

Lazarus

- en digital gravbok

Erik Björkman



2023:17

Datum för godkännande: 25.05.2023
Handledare: Kim Gylling

EXAMENSARBETE

Högskolan på Åland

Utbildningsprogram:	Informationsteknik
Författare:	Erik Björkman
Arbetets namn:	Lazarus - en digital gravbok
Handledare:	Kim Gylling
Uppdragsgivare:	Norra Ålands församling

Abstrakt

Det här projektet är ett samarbete med Norra Ålands församling där jag utvecklar en applikation som de kan använda för att hantera sina uppgifter om begravda människor. Målet med applikationen är att det ska finnas en gemensam plattform för församlingens fyra begravningsplatser, där samma struktur används och datan är lagrad på samma plats. Resultatet blev en webbapplikation som kommer att köras lokalt på en server, där användaren kan lägga till och ta bort information om gravar.

Nyckelord (sökord)

Begravningsplats, Grav, JavaScript, Node.js

Högskolans serienummer:	ISSN:	Språk:	Sidantal:
2023:17	1458-1531	Svenska	25

Inlämningsdatum:	Presentationsdatum:	Datum för godkännande:
03.05.2023	16.05.2023	25.05.2023

DEGREE THESIS

Åland University of Applied Sciences

Degree Programme:	Bachelor of Information Technology
Author:	Erik Björkman
Title:	Lazarus - a digitalization of burial records
Academic Supervisor:	Kim Gylling
Commissioned by:	Norra Ålands församling

Abstract

This project is a cooperation with Norra Ålands församling where I develop an application that is used for managing data of buried people. The objective is to have a common platform for all four cemeteries within the congregation, where they can use the same structure and data storage. The result is a web application that will run on a local server, where the user can add and delete information about graves.

Keywords

Cemetery, Grave, JavaScript, Node.js

Serial number:	ISSN:	Language:	Number of pages:
2023:17	1458-1531	Swedish	25

Handed in:	Date of presentation:	Approved:
03.05.2023	16.05.2023	25.05.2023

INNEHÅLLSFÖRTECKNING

1. INLEDNING	5
1.1 Bakgrund	5
1.2 Syfte och målsättning	6
1.3 Metod	6
1.4 Avgränsningar	7
2. TEORI	8
2.1 Begravningsplats	8
2.2 Utvecklingsverktyg	9
2.2.1 JavaScript	9
2.2.2 Node.js	9
2.2.3 MySQL	10
3. KRAVSPECIFIKATION	10
3.1 Skall-krav	10
3.2 Bör-krav	11
3.3 Bra-att-ha-krav	11
4. IMPLEMENTATION	11
4.1 Struktur	11
4.2 Grafisk representation av gravgården	12
4.3 Databasen	14
4.3.1 Vilken data sparas?	14
4.3.2 Applikationens kommunikation med databasen	16
4.4 Gravdata	19
4.5 Sökfunktion	21
5. SLUTSATSER	21
5.1 Resultat	21
5.2 Diskussion	22
KÄLLFÖRTECKNING	24

1.2 Syfte och målsättning

Uppdragsgivare för projektet är Norra Ålands församling som grundades i januari 2022, då tidigare församlingarna Finström-Geta och Sund-Vårdö slogs ihop. Församlingen har för tillfället 18 anställda och antalet medlemmar var uträknat till 3301 i mars 2023 (*Medlemsstatistik*, 2023).

Det nuvarande registret finns uppdelat i olika Excel-dokument. Excel har ett stort utbud av funktionalitet vilket kan vara till nytta, men samtidigt kan ovana användare bli överväldigade av alla menyer och inställningar som finns. Fördelen med Excel är att det är flexibelt. Då viss information är så gammal kan uppgifter saknas eller vara oregelbundna.

Lazarus ska bara innehålla de funktioner som behövs för att göra det lätt att använda och minimera risken att göra fel.

De tidigare församlingarna hade sina egna register och system som hade hanterats av olika personer, vilket gjorde att det såg olika ut. Ett av målen för den nya applikationen är att kunna ha all information på samma ställe och strukturera uppgifterna på samma sätt.

De fyra olika kyrkogårdarna har sina egna Excel-filer och papperskartor där filerna finns i olika mappar på deras server och kartorna bara finns i pappersformat som hänger på väggarna i församlingshemmen. På deras server finns fortfarande gamla kopior från de tidigare församlingarnas register. När applikationen är färdigt kommer alla kartor och uppgifter att finnas på ett och samma ställe. Då kommer det vara lätt att komma åt all data utan att behöva söka bland de olika filerna. Eftersom jag själv har tagit initiativet att skapa en applikation som kan lösa de problem som finns med deras nuvarande register, så blir en av utmaningarna att övertyga uppdragsgivaren vilka möjligheter som finns så att de i framtiden kan ha användning för det.

1.3 Metod

Eftersom uppdragsgivaren inte är insatt inom informationsteknik och inte känner till alla möjligheter med en applikation, så måste jag själv vara ledare för dess utveckling. Det inkluderar bland annat utseende, design, struktur och funktionalitet. Allt måste dock vara baserat på uppdragsgivarens behov och idéer. Det har inte skett officiella möten om projektet utan det har uppkommit mindre diskussioner då det har behövts.

De gravkartor som finns är ritade för länge sedan och är inte helt pålitliga. Förr i tiden sköttes alla gravgårdar för hand, men nu måste de vara planerade så att man ska kunna köra med maskiner för att till exempel gräva gravar eller flytta på gravstenar. Även de områden som är utritade på kartorna är inte alltid logiska så därför kom jag överens med uppdragsgivaren att jag kan rita om kartorna eller ändra på områdena ifall jag hittar något bättre system. Därför kommer jag också att behöva gå runt på de olika gravgårdarna för att kunna förbättra kartorna så att de blir så lika verkligheten som möjligt.

Begravningsplatsernas prioritetsordning blev mestadels att bero på tillgängligheten.

Finströms begravningsplats var fysiskt närmast och har väldefinierade delområden vilket gjorde det lättast att börja där, trots att det var det ställe som var i minst behov av ett nytt system. Alla fyra ställen kommer dock att ingå i applikationen innan den kan användas.

