

Bagpacker

Grupp 9

Designdokument

V. 4.0
2019-06-02

Dokumenthistorik

Datum	Version	Beskrivning	Författare
190324	1.0	Påbörjat <i>Produktbeskrivning</i>	Sebastian Tyrting
190325	1.1	Skrivit syfte och redigerat <i>Produktbeskrivning</i>	Sebastian Tyrting
190326	1.2	ER-diagram och tillhörande beskrivning	Christina Knepper
190326	1.3	Skrivit användningsfall	Iris Brinkborg
190328	1.4	Beskrivning av systemdiagram och information om databasen.	Christina Knepper
190327	1.5	Tillägg: Användningsfall	Iris Brinkborg
190328	1.6	Tillägg: Systemdiagram	Iris Brinkborg
190328		Tillägg: Beskrivning av systemdiagram och kompletterat med information om databasen.	Christina Knepper
190329	1.7	Skisser av användargränssnittet	Iris Brinkborg
190409	2.0	Revidering av dokument	Iris Brinkborg
190411	2.1	Omarbetning av stycket <i>Produktbeskrivning</i>	Johan Wingren
190414	2.2	Gjort ändringar efter gransknings-input. Samt skrivit om Syfte. Korrekturläst.	Sebastian Tyrting
190416	2.3	Omarbetning av stycket <i>Produktbeskrivning - Webbapplikation</i>	Christina Knepper
190417	2.4	Ändrat ordlistans utseende	Sebastian Tyrting
190418	2.5	Tillägg: ordlista. Klassdiagram	Ekaterina Korotetskaya
190424	3.0	Reviderat dokumentet efter feedback från retrospektmöte 2.	Iris Brinkborg
190509	3.1	Uppdatering av skisserna för utseendet på webbgränssnittet, uppdaterat ordlista och omformulerat produktbeskrivning.	Iris Brinkborg
190509	3.2	Korrektur och formatering	Johan Wingren
190528	4.0	Omarbetning av stycket <i>Produktbeskrivning</i> , revidering av dokumentet efter retrospektmöte 3, tillägg: <i>Pythonapplikationen på webbservern</i> .	Christina Knepper
190531	4.1	Har gjort om klassdiagrammet	Ekaterina Korotetskaya
190601	4.2	Omarbetning av ER-diagrammet samt tillhörande beskrivning. Lagt till ord i ordlistan.	Christina Knepper
190602	4.2	Skrivit inledning till <i>Scenarion/Användningsfallsbeskrivningar</i>	Iris Brinkborg

Innehåll

Dokumenthistorik	2
Designdokument	4
Syfte	4
Ordlista	4
Referenser	6
Produktbeskrivning	6
Webbapplikation	6
Androidapplikation	8
Databas	8
Systemdiagram	8
Klassdiagram	9
Användningsfallsdiagram	10
Scenarion/Användningsfallsbeskrivningar	10
Användargränssnitt	13
Diagram/skiss	14

Designdokument

Syfte

Detta dokument är ett komplement till produktbeskrivningen som finns i projektplanen. Syftet med dokumentet är att tillhandahålla fler detaljer som hjälper till att visualisera hur applikationerna som ingår i systemet kommer att se ut samt vilka krav som kommer att finnas med.

Dokumentet kommer bland annat att innehålla användningsfallsdiagram som visar hur användaren interagerar med produkterna. Scenarion eller användningsfallsbeskrivningar som beskriver flödet mellan systemet och användaren. Skisser på användargränssnitt visar hur produkterna är tänkta att vara utformade.

Ordlista

Android-applikation	Mjukvara som exekverar på en Android-enhet.
användarinput	All form av data som användaren matar in i systemet.
API	Akronym för Application Programming Interface. Fungerar som gränssnitt mellan applikation och tredjeparts kodbibliotek.
autofyll	Automatisk ifyllning av inmatningsfält i formulär.
avresa	Parameter som anges i webbformuläret. Avser det datum då användaren planerar att påbörja sin resa.
ER-diagram	Står för <i>entity relationship diagram</i> . ER modellen är en konceptuell datamodell som beskriver samband mellan olika entiteter (t.ex. saker, personer) som finns i en databas.
etikett	Ett identifierande adjektiv som användaren uppger för att sätta en prägel på resan, vilket i sin tur påverkar filtreringen av packabler till packlistan.
formulär	Element på en webbsida som innehåller fördefinierade fält som användaren kan fylla

i. När användaren klickar på en knapp skickas användarens input till webbservern via en HTTP-förfrågan.

