uppdateras generellt inte med nyutveckling utan endast fel korrigeras. Man stöder och uppdaterar endast versioner för ett par år bakåt men inom det så är det fritt fram för HFS-kunder att köra vilken version de vill beroende på behovet av funktionalitet. HFS-kunders servrar och miljöer hyses på olika platser som t.ex. Hogias datahall i Stenungsund eller i molntjänster som Amazons AWS och Microsofts Azure. Kunderna kör oftast med tre olika miljöer: en produktionsmiljö, en testmiljö för testning före en uppdatering tas i bruk i produktionsmiljön och en utvecklingsmiljö för testning av den senaste versionen HFS utgivit.

3.3 Användning

Användningen av BOOKIT sker förenklat på två sätt:

– En BOOKIT-användare skapar via en skrivbords- eller mobilapplikation en bokning samt matar in alla uppgifter åt en slutkund som skall resa.

– En slutkund loggar in på ett kundkonto på en webbsida som fungerar som gränssnitt mot BOOKITs API och skapar själv en bokning samt fyller i alla uppgifter.

4 Verktyg

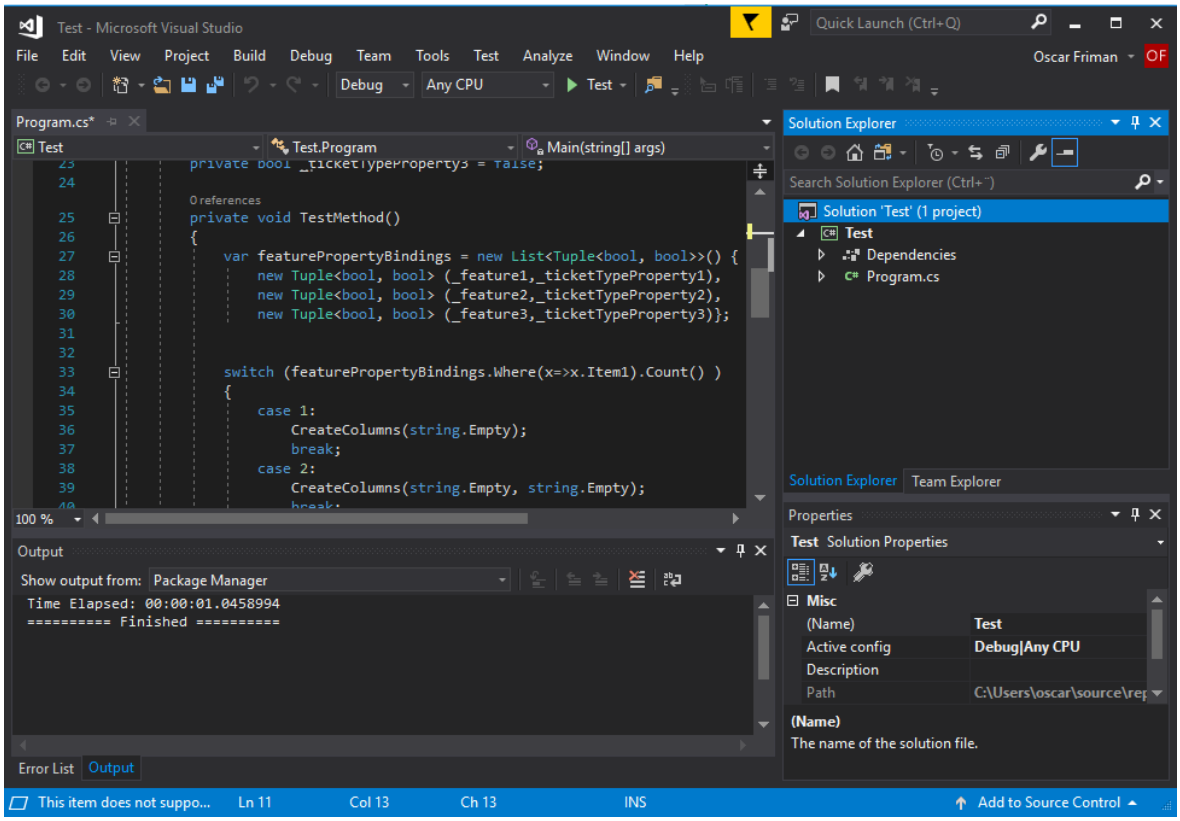
Detta kapitel beskriver verktyg och tekniker som använts för utförandet av arbetet.

4.1 Visual Studio

Microsoft Visual Studio (Se figur 2) är en integrerad utvecklingsmiljö (Integrated Development Environment, IDE) för operativsystemen Windows och Mac. Visual Studio används för att skapa t.ex. skrivbords-, mobil- och webbapplikationer. I Visual Studio ingår kompilatorstöd för flera programmeringsspråk, bland annat språken Visual Basic och C#.

(Microsoft, 2018)

Största delen av all utveckling av BOOKIT sker med hjälp av Microsofts Visual studio.



Figur 2. Microsoft Visual Studios användargränssnitt.