1.4 Avgränsningar

I början av projektet övervägdes det mellan om det nya gravregistret skulle vara en webbapplikation som ligger på nätet eller en desktopapplikation. Fördelen med en webbapplikation är att de anställda då skulle kunna komma åt registret hemifrån eller via en portabel enhet på kyrkogården. Applikationen skulle kunna göras publik så att vem som helst som är intresserad kan se vilka som är begravda utan att de behöver besöka någon kyrkogård. Det skulle dock kräva att systemet har olika rättigheter för användare och personal. Då måste ett säkert inloggningssystem implementeras. Ursprungligen är applikationen menad att bytas ut mot deras nuvarande register som enbart är till för församlingens bruk. Allmänna personer som vill ha reda på information får istället kontakta församlingen. Slutligen bestämdes det att det nya gravregistret ska vara uppbyggt som en webbapplikation då det i framtiden skulle ge rum för att ha det ute på nätet. Det kommer dock att köras på en lokal server så att man bara kommer åt applikationen via församlingens lokala nätverk. När projektet är klart kommer åtminstone den funktionalitet som finns i det nuvarande registret också att finnas i den nya applikationen. Därefter kommer vidareutveckling att ske tills alla nödvändiga krav och behov har uppnåtts.

Användaren kommer inte att kunna ändra på det grafiska innehållet på gravkartan då det skulle finnas risk för att det blir oredigt. Så som det ser ut idag kommer det finnas tillräckligt med gravplatser på kyrkogårdarna att de inte kommer behöva byggas ut eller läggas till fler platser än de som finns utsatta. Bara en bråkdel av alla begravningar sker i en ny grav då de flesta redan har familjegravar som de kan använda.

2. TEORI

2.1 Begravningsplats

Kyrkolagen innehåller information om vad församlingen ska bistå med under en begravning och vilka skyldigheter de har, men det finns inga direktiv för hur en begravningsplats skall skötas. Det är upp till församlingen att själv komma fram till det system de skall använda. Därför är det viktigt att känna till de regler och tillvägagångssätt som Norra Ålands församling har, då deras system är grunden för hur applikationen är uppbyggd. Källan till informationen är från de som arbetar i församlingen.

En gravplats är ungefär 1 meter bred, vilket betyder att det i teorin går lika många gravplatser på ett område som det är brett. I verkligheten blir det dock ofta färre då det kan finnas hinder, till exempel träd och stora stenar. Gravstenssättningen sköts inte av församlingen så de blir ibland placerade lite snett, vilket gör att en gravplats tar mer utrymme än det teoretiska. Rätten till en gravplats varar i 25 år. Det är ungefär den tiden som det tar för en kista att förmultna innan en ny person kan begravas på samma plats. Församlingen har den standarden att lägga kistor på ett djup av 180 cm vilket stämmer överens med §6 i Kungl. Maj:ts begravningskungörelse där en kista skall ligga på sådant djup att det finns åtminstone en meter jord ovanför (*Kungliga Maj:ts Begravningskungörelse* 1963). Om två personer är planerade att begravas i samma grav inom 25 år kan det lösas med en djup grav, som är 240 centimeter djup. Denna metod rekommenderas inte av församlingen då det på många ställen på begravningsplatserna inte går att gräva så djupt eftersom det kommer berg emot. Ett annat problem med djupa gravar är att det blir en betydligt större mängd jord som skall tas upp. Om det är trångt mellan gravstenarna finns det inte tillräckligt med plats för jorden. Urngravar har

ett mått på 60 x 60 centimeter och ett djup på 80 centimeter. Det är beräknat att få plats ungefär fyra urngravar på samma yta som en kistgrav och de kan grävas ovanpå en nyligen begravd kista. En gravplats kan beställas som enkel- eller dubbelgrav. Dubbelgravar utlöses då den anhöriga vill reservera en plats åt sig själv eller någon annan släkting bredvid den som skall begravas. Det här är det enda sättet församlingen tillåter reservation av gravplatser då åtminstone en av personerna i en dubbelgrav måste vara avliden.

Gravar som inte sköts kan tas bort ifall gravstenen är instabil och det finns en risk att den faller och kan skada någon. Detta är dock en lång process då eventuella anhöriga måste kontaktas. Det är viktigt för kyrkogården att få en ny tillgänglig begravningsplats som kan användas i framtiden.

2.2 Utvecklingsverktyg

2.2.1 JavaScript

JavaScript är ett skriptspråk som vanligen används för webbaserad programmering där koden körs på klienten. Språket är en del av en treenighet som även består av HTML och CSS, som tillsammans bygger upp de flesta webbapplikationer. HTML beskriver strukturen på elementen, till exempel paragraf, tabell eller bild. CSS-kod används för att elementen ska få någon design, till exempel färg, form eller position. JavaScript är den del av applikationen som reagerar på händelser för att därefter ändra på innehållet i HTML- eller CSS-koden (*What Is JavaScript? - Learn Web Development | MDN, 2023*).

JavaScript lämpar sig bra för det här projektet då majoriteten av koden finns på klient-sidan vilket gör att applikationen får en kort exekveringstid.

2.2.2 Node.js

Node.js är en open source-exekveringsmiljö skapad av Ryan Dahl år 2009, som baserar sig på JavaScript och kan användas för att köra en webbapplikation på en server. Eftersom Node.js använder sig av JavaScript så kan programmeraren skriva både frontend och backend i samma språk. Funktionerna i Node.js körs asynkront, vilket betyder att de körs parallellt och inte väntar in på returnerad data förrän nästa funktion körs. På det viset förhindrar man att funktionerna blockerar eller fördröjer varandra (*About Node.js, n.d.*). Node.js använder sig av lite RAM- och CPU-resurser jämfört med till exempel PHP, vilket gör att det passar bra för ett mindre program (Almeida, 2022).