HTML

Hypertext Markup Language, språket som används för att skapa webbsidor.

HTTP-förfrågan

HTTP står för hypertext transfer protocol och betecknar kommunikationsprotokollet som används för att överföra data mellan klient och server.

klassdiagram

presenterar klasser som ingår i systemet och relationer mellan dem.

kryssruta

Ruta som kan markeras och avmarkeras i användargränssnittet. Användaren kan markera flera alternativ.

MySQL

Databashanterare som använder frågespråket SQL.

mockup

En skissartad modell av en produkt. Används för att beskriva interaktionsvägar och grafisk design för produkten.

ogiltig inmatning

Kan innebära att användaren angett ett felaktigt format på lösenord, ex för kort. Kan även referera till att lösenorden inte matchar eller att användarnamnet är upptaget. Utöver detta kan det innebära att lösenord eller användarnamn är felaktiga.

packabel ~n, pl. ~er

Objekt som finns i packlistan. Varje föremål har ett antal olika parametrar, t.ex. namn, vikt, kategori och min-/max-temperatur.

parameter

Data som användaren matar in via ett formulär

returresa

Parameter som anges i webbformuläret. Avser det datum då användaren planerar att avsluta sin resa.

webbapplikation

Programvara som exekveras på en webbserver och som användaren kan

interagera med via en webbläsare.

widget

Förkortning för “window gadget”. Grafisk användargränssnittskomponent. Här syftar det till kalendern som visas när användaren klickar på inmatningsfältet

Referenser

- [1] I. Sommerville, *Software Engineering*. 10th ed., Essex: Pearson Education Limited, 2016.
- [2] J. Leroy, *Packr*. Paris: 2018.
- [3] M. Wechsler, *Packking*. Weissensberg: 2017.
- [4] “Jordens huvudklimattyper”, *SMHI*, 20 mars 2019, [Online]. Tillgänglig: <http://www.smhi.se/kunskapsbanken/klimat/jordens-huvudklimattyper-1.640> (hämtad: 30 maj 2019).

Produktbeskrivning

Bagpacker är en tjänst som ska förenkla planeringen och förberedelserna inför en resa genom att tillhandahålla en skräddarsydd packlista. Användaren anger resmål, resans längd, planerade aktiviteter osv, sedan genereras en packlista utifrån dessa parametrar. Användaren ska även tillåtas ändra i packlistan efter behov.

Systemet består av två delar. Dels en webbapplikation, dels en Androidapplikation. Dessa hämtar information från en databas som projektgruppen skapar. Både i Android- och webbapplikationen ska användaren kunna ladda ner, spara och redigera färdiga listor.

Webbapplikation

Webbapplikationen är den webbaserade plattformen för interaktion mellan användaren och programmet. Startsidan visar ett formulär där användaren får ange ett antal parametrar, varpå ett förslag på packlista genereras och visas för användaren. I ett senare skede kommer startsidan eventuellt även att ge användaren möjlighet att logga in alternativt skapa ett konto. Relevanta parametrar som ska anges i formuläret är:

- *Resmål* - Den geografiska plats som utgör målet för resan. Genom en autofyllfunktion (som stöds av en API) ska användaren få hjälp att mata in destinationen.
- *Resans längd* - Det antal dagar som resan omfattar. Dessa anges som avresedatum och returresedatum i en kalender. Datum för resan behövs för att kunna ta reda på vädret på destinationen - antingen utifrån aktuella prognoser eller historisk väderdata.
- *Färdmedel* - Vilken typ av färdmedel som ska användas för att ta sig till resmålet.
- *Boende på resmålet* - Typ av boende, t.ex. hotell, lägenhet, husvagn, tält.
- *Planerade aktiviteter* - Dessa väljs från en lista av aktiviteter som kräver diverse specialutrustning. Användaren ska kunna ange flera olika aktiviteter.