4.2 .NET Framework

.NET Framework är ett mjukvaruramverk utvecklat av Microsoft. .NET Framework består av klassbibliotek för ett brett stöd av funktionalitet och industristandarder som t.ex. stöd för olika programspråk att kunna fungera med varandra, minneshantering och felhantering. Klassbiblioteket innehåller även funktionalitet för bland annat:

– Användargränssnitt

– Databashantering

– Numeriska algoritmer

– Kryptografi

– Nätverk

(Microsoft, 2018)

4.3 Visual Basic .NET

Visual Basic .NET (Se kodexempel 1) introducerades år 2002 av Microsoft och är uppföljaren till programspråket Visual Basic. Visual Basic .NET är ett objektorienterat programspråk skrivet för användning av .NET Framework likt Microsofts programspråk C# (Se kodexempel 2). Största skillnaden mellan Visual Basic .NET och C# är syntaxen, men i övrigt är de så lika så man rentav kan konvertera från det ena språket till det andra med endast en simpel översättare (Se figur 3).

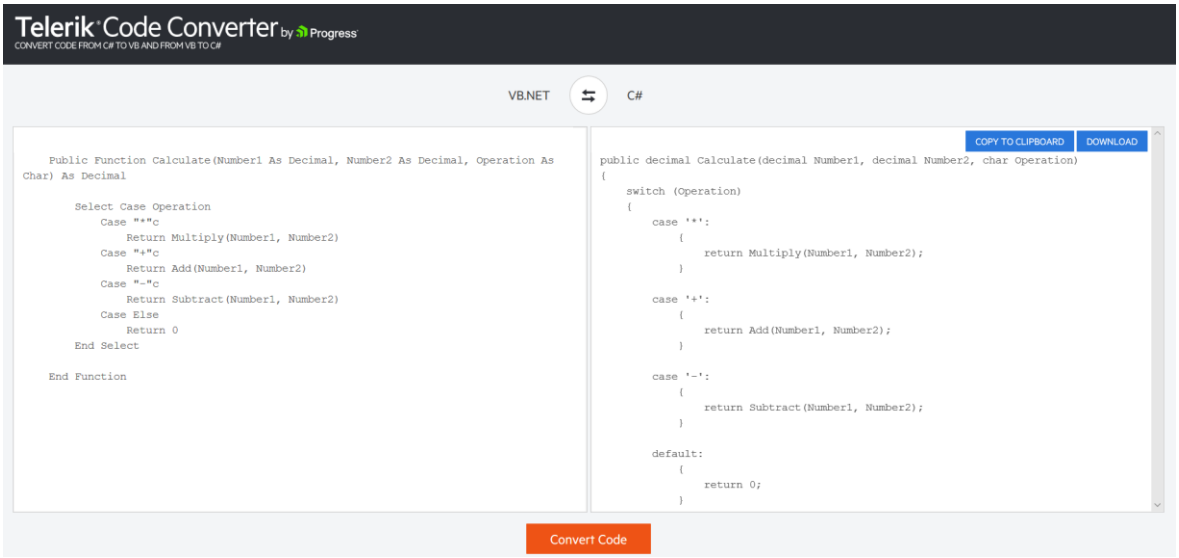
(Visual Basic .NET, 2018)

Kodexempel 1. Visual Basic.NET.

```
Public Function Calculate(Number1 As Decimal, Number2 As Decimal, Operation As Char) As Decimal

    Select Case Operation
        Case "*"c
            Return Multiply(Number1, Number2)
        Case "+"c
            Return Add(Number1, Number2)
        Case "-"c
            Return Subtract(Number1, Number2)
        Case Else
            Return 0
    End Select

End Function

1 reference
Private Function Multiply(factor1 As Decimal, factor2 As Decimal) As Decimal
    Dim result As Decimal = 0

    result = factor1 * factor2

    Return result
End Function

1 reference
Private Function Add(term1 As Decimal, term2 As Decimal) As Decimal
    Dim result As Decimal = 0

    result = term1 + term2

    Return result
End Function

1 reference
Private Function Subtract(term1 As Decimal, term2 As Decimal) As Decimal
    Dim result As Decimal = 0

    result = term1 - term2

    Return result
End Function
```

Kodexempel 2. C#.

```
public decimal Calculate(decimal Number1, decimal Number2, char Operation)
{
    switch (Operation)
    {
        case '*':
            {
                return Multiply(Number1, Number2);
            }

        case '+':
            {
                return Add(Number1, Number2);
            }

        case '-':
            {
                return Subtract(Number1, Number2);
            }

        default:
            {
                return 0;
            }
    }
}

1 reference
private decimal Multiply(decimal factor1, decimal factor2)
{
    decimal result = 0;

    result = factor1 * factor2;

    return result;
}

1 reference
private decimal Add(decimal term1, decimal term2)
{
    decimal result = 0;

    result = term1 + term2;

    return result;
}

1 reference
private decimal Subtract(decimal term1, decimal term2)
{
    decimal result = 0;

    result = term1 - term2;

    return result;
}
```

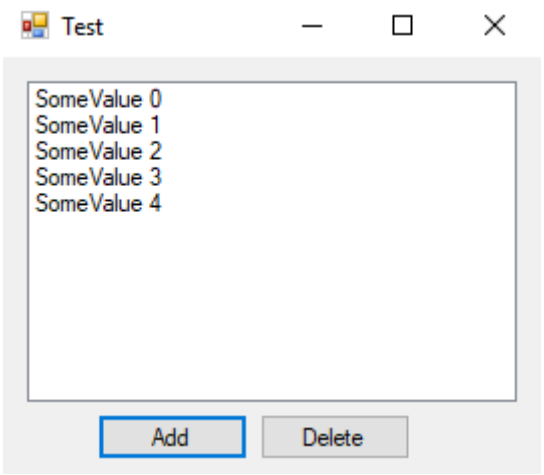


Figur 3. Exempel på webbaserad kodkonverterare.