2.2.3 MySQL

MySQL är ett system för hantering av relationsdatabaser. En sådan databas delar upp hela databasen i mindre tabeller där de olika tabellerna ofta är anknutna till varandra på olika sätt. På det viset kan man undvika dataredundans då vissa tabeller bara behöver spara en referens till en annan tabell istället för att spara all dess data på nytt. Som man hör på namnet använder sig MySQL av Structured Query Language (SQL) som är det programmeringsspråk som används för att hantera datan i en relationsdatabas (*What Is SQL? - Structured Query Language (SQL) Explained - AWS*, 2023). Snabbt, flexibelt och säkert är de egenskaperna som MySQL har, vilket har gjort det till en av de populäraste databaserna (Castro, 2021). Dessutom är det gratis att använda vilket har gjort att det blev valet av databas till gravregistret.

3. KRAVSPECIFIKATION

3.1 Skall-krav

Det som absolut måste finnas innan applikationen kan börja användas är den funktionalitet som redan finns och används i det nuvarande gravregistret. Utan dessa krav skulle den här applikationen vara en nedgradering.

- En gravkarta som visar adressen för alla gravar och inkludera olika objekt på kyrkogården, till exempel kyrkan, för att man lätt ska kunna orientera sig och hitta en grav baserat på en adress.
- Användaren ska kunna lägga till, ändra och ta bort information om både gravar och personer.
- Det ska gå att söka efter namn på personer för att hitta den grav de ligger i
- Gravkartan och en lista med alla begravda personer ska kunna skrivas ut i ett snyggt och läsbart format, där listans sidor är uppdelade efter kvarter.
- Alla fyra av församlingens gravplatser ska ingå i applikationen.

3.2 Bör-krav

De här kraven innehåller funktionalitet som inte finns och skulle förbättra det nuvarande systemet.

- Interaktivitet i kartan. Om det går att både zooma och förflytta kartan så går det att få en överblick över hela kyrkogården och sedan titta närmare på vissa områden för att lättare kunna läsa siffrorna.
- En bild på gravstenen som skulle göra det lättare att hitta den på kyrkogården, speciellt om det är en sådan sten som är svårläslig.
- Likadant system för adressering av gravplatser på de fyra olika kyrkogårdarna
- Inaktivering av gravplatser. Eftersom det teoretiska antalet gravar som får plats på en rad inte alltid stämmer överens med det praktiska så kommer det gå att inaktivera gravplatser om de finns utsatta på kartan men inte får plats i verkligheten. Den här funktionaliteten behövs också om det inte skulle gå att gräva på en viss plats.

3.3 Bra-att-ha-krav

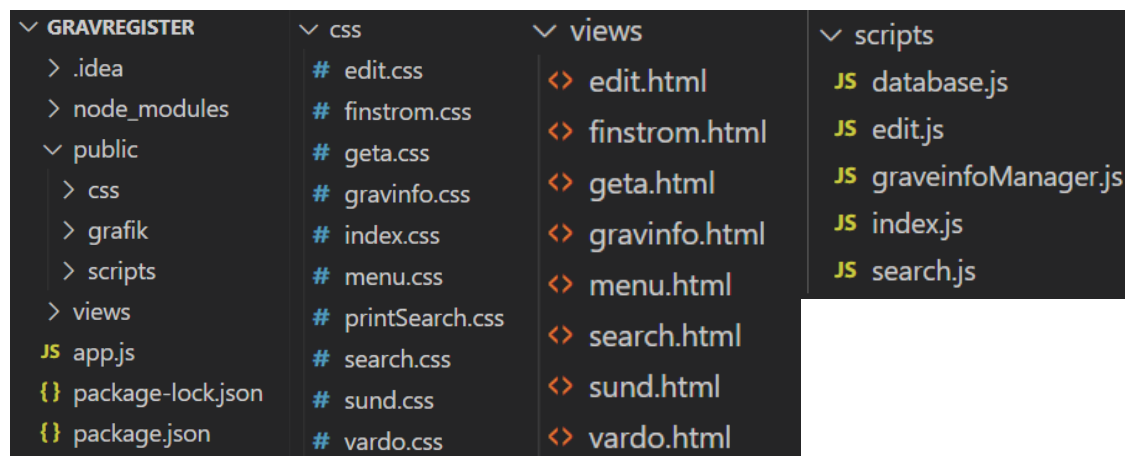
- Visning av statistik, till exempel hur många gravplatser som är lediga
- Instruktioner för användning av applikationen.
- Flera olika färger som visar olika typer av gravar, till exempel kulturgravar.

4. IMPLEMENTATION

4.1 Struktur

Filen *app.js* fungerar som en webserver och tar emot förfrågningar från webbläsaren. När en sida laddas in, skickas en GET-request till webbservern som svarar med sökvägen till rätt HTML-fil. Om sidan behöver tillgång till databasen kommer kommunikationen att ske med en POST-request. Node-servern kan i sin tur skicka förfrågningar till MySQL-databasen som returnerar datan. Före datan kan skickas tillbaka till webbläsaren formateras den till en JSON-sträng.

Användaren skall inte behöva skriva in något i webbläsaren och därför kunna navigera mellan alla vyer via gränssnittet. På samma sätt ska inte användaren heller kunna skriva in eller ändra på information i sökvägen som kan leda till att felaktiga gravadresser läggs till i databasen. Därför finns det implementerat ett system i applikationen som dirigerar om användaren ifall det sker ett olovligt försök att ändra på sökvägen.