Webbgränssnittet / formuläret ska gestaltas på ett användarvänligt sätt och samtidigt förebygga felaktig inmatning. Detta implementeras

1. via attribut i HTML-formuläret:

- placeholders: grå text i en inmatningsruta som exemplifierar hur rutan ska fyllas i
- required: om användaren har glömt eller underlåtit att fylla i ett fält i formuläret visas ett meddelande upp (kan se olika ut beroende på vilken webbläsare som används) och det går inte att klicka på "generera packlista"-knappen

2. med hjälp av Javascript/jQuery:

- autofyllfunktion som ger förslag på destinationer under tiden användaren matar in bokstäver via tangentbordet
- kalenderwidget: när användaren klickar i datum-fälten visas en kalender där start- och slutdatum för resan kan väljas genom att klicka
- validering av formuläret: ett felmeddelande visas upp och knappen för att ta sig vidare till nästa steg inaktiveras när användaren exempelvis inte har fyllt i något

3. med hjälp av klickbara bilder/ikoner som symboliserar olika aktiviteter / typer av boende eller transport och som skiftar färg när man klickar på dem

Ambitionen med formuläret är att begränsa inmatning via tangentbord eftersom det är en möjlig felkälla. I formuläret finns det bara ett sådant fält, alla andra parametrar ska matas in via kryssrutor. Nackdelen med förvalda alternativ är att formuläret kan upplevas som för begränsat eftersom det kanske inte täcker in alla önskemål som användaren har. Fördelarna är dock att felaktig inmatning minimeras samt att det är enkelt och tidsbesparande för användaren att fylla i formuläret. Ur ett utvecklingsperspektiv är förutbestämda alternativ dessutom en viktig förutsättning för att kunna utforma databasen, t.ex. genom att koppla vissa packabler till parametrarna i formuläret.

När det gäller presentationen av packlistan ska den inmatade destinationen samt information om vädret på destinationen visas upp för användaren. Föremål på packlistan visas grupperat efter olika kategorier för att underlätta själva packningsprocessen. Användaren ska även få

information om packningens förväntade totalvikt samt förslag på antal som bör packas av respektive packabel. I ett senare skede är det tänkt att användaren ska kunna modifiera listan genom att lägga till och ta bort packabler eller genom att justera antal. Utöver det ska användaren ha möjlighet att spara listan och välja en egen titel för den.

Möjliga tillägg är flikar med bakgrundsinformation om själva projektet samt vem som har utvecklat applikationen, ett kontaktformulär där användare kan lämna förbättringsförslag eller en informationssida med tips på smarta packlösningar.

