

Course: Software Engineering, DAT255

Examinator: Jan-Philipp Steghöfer

Authors: Adrian Lindberg, Axel Ljungdahl, Erik Jansson, Jonatan Nylund, Jonathan Sundkvist, Sebastian Nilsson and Matthias Andersson

Prototype details

1 Design Rationale

The application design is based around one main **Activity** which holds the state of the app. The various views are loaded as **Fragments** into the main **Activity**. All communication between the different fragments goes through the main **Activity**. Various helper classes are responsible for extended functionality, e.g. directions, view management, event saving and others. For further documentation on the specific classes, see the project's *Javadoc* (available on the **master** branch in both comment and *HTML* form.)

1.1 Android API Level

Due to a lack of consideration for the user demographic, parts of the application was written using code that required at least level 21 of the Android API. Unfortunately, as a direct consequence, only around half of the current Android users (as of October 21, 2016) can use the application.¹

1.2 External Dependencies

1.2.1 Lombok

Lombok was used for **@Getter** and **@Setter** (for automatic creation of accessor and mutator methods, respectively), **@AllArgsConstructor** (for automatic creation of all-argument constructors for classes with many member variables) as well as **val** (used to shorten long lines of code by replacing the static type specifier).

1.2.2 Gson

After consideration of various different **JSON**-parsing libraries, **Gson** was chosen based on online benchmarks. As it is developed by Google, we believed it would be fully functional and well-tested.

1.2.3 Google Maps

The Google Maps API was used to display the map, event markers placed on it, corresponding info windows and functionality related to the map.

1.2.4 Google Directions

The Google Directions API was used to display the directions on the map. In its current state, the application only displays directions by “walking”, but can easily be extended to incorporate both driving, transit and cycling (all necessary code is present for this to be changed).

An important note is that the directions API requires an internet connection in order to display any directions on the map - as a result, a fallback has been implemented for users without an active internet connection, since this is likely a large part of the user base.

¹ <https://developer.android.com/about/dashboards/index.html> (2016-10-26)

1.3 Database Structure

The database is stored in a single *JSON* file, as an array of objects (unordered sets of name-value pairs), each with the following format:

Key	Value
id	The unique identifying non-negative integer of the event.
ownerID	The identifying non-negative integer tied to the event's organization. (Unused, but left in place in case the prototype is developed further.)
categories	An array of integers (0, 1, 2 or 3), each representing one of the four categories.
tags	An array of string tags, used when searching using the text-based search field.
latitude	A double-precision floating-point number representing the latitude of the event.
longitude	A string describing the time at which the event occurs.
dateInformation	An object with string keys, each representing a language (e.g. " en " for English, " sv " for Swedish and " ar " for Arabic) and tied to a string value containing the title of the event in the corresponding language. An English title is required; all others are optional.
titles	An object with string keys, each representing a language (e.g. " en " for English, " sv " for Swedish and " ar " for Arabic) and tied to a string value containing the title of the event in the corresponding language. An English title is required; all others are optional.
address	A string containing the address (in Swedish) at which the event is located.
contactInformation	A string containing various contact information (e.g. phone numbers, email

	addresses).
descriptions	An object with string keys, each representing a language (the exact same keys as for the “ titles ” object) and tied to a string value containing a description of the event in the corresponding language.
repeating	A boolean value (true or false) representing whether the event is a one-time occurrence or not (unused).
timeData	An array of five-element integer arrays - in order, the integer elements represent the week day (0 to 6), the starting hour (0 to 24), the starting minute (0 to 60), the ending hour (0 to 24) and the ending minute (0 to 60).

2 Overview

2.1 Behavioral overview

When starting, the application loads the map view (presented using the *Google Maps* API) and polls the phone for the current location. The map is then centered on the user’s position and events are presented to the user as markers on the map. The user also has the option to filter events by category via the bottom toolbar. When filtering by category, only one can be active at a time. All markers that do not belong to the chosen category are partially hidden (by lowering their opacities) in order to highlight the relevant events.

In the top right of the map and list views, there is an icon which switches to the list view, giving an overview of all events close to his or her current position. As in the map view, events can be filtered by any combination of category, tags and title in the list view.

In addition to the aforementioned filter options, it is also possible to filter events by distance or time.