4.4 Windows Forms

Windows Forms är ett klassbibliotek för grafiska användargränssnitt för skrivbordsapplikationer och är en del av .NET Framework. En Windows Forms applikation (Se figur 4) är händelsedriven (event-driven), vilket betyder att programmet ligger och väntar på att en händelse sker, t.ex. en knapp trycks in eller en ruta får fokus, innan koden som är kopplad till händelsen exekveras (se kodexempel 3).

(Windows Forms, 2018)



Figur 4. Exempel på användargränssnitt i Windows Forms.

Kodexempel 3. Kodexekvering vid knapptryckshändelse.

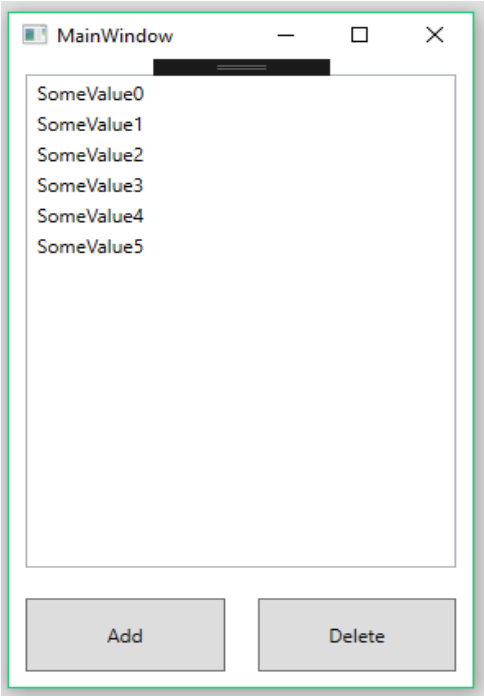
```
1 reference
private void btnAdd_Click(object sender, EventArgs e)
{
    lstValues.Items.Add("SomeValue " + lstValues.Items.Count);
}
```

4.5 Windows Presentation Foundation

Windows Presentation Foundation (WPF) är ett flexibelt verktyg för att skapa grafiska användargränssnitt för skrivbordsapplikationer (Se figur 5) och är en del av .NET Framework. WPF använder sig av XAML, ett XML baserat markeringsspråk, för deklareringen av det grafiska användargränssnittet (Se kodexempel 4).

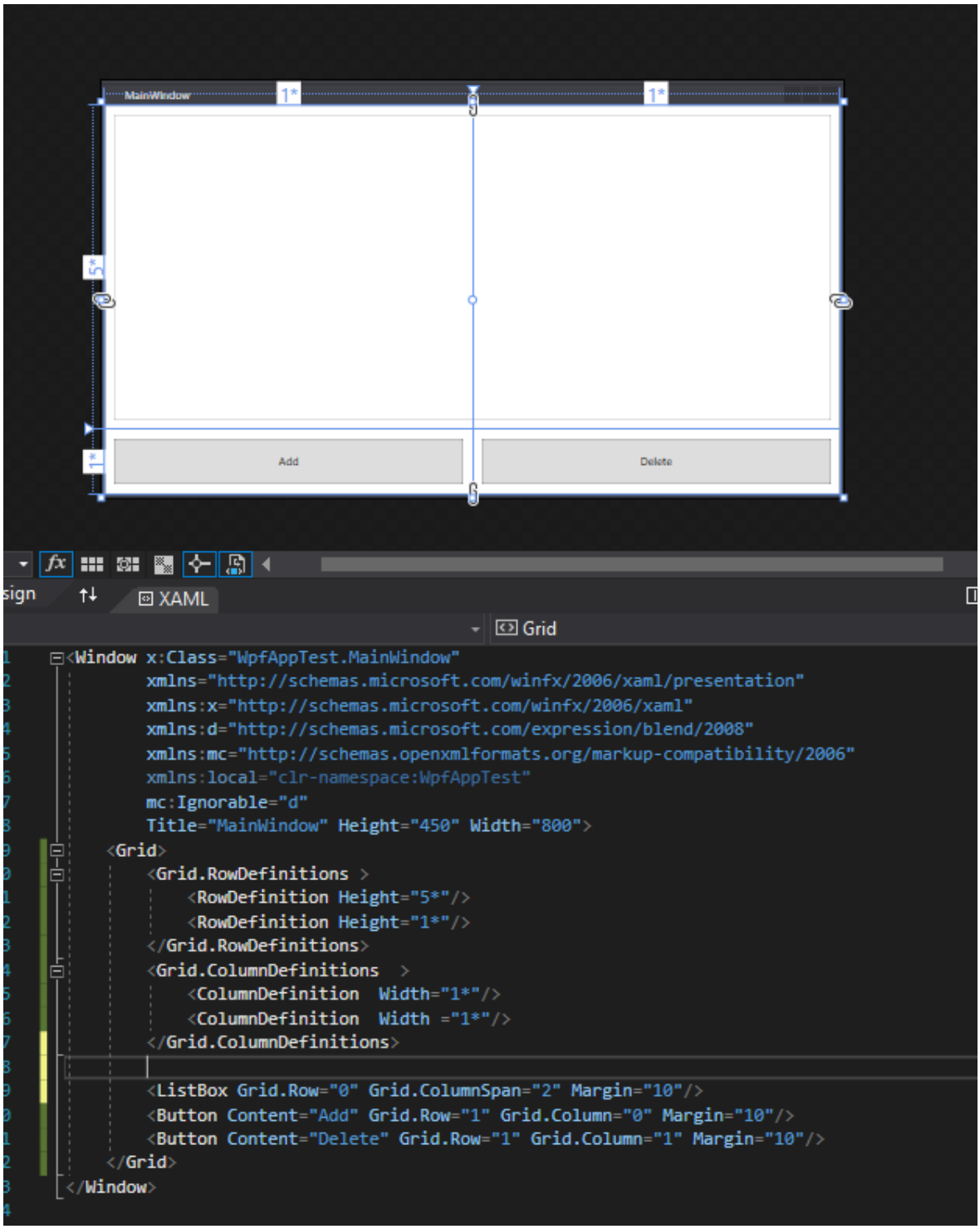
(Microsoft, 2016)