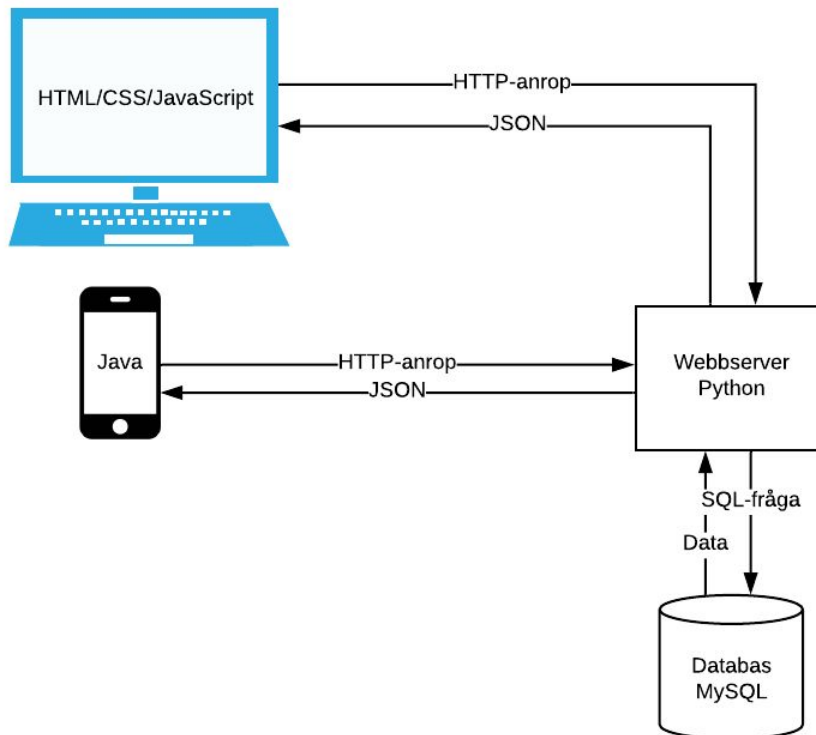
Android-applikation

Även Androidapplikationen innehåller ett formulär som liknar det ovan nämnda formuläret för webbgränssnittet och som tillåter användaren att ange parametrar gällande den planerade resan. Applikationen kommer att hämta listan som skapats på webbservern och låta användaren modifiera, spara och i ett senare skede eventuellt synkronisera denna med databasen. Modifieringen kommer bestå i att lägga till och ta bort packabler samt att markera de packabler som packats. Den grafiska designen kommer att spegla webbgränssnittets utseende, i den mening att färgval, symboler och fält kommer vara genomgående lika. Syftet med applikationen är att den ska fylla en praktisk funktion och göra vår tjänst mer flexibel och användarvänlig i och med att den förbättrar mobiliteten i tjänsten. En utökning av applikationens funktioner är också planerad men detta kommer göras i mån av tid och det är inget som är tänkt att implementeras i ett tidigt skede.

Databas

Projektgruppen kommer att implementera en databas där alla packabler sparas och där de två produkterna - Android- och webbapplikationen - hämta packabler utifrån användarens input. Databasen kommer till en början innehålla ett begränsat antal packabler för att kunna testa funktionaliteten. I takt med att produkten utvecklas så utvecklas även databasen och den fylls på med fler packabler. Det finns inget krav på användarvänlighet när det gäller databasen då den inte syns för användaren. Däremot är det viktigt att databasens arkitektur är väl genomtänkt för att information ska kunna hämtas ut på ett snabbt och effektivt sätt. För mer detaljerad information om databasens struktur se ER-diagram med tillhörande förklaring nedan.

Systemdiagram



På klientsidan av systemet finns det två olika användargränssnitt, en hemsida som användaren interagerar med via en Android- eller webbapplikation. Användarens input som matas in via respektive gränssnitt skickas till en Python-applikation med hjälp av HTTP-anrop. Python-applikationen exekveras på en webbserver, i vårt fall pythonanywhere-servern. Baserad på användarens input hämtar applikationen information från tredjepartstjänster (bl.a. en väder-API) och kontaktar sedan en MySQL-databas som ligger på en databasserver inom pythonanywhere-miljön. Kommunikationen mellan applikation och databas sker via SQL-kommandon. Innehållet som har hämtats från databasen skickas tillbaka till klienten i json format och visas upp i användargränssnittet.

Pythonapplikationen på webbservern

Nedan följer en kort beskrivning av några av de mest väsentliga funktionerna i pythonapplikationen som exekveras på webbservern och som tar emot och skickar data till både webbgränssnittet och android-applikationen:

getGeolocation:

En funktion som konverterar destinationen som användaren har matat in till geografiska koordinater (längd- och breddgrad), vilket behövs för att unikt kunna identifiera destinationen. Dessa koordinater används sedan för att ta reda på aktuell väderdata respektive klimatzon enligt Köppen-Geigers klimatklassifikation (se nedan) samt för att direktvalidera användarens inmatning av destination i formuläret.

getWeatherForecast:

En funktion som skickar destinationens geografiska koordinater till en API (Open Weather Map) som i sin tur returnerar aktuell väderdata (minimal- och maximaltemperatur, genomsnittlig temperatur samt regnrisk för de kommande fem dagarna) för destinationen. Denna funktion anropas enbart om resan påbörjas inom de närmaste fem dagarna.

getClimateZone:

En funktion som skickar destinationens geografiska koordinater till en API som returnerar vilken klimatzon enligt Köppen-Geigers klimatklassifikation destinationen befinner sig i. Klassifikationssystemet utvecklades av klimatologerna Wladimir Köppen och Rudolf Geiger och delar in världen i fem huvudklimattyper samt ett antal undertyper vilket resulterar i cirka 30 olika klimatzoner [4]. Orter som befinner sig i samma klimatzon kan antas ha liknande genomsnittliga månatliga temperaturer och nederbörd. I projektgruppens databas har det skapats en relation som innehåller uppgifter om minimal-, maximal- och genomsnittstemperatur samt nederbördsrisk för de fyra årstiderna för varje klimatzon. Mer exakta resultat hade kunnat uppnås om man hade lagt in information för varje månad dock hade det varit betydligt mer tidskrävande.

getSeason

En funktion som beräknar vilken årstid det är på destinationen baserad på avresedatum. Funktionen anropas enbart om avresedatumet inte ligger inom de närmaste fem dagarna.

getHistoricWeatherData

Om avresedatumet ligger mer än fem dagar från aktuell datum baseras packlistan på historisk väderdata, dvs. information från Köppen-Geigers klimatklassifikation. Funktionen anropar *getClimateZone* och *getSeason* och hämtar sedan uppgifter om temperatur och nederbörd på destinationen från databasen.

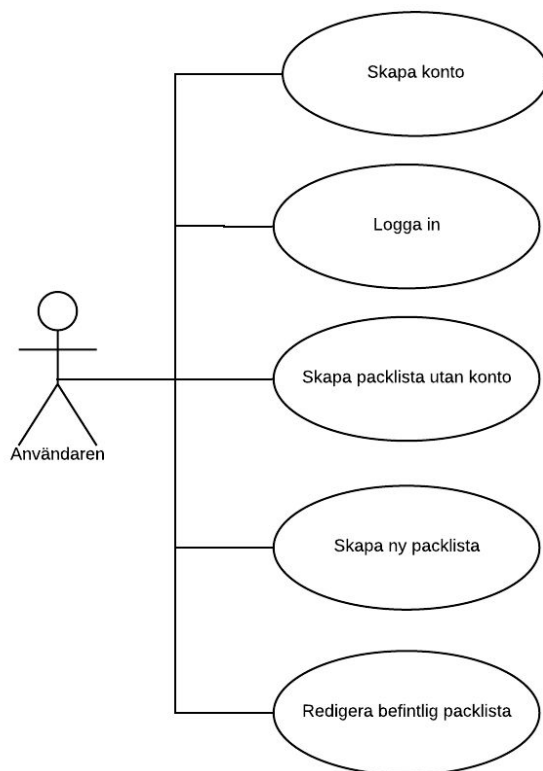
getItems

En funktion som hämtar packabler från databasen baserad på användarens input. Varje packabel i databasen har ett antal olika attribut, bl.a. vikt och kategori. Denna information skickas vidare till Android- respektive webbgränssnittet för att kunna presentera informationen på ett användarvänligt sätt, t.ex. genom att dela in packablerna i olika kategorier som exempelvis kläder, elektronik etc.

getTempItems

En funktion som hämtar temperaturberoende packabler från databasen. Alla packabler som påverkas av temperaturen på destinationen har samlats i en relation i databasen där de kopplas till en minimal och maximaltemperatur. Om medeltemperaturen på destinationen (hämtas med hjälp av funktionerna *getWeatherForecast* respektive *getHistoricWeatherData*) ligger inom detta intervall läggs packabeln till i packlistan.

Användningsfallsdiagram



Scenarion/Användningsfallsbeskrivningar

Någa av användningsfallsbeskrivningarna skrevs i ett tidigt skede av systemets utveckling. Användningsfallen *Registrera konto* och *Logga in* är fall som inte längre är relevanta för systemet eftersom registrerings- och inloggningsfunktion har bortprioriterats. Resten av fallen matchar systemets funktionalitet som det ser ut i den sista versionen.

Användningsfall	Registrera konto
Kortfattad beskrivning	Användaren registrerar ett konto genom att ange ett unikt användarnamn och skriva in lösenord två gånger.
Grundflöde	<ol style="list-style-type: none">1. Klickar på "Skapa konto"2. Fyller i lösenord två gånger3. Fyller i användarnamn4. Klickar på "Skapa konto/OK"
Alternativt flöde	Ogiltig inmatning som ger felmeddelande.
Speciella krav	Inga speciella krav definierade än

Förutsättningar	Användaren måste ha tillgång till en Androidtelefon med installerad applikation eller en enhet med webbläsare.
Post-conditions	Användaren får tillgång till applikationen.