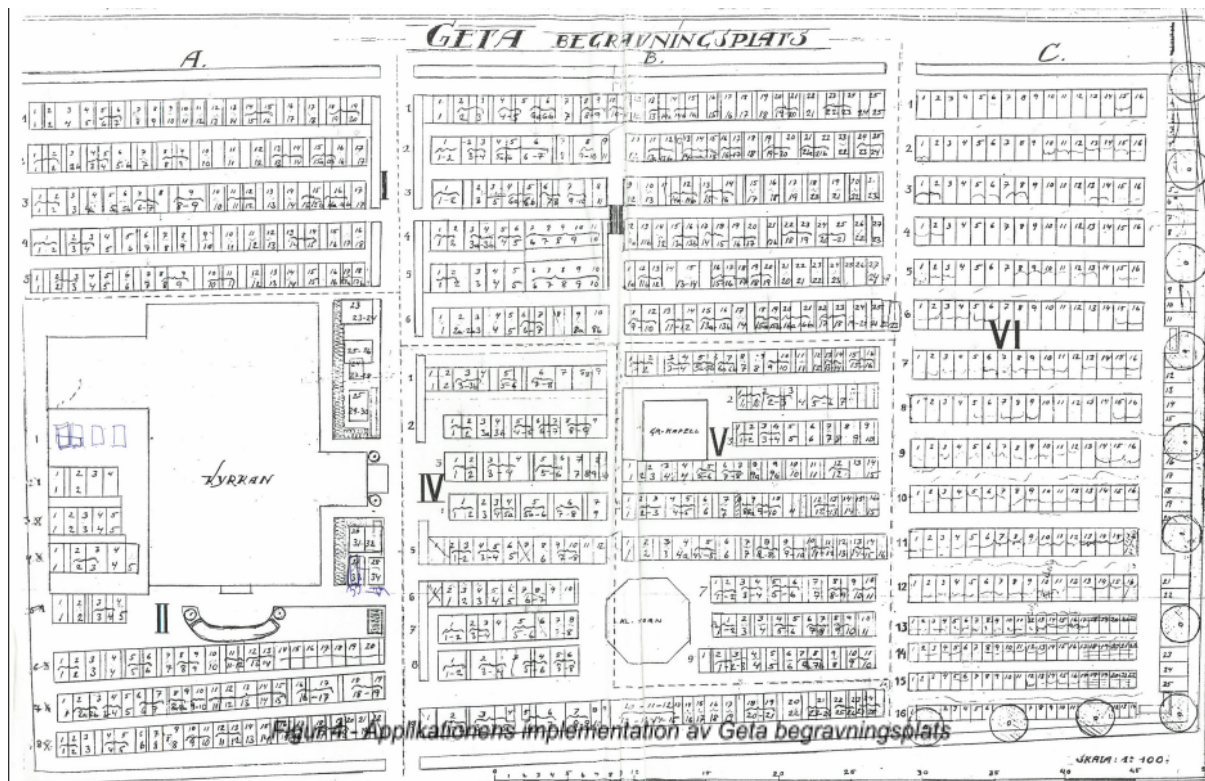
Figur 2 - Filstruktur för applikationen

De olika kartorna fungerar likadant trots att de är olika till utseende. Därför kan de också använda sig av samma skriptfil, vilket är *index.js*, som bland annat innehåller funktionalitet som gör att användaren kan zooma in och ut. Filstrukturen visas i figur 2.

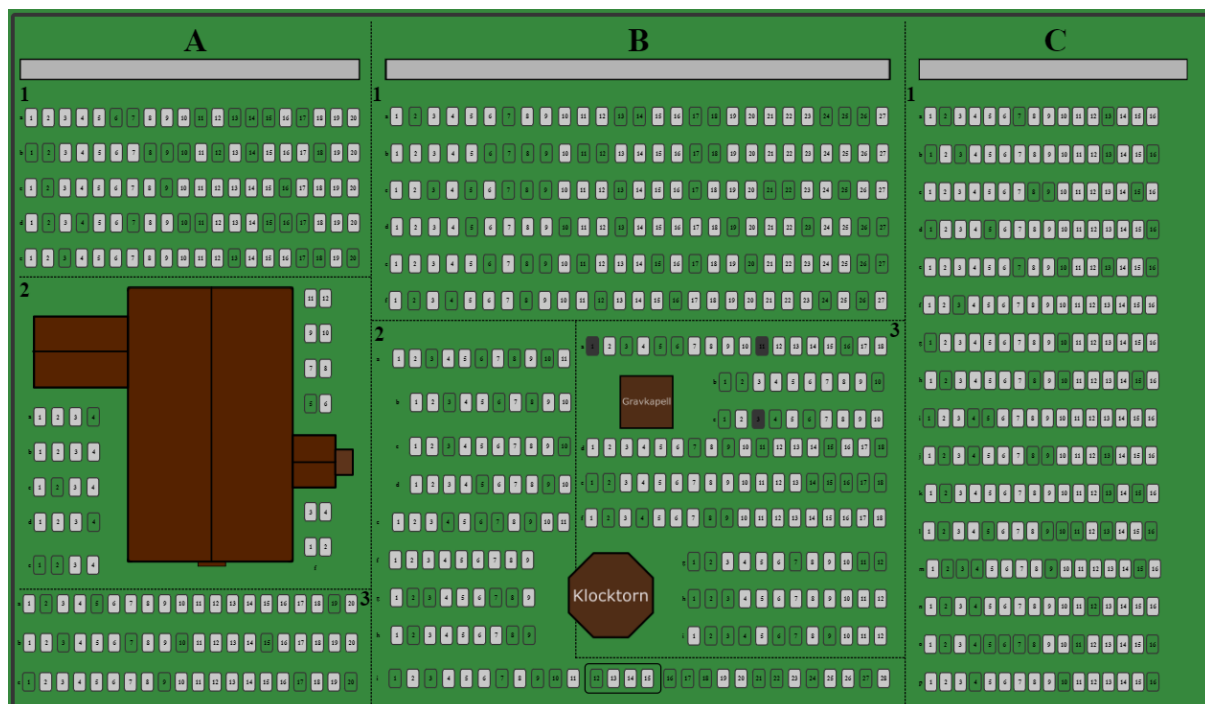
4.2 Grafisk representation av gravgården

Applikationens kartor är huvudsakligen baserade på de gamla ritningarna på begravningsplatserna (Figur 3). Syftet med en gravkarta är att det ska gå att hitta en gravplats baserat på dess adress. För att förenkla den processen krävs det att kartan är tydlig och stämmer överens med verkligheten. Det är nog att raderna med gravar ligger parallellt med varandra på samma sätt som på kyrkogården och att de är positionerade relativt till byggnader och andra objekt i närheten. Kartan kan annars bli missvisande och försvåra arbetet. De olika byggnaderna är ritade i open source-programmet Inkscape som är baserat på vektorgrafik. Användaren kan både flytta runt på kartan och zooma in för att kunna få en översikt över hela begravningsplatsen och sedan titta närmare på mindre områden. En annan fördel med en zoomningsbar karta är att den då kan konstrueras med konstanta värden, vilket underlättar för exaktheten. Storleken och avstånden beräknas i pixlar, vilket gör att kartan kan uppfattas som

liten på stora skärmar och stor på små skärmar. Med den här metoden behöver man bara ändra på skalan för att få en passlig storlek. Applikationens karta visas i figur 4.