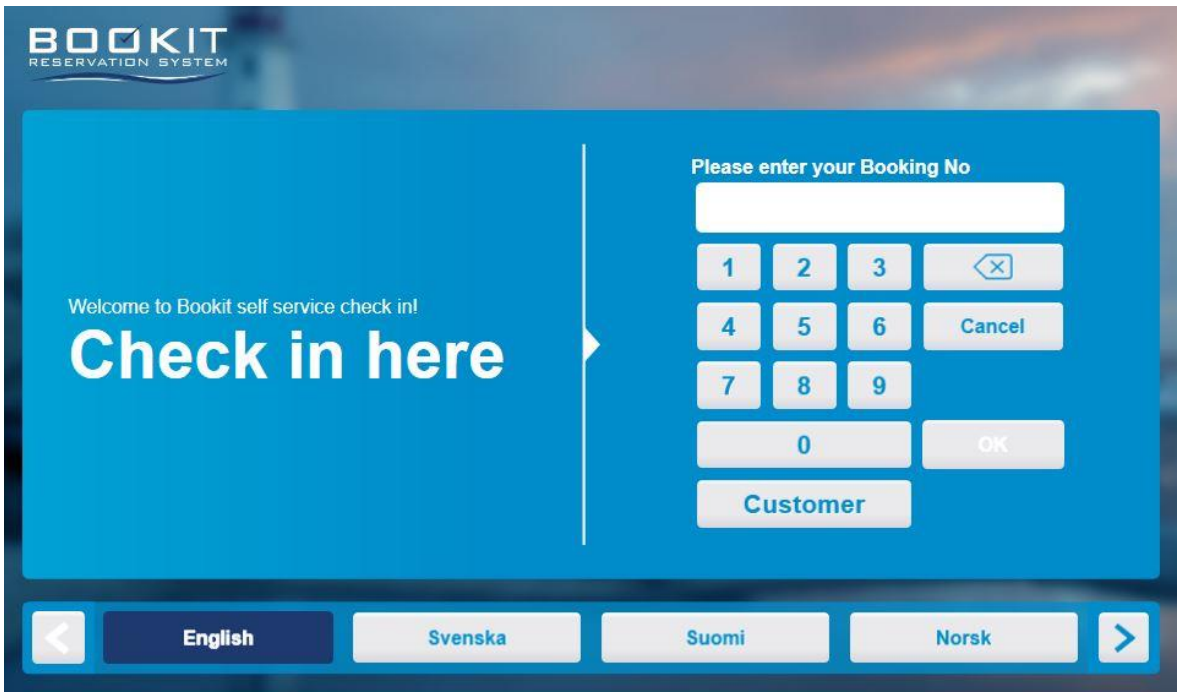
I BOOKIT används WPF för t.ex. incheckningsmjukavara som slutkunderna själva kan använda för att checka in. Figur 6 visar ett exempel på incheckningsmjukvara i WPF.



Figur 5. Användargränssnitt byggt med WPF.

Kodexempel 4. Deklarering av användargränssnitt i XAML.





Figur 6. Exempel på incheckningsapplikation i WPF.

4.6 SQL

Structured Query Language eller SQL är ett kommandospråk för definiering samt manipulering av relationsdatabaser. Med SQL kan man bland annat lägga till, uppdatera, radera och söka information i en databas.

De vanligaste SQL-kommandona:

– Create table, för skapande av nya tabeller (Se kodexempel 5).

– Alter table, för modifiering av existerande tabeller.

– Drop table, för radering av tabeller.

– Insert into, för inmatning av nya rader i en tabell (Se kodexempel 6).

– Update, för modifiering av existerande rader i en tabell.

– Delete, för radering av rader i en tabell.