Användningsfall	Logga in
Kortfattad beskrivning	Användaren loggar in i applikationen eller via webbläsaren genom att ange användarnamn och lösenord.
Grundflöde	<ol style="list-style-type: none"> 1. Skriver in användarnamn och lösenord 2. Trycker på enter eller klickar på "OK". 3. Användaren är inloggad och startsidan visas.
Alternativt flöde	Ogiltig inmatning som ger felmeddelande
Speciella krav	Inga speciella krav definierade än
Förutsättningar	Användaren har gått in på Backpackers hemsida eller öppnat applikationen och användaren har ett befintligt konto hos Backpacker.
Post-conditions	Användaren kopplas till applikationens startsida eller Backpackers startsida i webbläsaren.

Användningsfall	Skapa packlista utan konto
Kortfattad beskrivning	Användaren skapar en packlista baserad på specifika parametrar utan att registrera sig.
Grundflöde	<ol style="list-style-type: none"> 1. Klickar på "Skapa packlista utan konto" 2. Anger destination 3. Anger resans tidsintervall 4. Anger resekategori och adjektiv 5. Anger boende och färdmedel. Användaren kan lägga till medresenärer. 6. Anger packlistans titel. 7. Systemet genererar packlista som visas för användaren. Användaren kan lägga till och ta bort packabler ur packlistan. 8. Anger e-post och klickar på "Skicka" 9. Systemet skickar packlistan till mejladressen

Alternativt flöde	Ogiltig inmatning av mejladress som ger felmeddelande
Speciella krav	Inga speciella krav definierade än
Förutsättningar	Användaren har applikationen eller en enhet med en webbläsare
Post-conditions	En packlista skickas till användarens mejladress

Användningsfall	Skapa ny packlista
Kortfattad beskrivning	Användaren skapar en ny packlista i applikationen eller webbläsare.
Grundflöde	<ol style="list-style-type: none"> 1. Klickar på +-ikonen under rubriken <i>Mina packlistor</i> 2. Anger destination 3. Anger resans tidsintervall 4. Anger resekategori och adjektiv 5. Anger boende och färdmedel. Användaren kan lägga till medresenärer. 6. Anger packlistans titel 7. Systemet genererar packlista som visas för användaren. Användaren kan lägga till och ta bort packabler ur packlistan.
Alternativt flöde	Inget alternativt flöde definierat än
Speciella krav	Inga speciella krav definierade än
Förutsättningar	Användaren är inloggad via webbläsare eller mobiltelefon
Post-conditions	En ny packlista skapas

Användningsfall	Redigera befintlig packlista
Kortfattad beskrivning	Användaren avmarkerar kryssrutor i en befintlig packlista i applikationen och lägger till och tar bort packabler.
Grundflöde	<ol style="list-style-type: none"> 1. Klickar på packlistans titel under rubriken <i>Mina packlistor</i> 2. Redigerar, t.ex. lägger till eller tar bort packabler eller resenärer
Alternativt flöde	Inget alternativt flöde definierat än
Speciella krav	Inga speciella krav definierade än
Förutsättningar	Användaren är inloggad via webbläsare eller mobiltelefon. Användaren har en befintlig packlista.