Figur 3 - Gravkarta av Geta begravningsplats



Figur 4 - Grafisk representation av Geta begravningsplats

Varje gravelement på kartan har ett färdigt ID som motsvarar gravadressen. Då behöver inte användaren själv skriva in adresser och det finns ingen risk att någon felaktig adress sparas i databasen. HTML tillåter inte element att ha ett id som börjar med en siffra, vilket gör att de gravarna som har en sådan adress använder sig av romerska siffror som sedan översätts med funktionen *transformId*. På gravens färg kan man se dess tillstånd, det vill säga, om graven är ledig, använd eller inaktiv. När man trycker på graven skickas adressen vidare som en parameter via URL-länken som sedan används av webbservern (Figur 5).

```
element.addEventListener("click", event => {  
    var graveId = transformId(event.target.id);  
    window.location.href += "info?id="+graveId;  
});
```

Figur 5 - Kodexempel som ändrar sökvägen och skickar gravens adress som en parameter

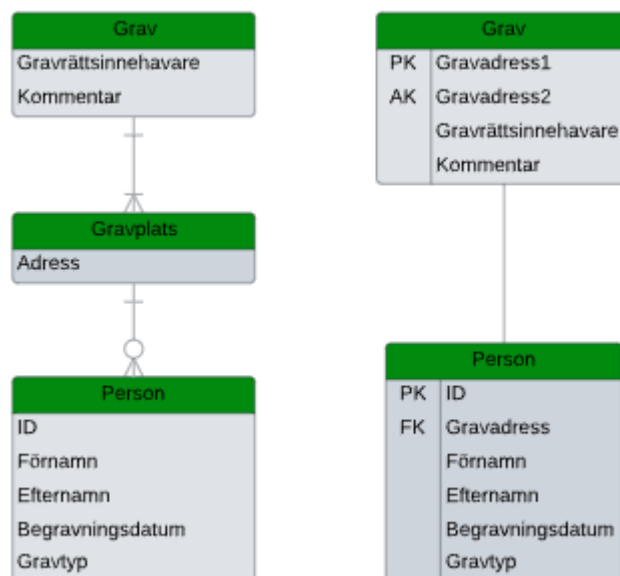
4.3 Databasen

4.3.1 Vilken data sparas?

Databasen innehåller två tabeller med data om gravar och personer. I praktiken skulle det också gå att definiera gravplats som en entitet, men det skulle inte tillföra databasen någonting att ha en till tabell som bara innehåller en adress. Gravplatsens adress finns istället som attribut för både gravar och personer. En persons gravadress hänvisar till kolumnerna *Gravadress1* eller *Gravadress2* i databasen (Figur 6).

En grav består av antingen en eller två gravplatser beroende på om det är en enkelgrav eller en dubbelgrav. Varje gravplats har i sin tur sin egen adress.

En gravadress innehåller tre delar som separeras med bindestreck, till exempel *B2-f-5*. Den första delen visar gravens kvarter, som är ett mindre område på begravningsplatsen som oftast skiljs åt från andra kvarter med grusvägar eller stigar. I de flesta fallen är begravningsplatsens kvarter strukturerade som ett rutnär och då blir namnet på kvarteret baserat på raden och kolumnen. Del två är en bokstav som representerar radnumret där A är första raden, B är andra raden och så vidare. Den sista delen av adressen är numret på var i raden graven finns. Rad och nummer betecknas olika i församlingens nuvarande register beroende på vilken kyrkogård det är.



Figur 6 - Till vänster, ett E-R diagram över databasens entiteter. Till höger, ett diagram över databasens struktur med angivna nycklar.

Det nya registret kommer dock att använda samma system för alla ställen. Fördelen med att ha både bokstav och nummer är att det finns mindre risk att blanda ihop dem ifall adressen måste kommas ihåg.

Det finns två fält i tabellen för adresser, på enkelgravar är den ena alltid tom och på dubbelgravar är båda ifyllda, trots att det kanske bara är någon begravd i den ena gravplatsen. En gravrättsinnehavare är den person som har rätt att använda en grav, åt sig själv eller någon annan (*Att Vara Gravrättsinnehavare En Liten Guide*, n.d.). Oftast blir det den som är närmast anhörig till den som senast blev begravd där. Det är viktigt att det finns uppgifter om gravrättsinnehavaren då den måste kontaktas, till exempel om något händer med graven. Speciellt på äldre gravar kan innehavaren ofta vara okänd. Det kan hända till exempel om personen avlider och det inte går att få tag på någon annan anhörig. GDPR-lagen, som är EUs globala dataskyddslag, innehåller ingen artikel som explicit uttrycker något gällande uppgifter om avlidna personer. Enligt artikel 4.1 definieras persondata till den information som hör till en fysiskt identifierbar person (*General Data Protection Regulation* 2016). Någon som är begravd kan inte identifieras på det sättet, speciellt om det gått några år efter begravningen. Vissa länder har sina interna lagar som skyddar uppgifterna, men det förekommer inte i Finland (*GDPR Tracker - Personal Data of Deceased Persons*, 2023).

Irregularitet förekommer med jämna mellanrum på en gravgård, det kan vara en grav som måste grävas på annat djup än normalt eller någon annan faktor som försvårar en grävning. Därför finns det ett fält i databasen för anteckningar som gäller generellt för hela graven. Där kan saker noteras om gravstenen, gravplatsen eller enskilt begravda personer.

Den andra tabellen innehåller information om den begravda personen, det vill säga alla förnamn, efternamn, begravningsdatum, adress på gravplats och gravtyp (kista eller urna). Det diskuterades med uppdragsgivaren ifall födelsedatum och dödsdatum också skulle ingå i registret, men dessa uppgifter behövs mycket sällan så de uteblir för tillfället. Eventuellt kan det ingå i en senare version av applikationen, men det kommer att bli valfritt att fylla i det. Ett annat argument är också att det är föredraget att spara så lite information som möjligt. Inget av dessa attribut skulle fungera som primärnyckel och därför genererar databasen ett ID till varje person. Detta ID är osynligt för användaren men det behövs till exempel för att ta bort en person ur databasen. Flera personer kan begravas på en och samma gravplats. Databasen har ingen begränsning för hur många personer som kan ha samma adress, men applikationen har ett maxantal på 9 personer per gravplats. I praktiken kommer troligtvis det antalet aldrig att uppnås då en generation skulle behöva bruka en och samma grav i över 200 år.