– Select, för hämtning av rader från en tabell (Se kodexempel 7).

(W3schools,2018)

Kodexempel 5. Skapande av ny tabell.

```
Create table Customers (
                FirstName varchar(255),
                LastName varchar(255),
                Email varchar(255),
                Consent bit );
```

Kodexempel 6. Inlägg av ny rad i existerande tabell.

```
Insert into Customers (FirstName,LastName,Email,Consent)
values('Test','Person','TestPerson@Test.Com',1);
```

Kodexempel 7. Hämtning av kunder som har ett "Consent" värde olikt 0.

```
select * from Customers where Consent <> 0;
```

16 [386] INS

Log | 1: Customers [1] ×

FirstName	LastName	Email	Consent
Test	Person	TestPerson@Test.Com	1

5 Utförande

I detta kapitel beskrivs utförandet arbetet genom informationssökning, planering av ny funktionalitet samt implementering av den nya funktionaliteten.

5.1 Informationssökning

Arbetet inleddes med att söka reda på information gällande den nya förordningen. Detta innebar att läsa och tolka förordningen själv samt att ta reda på hur andra företag och personer tolkat den. Sedan identifierades alla personuppgifter som hanteras i HFS-

programvara samt deras användningsområde för att få en uppfattning om hur deras personuppgiftshantering såg ut. Med detta kunde man få fram de huvudsakliga punkterna som påverkar produkten. Dessa punkter var:

– Medgivande för lagring samt behandling av persondata.

– Pseudonymisering av persondata.

– Radering av persondata.

– Rapporter på användningen av persondata.

– Loggning av behandlingen av persondata.

– Förhindrande av databehandling.

5.2 Planering

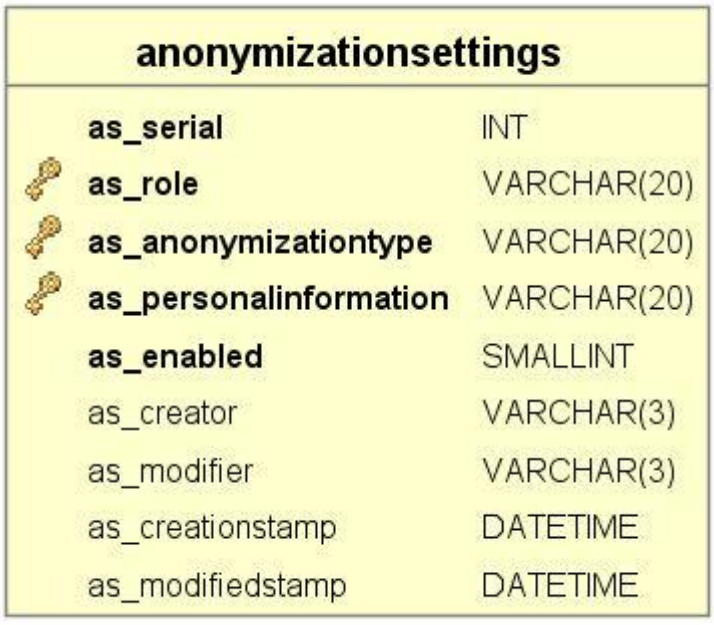
Planeringen gick ut på att ta reda på vilken funktionalitet som behövde utvecklas för att kunna följa förordningen, hur funktionaliteten skulle implementeras samt hur användargränssnittet skulle se ut. Viktigt att ta i beaktande under planeringen var att existerande funktionalitet inte skulle förändras och att användarflödet inte skulle påverkas för mycket. Planeringen gick även ut på att framställa ett dokument som innehöll alla förändringar som skulle utföras och en tidsuppskattning för hur länge projektet skulle ta. Planen skulle sedan få ett godkännande av produktägare, utvecklingschef samt systemarkitekt. Då allt detta var klart kunde arbetet påbörjas.