Post-conditions

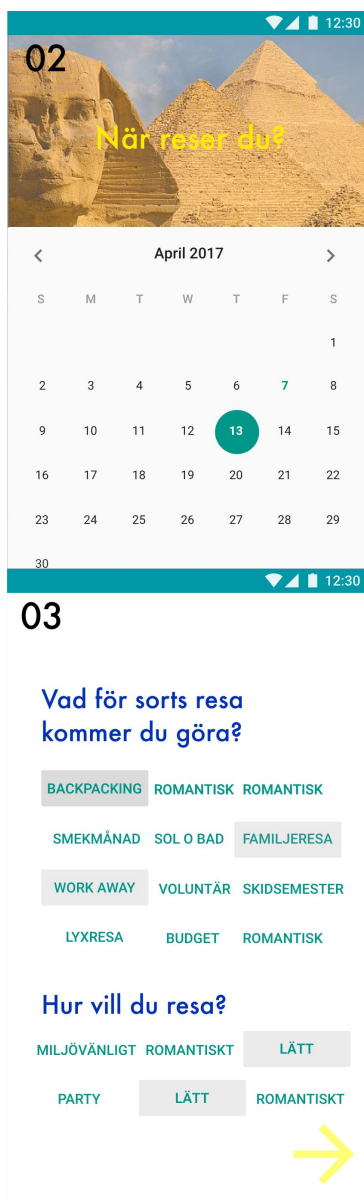
Packlistan har uppdaterats och redigerats

Användargränssnitt

För ytterligare information där de olika kraven tydliggjorts i mockups för Androidapplikationen, se länk:

<https://xd.adobe.com/view/56d932ae-4812-4752-7079-6dcb51df0d4a-20e8/>

Skisser till användargränssnittet i Androidapplikationen



Skiss 1 visar en vy över hur det kan se ut i applikationen när användaren väljer vilken tidsperiod resan ska äga rum vilket i sin tur ger systemet data för att kunna generera en relevant packlista med hänsyn till väderförhållanden.

Skiss 2 illustrerar hur användaren gör olika val som rör resan. I vyn ovan väljer användaren vilken sorts resa som ska göras och anger sedan en etikett för resan. Indata från de här valen avgör hur den föreslagna packlistan kommer att se ut.


Skiss till användargränssnittet i webbläsaren



Skiss 3 är en skiss över webbgränssnittets startsida. Här kan användaren välja om hen vill skapa en ny lista eller hämta en sparad lista.



Skiss 4 visar vyn dit användaren omdirigeras efter att ha klickat på "Skapa ny lista". Punkterna indikerar var i formuläret användaren befinner sig.

BAGPACKER  **OM OSS** **TIPS INFÖR RESAN** **KONTAKT** **LÄNKAR**

KLÄDER

Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>

NECESSÄR

Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>

SPECIALUTRUSTNING

Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>

ELEKTRONIK

Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>
Packable	<input checked="" type="checkbox"/>

CLASSIC 20
SPF 20
Medium Protection
Moisturizing with Vitamin E
Water resistant*

Skiss 5 visar den genererade packlistan där användaren kan välja att spara listan till datorn eller skriva ut den direkt.

Diagram/skiss

1. E/R-diagram

Nedanstående diagram illustrerar arkitekturen i databasen som används när projektgruppens applikation genererar packlistan. Rutorna med blå bakgrund har hunnits implementera i den slutgiltiga versionen av databasen, de andra relationerna har projektgruppen planerat för dock har dessa på grund av tidsbrist fått prioriteras bort.

I centrum står en tabell (relation) som innehåller samtliga packabler i databasen. Varje packabel har ett unikt id och ett antal olika attribut - bland annat vikt och kategori (t.ex. kläder, hygienartiklar etc.). De omkringliggande rutorna representerar de övriga tabellerna i databasen som var och en är en delmängd av tabellen i centrum (förutom *climate classification* som inte har någon direkt koppling till övriga relationer). Dessa tabeller innehåller sådana packabler som påverkas av användarens input, t.ex. föremål som enbart ska vara med på packlistan om temperaturen på destinationen ligger inom ett visst intervall eller ifall användaren planerar att utöva en viss aktivitet. Relationerna *activities*, *transport* och *accommodation* består i nuläget enbart av två kolumner, ett id som är främmande nyckel kopplat till *all items* relationen och typ av transport, boende respektive aktivitet. Alternativt hade denna information kunnat placeras i *all items* tabellen och på så sätt hade antalet anrop till databasen från python applikationen kunnat minskas. Då det dock i planeringsfasen var oklart ifall dessa relationer skulle innehålla fler attribut har projektgruppen skapat egna tabeller.

I *saved lists* sparas alla föremål som ingår i en specifik, användargenererad packlista med en slumpgenererad sexsiffrig kod som id. Till denna relation kopplas väderdatan som ligger till grund för användarens sparade lista.