4.3.2 Applikationens kommunikation med databasen

En gravs adress är nyckeln till databasen. Både gravens och personernas data hämtas ut via den. Adressen skickas med en POST-request till node-servern som använder en funktion ur filen *database.js* som kör en SQL-query och returnerar motsvarande data från MySQL-databasen. Datahämtning kommer att ske två eller tre gånger beroende på om graven innehåller en eller två adresser. Slutligen sammanfogas all data till en JSON-sträng som skickas till frontend-delen av applikationen. Figur 7 visar koden som körs på webbservern för att hämta data ur databasen och sedan returnera datan till klienten.


```

app.post("*/info", async function(req, res) {

    //Funktion som hämtar data om en grav
    const gravedata = await getFromIdGrave(req.query.id);

    if(gravedata.length <= 0){
        //console.log("Data not found");
        return res.send(JSON.stringify(gravedata)); //Skickar tillbaka en tom JSON-sträng
    } else {
        //Hämtar persondata på de som ligger i gravplats 1
        const persondata1 = await getFromIdPerson(gravedata[0].first);
        //Hämtar persondata på de som ligger i gravplats 2
        const persondata2 = await getFromIdPerson(gravedata[0].second);

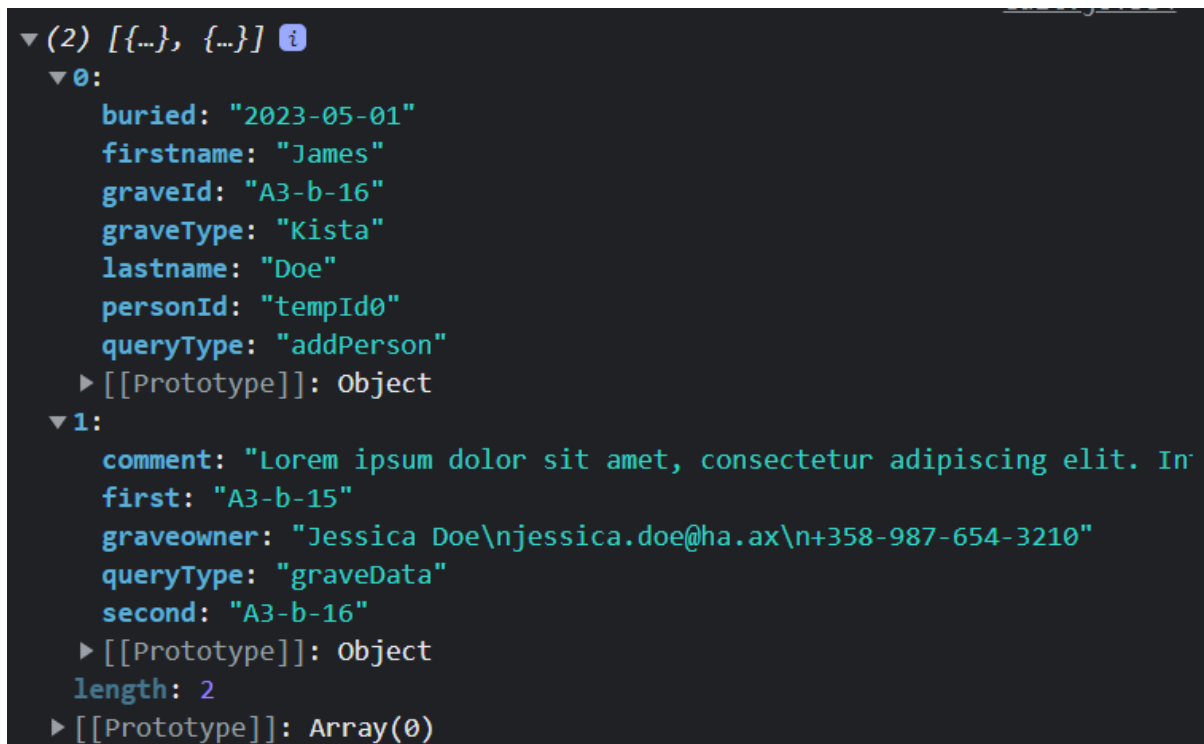
        //Sammanfogar all data till ett JSON-objekt
        const mergedJson = [];
        mergedJson.push(gravedata);
        mergedJson.push(persondata1);
        mergedJson.push(persondata2);

        const infoString = JSON.stringify(mergedJson);
        //Skickar tillbaka datan till klienten
        res.send(infoString);
    }
});

```

Figur 7 - Webbservern som hämtar ut data från databasen och skickar det vidare till klienten