5.3 Implementering

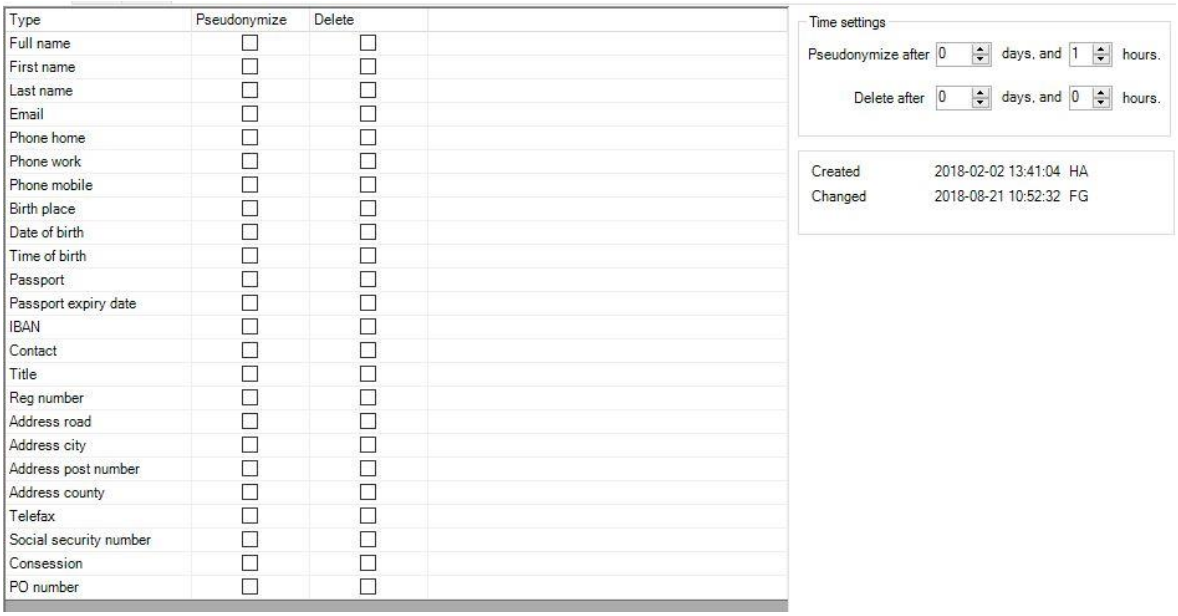
5.3.1 Register och tabeller

Nya databastabeller skapades för att hålla reda på alla inställningar som lades till för GDPR. Tabellerna designades på ett vis som tillåter att nya inställningar läggs till utan att behöva göra ändringar till tabellstrukturer. Inställningarna sparas som nyckel-värde-par (key-value).

Nycklarna är hårdkodade strängar i kodbasen och identifierar en inställning, medan värdena bestämmer om den inställningen är på eller av. På så vis har HFS möjlighet att lägga till inställningar även för kunder som använder äldre versioner vid behov. Figur 7 visar kolumnerna för en av de nya tabellerna som lagts till. I tabellen sparas information om hur personuppgifter skall anonymiseras i BOOKIT. Registret är BOOKIT inställningsgränssnitt, registret används för konfigurering av all BOOKIT funktionalitet. Här definieras t.ex. alla fartyg, rutter, resurser och priser. Nya inställningsmöjligheter lades till för de nya GDPR inställningarna. Figur 8 visar inställningarna för pseudonymisering samt radering av personuppgifter. I vänstra kolumnen listas alla uppgifter som BOOKIT hanterar och som kan tillhöra en person, alla behöver nödvändigtvis inte vara personuppgifter. I den mittersta kolumnen finns ett kryssalternativ för om uppgiften skall hanteras som en personuppgift och pseduonymiseras. I högra kolumnen finns ett kryssalternativ för om uppgiften skall raderas vid radering av personuppgifter. Längst till höger finns tidsinställningar för pseudonymiseringen samt raderingen. Dessa säger hur länge efter skapandet av informationen skall informationen börja pseudonymiseras eller raderas.



Figur 7. Anonymiseringsinställningstabell.



Figur 8. Inställningar för pseudonymisering och radering av personuppgifter.

5.3.2 Pseudonymisering

Eftersom registreringen och hanteringen av personuppgifter i BOOKIT ofta sker genom en tredje part och inte personuppgiftsägaren själv, ville HFS ge möjligheten att kunna förhindra användaren av BOOKIT att se persondata som inte är nödvändig, utan att behöva ändra behandlingen av persondata i fråga. Då största delen av den personliga informationen som hanteras i BOOKIT, manipuleras via Windows Forms vanliga TextBox-klass, kunde pseudonymisering i stort sett lösas genom att skapa en ny klass för att ersätta den existerande TextBox-klassen, den nya klassen ärver sina egenskaper från TextBox-klassen och fick en ny egenskap som heter PlainText. PlainText-egenskapen används för att lagra data i ren text medan basklassens Text-egenskap är vad som visas i användargränssnittet. För att kunna maskera data korrekt måste informationens typ, ålder samt roll (t.ex. slutkund eller användare), tas i beaktande. För detta skapades en metod i den nya klassen som tar dessa parametrar som input och sedan bestämmer om data som visas ut åt användaren behöver pseudonymiseras eller ej. Kodexempel 8 visar hur metoden Initialize tar metadata om persondata i fråga som inputparametrar. Ett verktyg som heter PIMaskingTool kan sedan jämföra parametrarna mot pseudonymiseringsinställningarna och bestämma om informationen skall maskeras samt hur den skall maskeras.

Kodexempel 8. Metoder för hantering av persondata.

```
99+ references
Public Sub Initialize(type As String, creationDate As Date, role As String, restricted As Boolean)
    _res = UnityBootstrapper.Current.Container.Resolve(Of PrivacySettingsCacheTool)
    _type = type
    _creationdate = creationDate
    _role = role
    _markedForPseudonymization = _res.MarkedForPseudonymization(_type, _role)
    _timeBeforePseudonymization = _res.HoursBeforePseudonymization(_role)
    _restricted = restricted
    _applicationId = ApplicationIdEnum.GenericApplication
    Dim session As BookitSession = GetCurrentSession(Me)
    If session IsNot Nothing Then
        _applicationId = session.ApplicationId
    End If

End Sub
```