By selecting an event in the map or list view, the details view of that event is displayed to the user, in which the user is presented with various types of information about the event, such as a description, time, address and contact information. In the details view, the user can also add or remove the event from their list of “favorites” or receive directional guidance.

By sliding from left to right or clicking on the icon in the upper-left part of the screen, the user reaches the navigation drawer, which holds information about the user’s “favorite” events as well as language settings.

2.2 Structural overview

As mentioned previously, the application uses a single **Activity** (named **MainActivity**) connecting three different **Fragments** (**MapFragment** for the map view, **ListFragment** for the list view and **DetailsFragment** for the details view). In addition to these classes, there are also several assisting classes. The classes have been divided into two basic packages - the **model** package contains all code unrelated to the “application” itself (i.e. the interface presented to the user), and the **app** package contains the rest.

The **model** package contains various classes, including **ArrayUtils** (which contains several convenience methods used when working with arrays), **DistanceCalculator** (which is used to calculate the great circle distance between two points on Earth), **EqualityChecker** (an alternative to **Comparator** with only two return values as opposed to three) and **Wrapper** (a generic class used to bypass Java’s rules related to the passing of arguments). It also has a subpackage **db**, which contains all database-related code - in the root, four interfaces **Database**, **Event**, **Filter** and **JSONTools** are placed; in the **impl** subpackage, the classes implementing the four interfaces are placed; and in the **sort** package, all interfaces and classes related to sorting lists of events are placed.

The **app** package is divided into two subpackages - **base** contains all code unrelated to the GUI portion of the application (such as the classes that initialize the database and a helper class for translating events), and **gui** contains the rest, including **MainActivity**, all **Fragment** classes and others.

2.3 Overview of online communications protocol

The communication between the application and the online database (hosted as a single file on Dropbox) is done in the form of a single download of the database file using the HTTPS protocol.

The very first time the application starts, as there is no offline copy of the database file, if the user does not have internet access, the application cannot run and therefore exits. Otherwise, the database file is downloaded from Dropbox and stored locally on the Android device. The remaining times (when there *is* a local version of the database file, even if it is an outdated one), the local database file will only be updated if the user has an active internet connection.

If some unforeseen communication error occurs when attempting to download from Dropbox, the application will simply behave as if there is no internet connection.