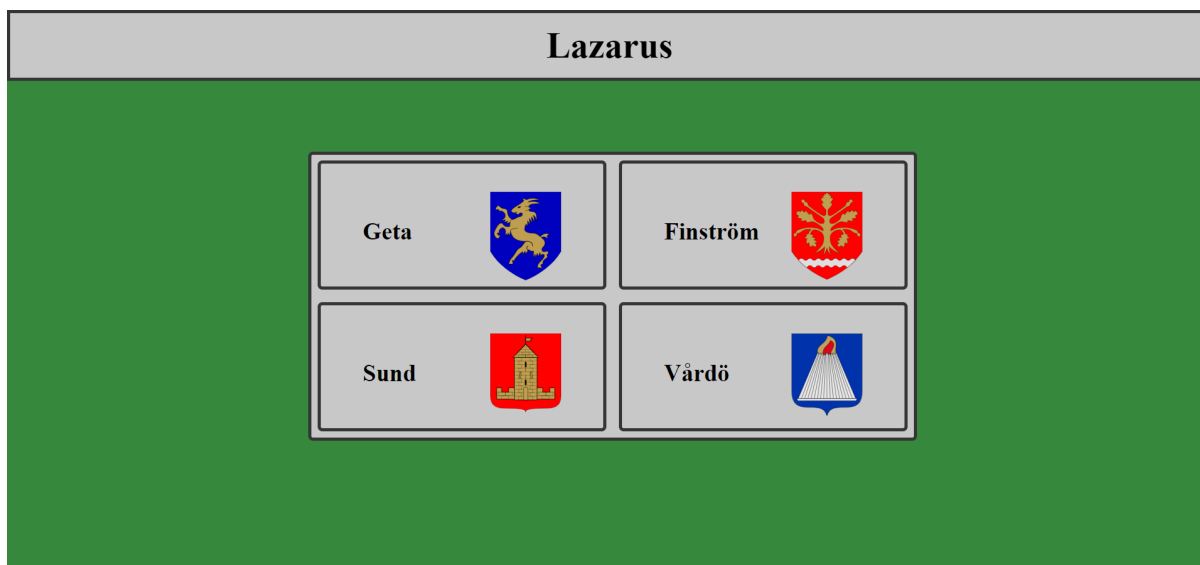
När en grav läggs till eller ändras så skall det gå att utföra flera operationer och sedan skicka alla ändringar till databasen på samma gång. Användaren ska också kunna avbryta en ändring för att kunna bevara den tidigare informationen ifall ett misstag sker. För att lösa det problemet sparas alla uppdateringar i en lista som sedan skickas till servern som hanterar dem en efter en. Listan består av JSON-objekt som innehåller alla värden som behövs för ändringen samt en variabel som innehåller vilken typ av ändring det är, till exempel tillägg, ändring eller borttagning av en person. Figur 8 visar ett exempel på en sådan lista.



Figur 8 - Lista med operationer som skickas till webbservern

Fördelen med en lista är också att vissa förändringar kan ske i listan och behöver inte bearbetas av databasen. Till exempel om användaren försöker lägga till och ta bort samma person upprepade gånger så kommer förändringen att tas bort från listan istället för att fråga databasen att lägga till och ta bort samma person. Ett tekniskt problem är också att en person inte får ett ID förrän den har lagts till i databasen, men den behöver sitt ID för att kunna tas bort. Personer som har lagts till i gränssnittet men inte i databasen kommer ha ett temporärt ID som kan identifieras. När en person tas bort kommer applikationen att kunna veta om personen finns i databasen eller är nyligen tillagd och därefter bestämma om den ska tas bort från listan eller från databasen. En grav kan bara bli vald till enkel- eller dubbelgrav då den läggs till. Församlingen tillåter inte sådana förändringar i efterhand.

Varje begravningsplats har ett eget schema som innehåller två tabeller med data om gravar och personer. Dessa tabeller har både samma namn och struktur, vilket gör att samma SQL-kod kan användas. När användaren väljer begravningsplats i huvudmenyn (figur 9) kommer webbservern att ändra till den valda begravningsplatsens schema.



Figur 9 - Applikationens huvudmeny där användaren kan välja begravningsplats

4.4 Gravdata

Användaren kan läsa och skriva till databasen via gränssnittet som visas i figur 10. Två tabeller innehåller informationen om personerna där den ena tabellen är dold om det är en enkelgrav. Gravens adress visas ovanför tabellen där den med lägre nummer alltid ligger till vänster. När en ny grav läggs till kan användaren välja om graven skall vara en dubbelgrav och isåfall välja vilken närliggande grav som skall ingå. Detta förutsätter att det finns en grav bredvid som är tillgänglig och att användaren inte redan har satt in någon person i graven.

När en persons uppgifter läggs till eller redigeras sker det i ett separat fönster (Figur 11). Då finns det tillräckligt med plats för alla nödvändiga element. Användaren kan inte heller interagera med bakgrunden förrän operationen är genomförd eller avbruten. Ett sätt för att förhindra otillåtna datum att skickas till i databasen är att använda *input*-elementet av typen *date*. Problemet är dock att datumet returneras i ett format där året skrivs först vilket gör att det måste finnas en funktion som kan översätta datumet innan det visas åt användaren. Alla ändringar som görs i det här gränssnittet läggs till i listan som nämns i kapitel 4.3.2. När användaren trycker på knappen för att spara informationen skickas listan vidare till webbservern.

Figur 10 - Gränssnitt då användaren lägger till en ny grav

Figur 11 - Gränssnitt då användaren lägger till en ny person

Det första som gjordes på det här gränssnittet var att organisera hur fälten skall vara placerade. Planen var att ha en bild i mitten och sedan personuppgifter på varsin sida och information om själva graven längst ned. Personuppgifterna tog mycket mer plats än förväntat, speciellt om det är någon med flera långa mellannamn. I en framtida version av det här gravregistret kommer troligen alla personuppgifter att visas i ett separat fönster på samma sätt som personer läggs till. Alla namn kommer fortfarande att visas på samma sätt.

4.5 Sökfunktion

En av de vanligaste situationerna då gravregistret behövs är när någon ska begravas i en gammal grav och det bara är namnet på någon i graven som man känner till. Då måste det gå att söka på det namnet för att hitta rätt grav. Sökfunktionen skall kunna ta in en sträng som innehåller ett eller flera namn och sedan hämta ut alla personer ur databasen med matchande namn. Det ska inte vara någon skillnad på i vilken ordning man skriver förnamnen eller efternamnet. Resultaten visas i en lista som är sorterad efter gravadress och användaren kan sedan trycka på en person i listan för att se fullständig information om den graven som personen ligger i. En lista på sökresultat visas i figur 12.

Adress	Förnamn	Efternamn	Begravningsdatum	Gravtyp
A3-b-15	Jane	Doe	12-12-1997	Urna
A3-b-16	John	Doe	22-06-1995	Kista
A3-b-16	James	Doe	01-05-2023	Kista

Figur 12 - Resultat av sökning med nyckelordet 'Doe'

En metod av säkerhetskopiering som uppdragsgivaren har använt sig av är att skriva ut hela gravregistret en gång i året och sätta alla papper i en pärm. Där är alla sidor indelade efter gravkvarter. Listan som sökfunktionen hämtar ut kan skrivas ut på det viset.