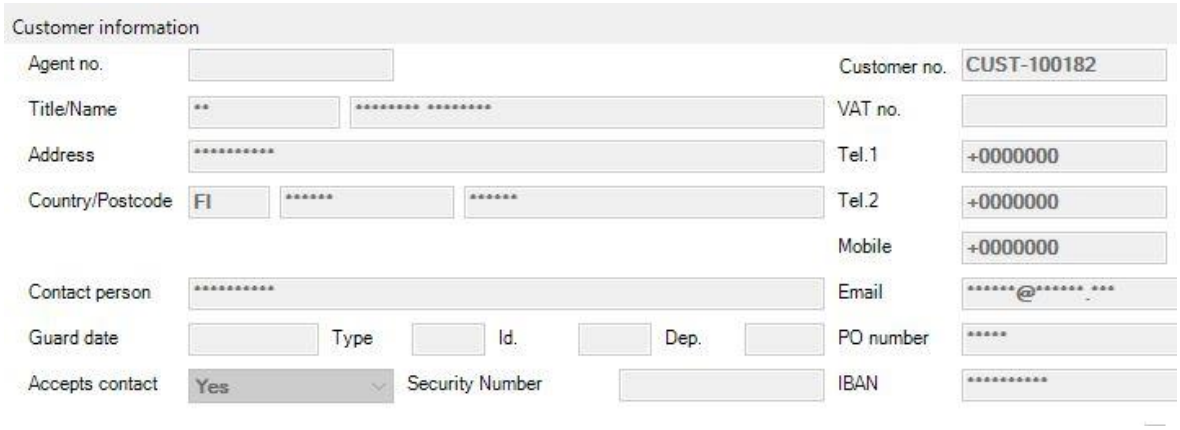
```
99+ references
Public Property PlainText As String
    Get
        Return _plaintext
    End Get
    Set
        If Value Is Nothing Then
            Value = String.Empty
        End If
        _plaintext = Value

        If Value <> String.Empty AndAlso Not PIMaskingTool.UsePlainText(_creationdate, _type, _timeBeforePseudonymization, _markedForPseudonymization, _applicationId) Then
            Me.Text = PIMaskingTool.MaskValue(_type, _creationdate, _role, Value, _applicationId)
            Me.Enabled = False
        Else
            If _restricted Then
                Me.Enabled = False
            End If
            Text = Value
        End If
    End Set
End Property
```

5.3.3 Medgivande

Möjlighet att ge medgivande samt kräva medgivande vid lagring av personlig information lades till på alla de platser där personlig information manipuleras, som t.ex. vid skapande av bokningar, kundkonton användare osv. både i skrivbordsapplikationen och via applikationsprogrammeringsgränssnittet. Även de webbsidor HFS skapat åt sina kunder uppdaterades med funktionalitet för medgivande. Man valde att även alltid spara vilken användare som fyllt i medgivande samt tids- och datumstämpel.

5.3.4 Radering

På grund av att lagen i olika länder till viss del bestämmer vilken information som måste sparas, samt att BOOKIT inte skulle kunna hantera radering av bokningar, kundkonton, användarkonton och så vidare, utan stora förändringar av funktionaliteten, valde man att sköta radering av personlig information på fältnivå istället. Då en kund t.ex. väljer att radera sitt konto, kommer istället alla de fält som enligt inställningarna räknas som personlig information, att fyllas med icke-unik information (Se figur 9). Denna data är samma för alla fält och alla bokningar, vilket gör att kontot/bokningen inte längre är kopplingsbar till någon specifik person men kan fortfarande hanteras av BOOKIT för t.ex. beräkning av statistik.



Figur 9. Exempel på en bokning vars personliga information blivit raderad.

Radering av personlig information utvecklades att ske på två olika sätt:

– Manuell radering:

- En användare går in på en bokning/kundkonto/användarkonto o.s.v. och väljer manuellt att den personliga informationen på detta konto eller bokning skall raderas.

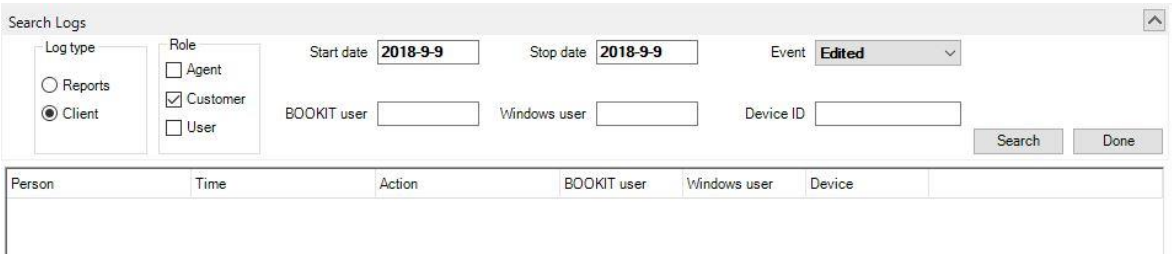
– Automatisk radering

- Medgivande på t.ex. ett kundkonto upphävs.
- En insticksmodul i det egenutvecklade Windows-serviceprogrammet för körning av schemalagda processer raderar alla personliga uppgifter som överensstämmer med inställningarna för när detta skall raderas.

