**Databasschema. Tabeller och kolumner i databasen, med tillräckliga förklaringar för att man ska kunna förstå vad innehållet betyder. Ett diagram över schemat i en eller annan form är utmärkt att ha – det gör det lättare att prata om – men inget absolut krav.**

Den fösta tabellen som gruppen har gjort är Anslutningar. Till den finns kolumnerna avgångsland, avgångsstad, avgångsadress, ankomstland, ankomststad och ankomstadress. Här är primärnycklarna avgångsland, avgångsstad, ankomstland och ankomststad.

Denna tabell har syftet att snabbt och lätt kunna presentera vilka resor som användaren kan välja mellan.

Tabellen Resa finns även och är en av de längre tabellerna. Här finns information om resor vilket gör att mycket information kommer att finnas med. Kolumnerna innehåller all information som finns om individuella resor.

Tabellen chaufför innehåller information om vilka chaufförer som finns på sidan. Tabellen innehåller standard personlig information som finns. Det ett personnummer som även är tabellens primärnyckel då detta värde alltid är unikt i tabellen. Sedan finns kolumnerna förnamn och efternamn vilket är rätt självklart för en tabell av denna typ. Samt att adress och telefonnummer finns med då den informationen kan vara viktig för arbetsgivaren att ha.

Sedan finns tabellen Kund som innehåller information som är nödvändig för att information om kunde ska kunna presenteras bra. Den första kolumnen som även är primärnyckeln är kundid. Detta är ett unikt nummer som varje kund får när den registrerar sig. Här finns även som i förra tabellen namn, adress, e-post och telefonnummer. Detta är bra information att ha för resebyrån så att de kan kontakta personen på flera olika sätt.

Efter det så finns tabellen Städer.

Den tabellen innehåller avgångsland, avgångsstad och avgångsadress. I denna tabell så finns två primärnycklar, avgångsland och avgångsstad. Dessa i kombination gör att det finns unika värde. Antagandet vilket gruppen bedömer som rimligt är att det inte finns mer än en stad i samma land med samma namn. Tabellen är utformad för att hantera vilka städer som finns tillgängliga.

**Redogörelse vad ni använt för att bygga applikationen. Programspråk, kodbibliotek osv.**

I enlighet med kursens krav används en databas, språket för databasen i fråga är postgresql. Gruppen har utvecklat en webbaserad applikation för Mörtfors buss. Denna webbapplikation bestående av html, css och javascript är baserad på ramverket bootstrap.

För att koppla ihop webbapplikationen med postgresql så har flask som är ett ramverk för python använts. För att brygga ihop python och postgresql så har databasadaptern psycopg använts.

**Kortfattad överblick över applikationens komponenter, från programmerarens perspektiv, och beskrivning vad de gör.**

Appen börjar med att definiera beroende som t.ex. databasen och dust hanterare. Efter det så etablerar Node.js en anslutning till databasen med användare, lösenord och adress. Sedan laddas fler dust beroende in och public folderns väg definieras. Efter det laddas huvudsidan tillsammans med databasen. Sedan kommer det paths som finns att navigera till. Dessa returnerar en dust template och hanterar eventuella errors som kan förekomma när hemsidan laddas in. Efter det så kommer en postgresql sats som kommunicerar med databasen. Detta kan vara en select, insert eller update sats för att funktionen ska få önskad effekt på databasen. Det kommer att finnas lämplig logik i programmet vid varje path. Detta kan innebära en if sats som kontrollerar att användaren inte kan köpa fler resor än vad som finns tillgängliga. Dust filer har även Node.js kod i sig då detta används för att visa information från databasen.

**Kortfattad beskrivning av hur man använder er applikation (eller era applikationer) från användarperspektiv.**

Då applikationens primära mål är för gruppen att uppvisa färdighet i postgresql så har valet gjorts att applikationen är utformad att kunna användas av både Mörtfors buss och resenärer under samma gränssnitt. Hade kursens krav varit att utveckla en applikation för ett riktigt företag så hade applikationen delats upp i två applikationer. Ett program för Mörtfors buss samt ett program för resenärer.

Applikationen innefattar flertalet olika delar som är sammankopplade och i vissa fall beroende av varandra. I stora drag är applikationen utformad så att en användare kan registrera kunder, registrera chaufförer, registrera hållplatser, och avgångar i respektive register. Det finns även möjlighet att se transaktioner dvs kunder som köpt resor.

Utöver denna funktion finns även möjlighet att söka efter avgångar samt köpa biljetter för respektive avgångar. Sökfunktionen möjliggör att specificera sin sökning så som: resor från Malmö eller resor som körs av chaufför x.

Då olika tabeller har sammanband och dessa sammanband är primärnycklar så är också de olika delarna beroende av varandra. Det går tex inte att skapa en avgång från London om inte först London finns registrerat i hållplatser. Om en kund försöker köpa fler biljetter än vad som finns kvar för en avgång går inte köpet igenom.

Applikationen ger även feedback till användaren i form av gröna eller röda textfält efter att användaren utfört något.