5. SLUTSATSER

5.1 Resultat

Lazarus blev en applikation som innehåller samma funktionalitet som den tidigare versionen av gravregistret. Användaren kan lägga till, ändra och ta bort nya gravar och personer i registret. Den största förändringen är den grafiska delen som gör registret mycket tydligare och jämfört med Excel innehåller det inte funktionalitet som är onödig och kan förvirra användaren. Kartan och all data finns samlade på ett ställe och det går snabbt att byta mellan dem. När en ny grav öppnas kan användaren hitta potentiella platser för graven genom att titta på kartan. Platsen måste dock kontrolleras ute på gravgården innan förslaget verkställs, men det förenklar uppgiften.

Genom att söka på en persons namn kan dennes grav snabbt hittas. Samma sökfunktion kan användas för att hämta ut alla personer som är begravda på en kyrkogård. Både den listan och kartan kan skrivas ut och användaren behöver inte tänka på formatet, det sköter applikationen.

Vissa gravplatser finns utmarkerade på kartan trots att de inte får plats i verkligheten, dessa gravplatser kan inaktiveras i applikationen och blir därmed markerade på kartan.

5.2 Diskussion

Den här typen av arbete är unikt, speciellt på Åland. Det skulle vara svårt att få ihop ett allmänt register som skulle kunna användas av vilken församling som helst, eftersom kartan är specifik för kyrkogården och det skulle kräva att alla kyrkogårdar och församlingar hade samma struktur och arbetssätt. En applikation som Lazarus måste skraddarsys för att passa in på församlingens behov. De flesta församlingarna på Åland är små och har mycket begränsade resurser, vilket betyder att de inte skulle ha råd att anlita ett IT-företag att skapa en liknande applikation.

Trots att applikationen innehåller den viktigaste funktionaliteten så finns det en hel del kvar att göra. Två av kartorna behöver fortfarande ritas, men det krävs också att begravningsplatserna kontrolleras i verkligheten. Vissa gravar vårdas av församlingen och dessa skall vara utmärkta på kartan för att kyrkogårdsskötarna ska kunna hitta dem. Varje grav skall innehålla en bild på gravstenen så att det blir lättare att identifiera graven på gravgården. Gränssnittet som visar informationen om graven kommer att få en ny design så att bilden ska passa in. Ytterligare uppdateringar kommer också att ske om nya önskemål tillkommer.

Applikationen har kört en del funktionstester under utvecklingens gång, men en mer övergripande testning kommer att krävas innan den kan användas. Detta kommer att ske innan applikationen kan tas i bruk då ny funktionalitet som läggs kommer att kräva en ny testsession. Tiden har varit begränsad, vilket gör att utvärdering och korrigering av applikationen kommer ske vid ett senare tillfälle. Det här projektet har lagt fokus på implementationen. När applikationen tas i bruk kommer den att köras parallellt med det tidigare registret under en viss tid för att säkerställa att inga uppgifter går förlorade. Filerna kommer att ligga på uppdragsgivarens server så att applikationen är tillgänglig på alla deras

maskiner. Node-servern kommer att startas via en exe-fil så att applikationen blir lättillgänglig via skrivbordet.

Förutom att få en bra fungerande applikation är ett av målen för det här projektet att utvecklas som programmerare. Arbetet har pågått under en längre tid och när jag tittar tillbaka på det som gjordes i början av projektet så inser jag att det är på en lägre nivå än det som är nytt. Gammal kod som jag inte är nöjd med kommer troligen att få en uppdatering i framtiden. Förutom bara programmering har jag även haft ansvar för strukturen av applikationen, det vill säga vilka funktioner som skall finnas, hur de ska vara uppbyggda och utseendet på användargränssnittet. Allting måste dock ligga innanför ramarna för uppdragsgivarens krav. Under projektets gång har jag också bekantat mig med församlingens gravplatser, vad det finns för regler och vilka arbetssätt som används.

KÄLLFÖRTECKNING

About Node.js. (n.d.). Node.js. Retrieved May 3, 2023, from <https://nodejs.org/en/about>

Almeida, J. (2022, June 9). *Node.js vs. PHP: An In-depth Comparison Guide for Web Development.*

DistantJob. Retrieved May 12, 2023, from <https://distantjob.com/blog/nodejs-vs-php/>

Att vara gravrättsinnehavare en liten guide. (n.d.). Svenska kyrkan. Retrieved May 3, 2023,

<https://www.svenskakyrkan.se/filer/Broschyr%20Att%20vara%20gravr%C3%A4ttsinnehavare.pdf>

Begravningslag, (457/2003). Justitieministeriet (2003).

<https://www.finlex.fi/sv/laki/ajantasa/2003/20030457>.

Bibel 2000: texterna : Gamla testamentet, Tillägg till Gamla testamentet - de apokryfa eller

deuterokanoniska skrifterna, Nya testamentet : Bibelkommissionens översättning 1999

(Sverige. Bibelkommissionen, Trans.). (2000). Verbum.

Castro, S. (2021). *5 Reasons Why MySQL Is Still The Go-To Database Management System.* Jobsity.

Retrieved May 12, 2023, from

<https://www.jobsity.com/blog/5-reasons-why-mysql-is-still-the-go-to-database-management-system>

GDPR Tracker - Personal data of deceased persons. (2023). Bird & Bird. Retrieved May 3, 2023,

from

<https://www.twobirds.com/en/capabilities/practices/privacy-and-data-protection/general-data-protection-regulation/gdpr-tracker/deceased-persons>

General Data Protection Regulation, (679/2016). Europeiska Unionen (2016).

<https://gdpr-info.eu/art-4-gdpr/>.

Kungliga Maj:ts Begravningskungörelse, (540/1963). Sveriges riksdag (1963).

https://www.riksdagen.se/sv/dokument-lagar/dokument/svensk-forfattningssamling/kungl-majts-begravningskungorelse-1963540_sfs-1963-540.

Medlemsstatistik 2023. (2023). Kirkon tilastot. Retrieved May 3, 2023, from

<https://www.kirkontilastot.fi/viz.php?id=247>

What is JavaScript? - Learn web development | MDN. (2023, March 4). MDN Web Docs. Retrieved

May 3, 2023, from

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript

What is SQL? - Structured Query Language (SQL) Explained - AWS. (2023). Amazon AWS.

Retrieved May 3, 2023, from <https://aws.amazon.com/what-is/sql/>