5.3.5 Loggning

I BOOKIT loggas alltid BOOKIT-användaren och tids- samt datumstämpel för senaste ändringen av ett objekt. Detta valdes att utökas till en ny loggtabell. I loggtabellen sparas tids- och datumstämpeln för händelsen, BOOKIT-användaren, Windows-användaren, maskin-ID, ID för individen som påverkats av händelsen, samt händelsetyp. Händelsestypen kan vara t.ex. att ett kundkonto har öppnats, modifierats, raderats, eller medgivandet

upphävts. Då även rapporter kan innehålla personlig information valde HFS även att logga rapportutskrifter och vilka parametrar en rapport hämtats ut med. Även möjligheten att söka upp händelse- och rapportutskriftsloggar skapades (Se figur 10).



Figur 10. Loggregister

5.3.6 Rapporter

För att kunna förse personer med all information som finns lagrat om dem i systemet skapades två nya rapporter.

– En rapport för att få ut all personlig information, denna rapport kan tas ut i PDF- samt Excel-format. Rapporten listar varje fält från BOOKIT som innehåller personlig information samt en beskrivning på vad informationen används till. Rapporten kan begäras på flera språk och beskrivningarna kan läggas till i registret för varje språk.

– En rapport för att få ut alla tjänster som är kopplade till personen i fråga. Detta är t.ex. alla bokningar, biljetter, kundkort och så vidare, som på något vis är kopplat till personen. Denna rapport kan endast fås i PDF-format.

5.3.7 Förhindrande av databehandling

Funktionalitet för att förhindra behandlingen av en individs personliga information utvecklades. Funktionaliteten gör det möjligt att ”låsa” t.ex. ett kundkonto, och på så vis förhindra att personlig information läggs till, redigeras eller raderas. Konton som blivit låsta kan inte heller användas för bokningar. Konton som låsts kan även låsas upp och användas som vanligt igen.

6 Resultat

Resultatet av arbetet blev flera tillägg till den existerande programvaran, som låter HFS-kunder följa den nya dataskyddsförordningen till den grad de vill, med möjlighet att utöka inställningar och regler utan att göra stora förändringar i kodbasen. På så vis kan HFS fortsätta stöda sina kunders behov, även om behoven varierar rätt så mycket.

7 Diskussion

Jag valde att utföra detta projekt samt att skriva om det som examensarbete, eftersom det lät som ett mycket intressant projekt även om jag visste att det skulle vara utmanande. Förvånansvärt mycket tid av projektet gick åt till att tolka förordningen och hitta lösningar som skulle passa HFS. Många idéer bollades och förkastades förrän vi kom fram till lösningen vi har idag.

Jag är nöjd med slutresultatet även om det finns en del förbättrings- och vidareutvecklingsmöjligheter. Jag har i efterhand till exempel implementerat regler på användar-nivå för ”ignorering” av pseudonymiseringsreglerna på vissa ställen i BOOKIT. Dessutom har jag implementerat möjligheten till pseudonymisering baserat på ett bokningsresedatum istället för bokningens skapningsdatum.

Arbetet var mycket lärorikt gällande hantering av persondata som man kanske inte alltid som utvecklare tänker på. Även hanteringen av ett så stort projekt som detta tror jag kan vara en erfarenhet som jag kommer att ha nytta av i framtiden. Arbetet var även mycket givande för kännedom av BOOKIT-kodbasen, vilket jag har nytta av i mitt dagliga arbete.

BOOKIT versionen med ändringarna som jag gjort under detta arbete har i skrivande stund tagits i bruk av ett antal kunder och funktionaliteten används i varierande grad. Jag hoppas att då flera kunder tar funktionaliteten i bruk så får vi även förbättringsförslag och vidareutvecklingsmöjligheter för funktionaliteten.

Källförteckning

Hogia, 2018, Hogia Group. [Online]
https://www.hogia.com/hogia-group/about [Hämtat 05.09.2018]

Hogia Ferry Systems, 2018, About Us [Online]
https://bookit.hogia.fi/about-us/ [Hämtat 05.09.2018]

Microsoft, 2018, About Visual Studio 2017 [Online]
https://docs.microsoft.com/en-us/visualstudio/ide/visual-studio-ide?view=vs-2017 [Hämtat 08.09.2018]

Microsoft, 2018, .NET Framework Guide [Online]
https://docs.microsoft.com/dotnet/framework/ [Hämtat 08.09.2018]

Microsoft, 2018, WPF overview [Online]
https://docs.microsoft.com/en-us/visualstudio/designers/introduction-to-wpf?view=vs-2017 [Hämtat 23.09.2018]

Personregisterlag 30.04. 1987/471 [Online]
www.finlex.fi [Hämtat: 05.09.2018]

Personuppgiftslag 22.04.1999/523 [Online]
www.finlex.fi [Hämtat: 05.09.2018]

THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) 24.05.2016/119 [Online]
www.eur-lex.europa.eu [Hämtat: 05.09.2018]

Visual Basic .NET, 2018 [Online]
https://en.wikipedia.org/wiki/Visual_Basic_.NET [Hämtat 09.09.2018]

Windows Forms, 2018 [Online]
https://en.wikipedia.org/wiki/Windows_Forms [Hämtat 09.09.2018]

W3Schools, 2018, introduction to SQL [Online]
https://www.w3schools.com/sql/sql_intro.asp [Hämtat 08.09.2018]

Figurförteckning

Kodexempelsförteckning