Summary	Issue key	Priority	Resolution
Som användare vill jag få en tydlig bild över detaljvyn för eventet för att lättare kunna ta till mig information om eventet	RM2-108	Highest	Done
Som användare vill jag absolut inte ha möjlighet att använda appen i "Landscape-läge"	RM2-111	Highest	Done
Alla beskrivningar i appen ska vara på ett annat språk.	RM2-134	Highest	Done
Alla labels i appen ska vara på ett annat språk.	RM2-133	Highest	Done
Som utvecklare vill jag ha ett interface som beskriver hur en aktivitet lagras i databasen för att kunna standardisera den information som kan förväntas av databasen.	RM2-31	Highest	Done
Som nyanländ vill jag kunna se en markör som representerar en aktivitet på en karta för att kunna se att aktiviteten finns och var den finns	RM2-5	Highest	Done
Som nyanländ vill jag kunna översätta detaljbeskrivningen av en aktivitet till mitt språk	RM2-127	High	Done
Som nyanländ vill jag kunna se aktiviteter i en lista.	RM2-49	High	Done
Som nyanländ vill jag kunna se aktiviteter laddade från en lokal databas på kartan så jag kan se vart de ligger.	RM2-50	High	Done
Som nyanländ vill jag kunna läsa om en aktivitet i detalj, så att jag kan få tag i bild, adress, beskrivning, och telefonnummer.	RM2-13	High	Done
Som nyanländ vill jag kunna byta mellan en kart- och listvy, så att jag kan navigera mellan kart- och listvy	RM2-29	High	Done
Som nyanländ vill jag kunna se (fake-) aktiviteter i en lista som ett alternativ till att se dessa i kartan	RM2-9	High	Done
Skapa JUnit-tester för databaskod	RM2-76	Medium	Done
Fallback för om GPS:en inte fungerar	RM2-135	Medium	Done
Som användare vill jag se en splash screen när appen startar	RM2-131	Medium	Done
Bottom toolbar remake	RM2-129	Medium	Done
Som användare vill jag kunna använda applikationen offline om jag har en lokal kopia av databasen	RM2-120	Medium	Done
Som användare vill jag se vilka aktiviteter som ligger närmast mig i tid.	RM2-138	Medium	Done
Som utvecklare vill jag rensa upp "spara till favoriter"-funktionalitet från Main-aktiviteten	RM2-124	Medium	Done
Som nyanländ vill jag se samma färg för kategorierna i detaljvyn	RM2-126	Medium	Done
Som nyanländ vill jag använda en snygg sidomeny	RM2-130	Medium	Done
Gör sökning (speciellt på distans) mer effektiv (huvudsakligen på listor)	RM2-105	Medium	Done
Förenkla lagring och sortering av Event-instanser (ta bort EventCollection, etc.)	RM2-121	Medium	Done
Som användare vill jag att detaljvyn visar faktiska event och inte dummyevent	RM2-117	Medium	Done
Bugfix: Vägbeskrivningen ska zooma till nuvarande position samt att vägen ska visas med en rött streck	RM2-113	Medium	Done
Som nyanländ vill jag se vägbeskrivningen till en aktivitet via gång	RM2-122	Medium	Done
Som användare vill jag enkelt kunna visuellt identifiera vilken kategori en aktivitet tillhör (via färg) i kartvyn	RM2-116	Medium	Done
Som nyanländ vill jag använda mig av "centrera mig på kartan"-knappen som Google Maps tillhandahåller	RM2-132	Medium	Done
Bugfix: Fixa så att "byta-vy-knappen" togglas vid back-button-tryckning	RM2-112	Medium	Done
Fixa bakåtknappen: Så att den växlar till där användaren var senast	RM2-125	Medium	Done
Ändra färger i listvyn till att matcha kategoriernas ikoner	RM2-128	Medium	Done
Som användare vill jag kunna se en aktivitets datum och tid	RM2-137	Medium	Done
Som användare vill jag, om möjligt, kunna läsa flera olika manuella översättningar av en aktivitets beskrivning	RM2-136	Medium	Done
Ta bort kategorinamn från databasen	RM2-109	Medium	Done
Som utvecklare vill jag använda mig av en adapter klass istället för implementera massor av interfaces	RM2-86	Medium	Done
Som nyanländ vill jag se vägbeskrivningen till en aktivitet via lokaltrafik	RM2-123	Medium	Done
Som utvecklare vill jag ha koden logiskt och strukturerat inlagt i paket!	RM2-114	Medium	Done
Som användare vill jag se information om avstånd och tid till eventet	RM2-119	Medium	Done
Som utvecklare vill jag ändra strukturen på interfacen till GMapFragment	RM2-118	Medium	Done
Fixa vytoggleknappen så att den inte hackar upp sig med detaljvyn	RM2-104	Medium	Done
Fånga bakåtknaptryck vid detaljvyn och listvyn	RM2-103	Medium	Done
Som nyanländ vill jag kunna se aktiviteter hämtade från internet	RM2-84	Medium	Done
Som nyanländ vill jag se vilka aktiviteter jag sparar.	RM2-60	Medium	Done
Som nyanländ vill jag använda den inbyggda bakåtknappen för att backa bland vyerna i appen	RM2-96	Medium	Done
Bugfix: Dölj vybytesknappen i detaljvyn	RM2-95	Medium	Done
Som utvecklare vill jag refactorera MainActivity för att få aktiviteten mindre och mer läsbar	RM2-107	Medium	Done
Som nyanländ vill jag kunna filtrera baserat på kategorier	RM2-65	Medium	Done
Som nyanländ vill jag komma åt mina sparade aktiviteter från ett menyalternativ	RM2-90	Medium	Done
Som nyanländ vill jag kunna filtrera aktiviteter baserat på avstånd	RM2-2	Medium	Done
Implementera som ett grafiskt sorteringsalternativ	RM2-94	Medium	Done
Som utvecklare vill jag refactorera GMapFragment för att få fragmentet mindre	RM2-87	Medium	Done
Lägg till metoder för att hämta aktiviteter via deras ID-nummer	RM2-106	Medium	Done
Implementera sortering av aktiviteter baserat på distans (i 'databasdelen')	RM2-61	Medium	Done
Dölj knappen	RM2-102	Medium	Done
Som nyanländ vill jag navigera mig runt i appen med hjälp av en profilmeny	RM2-91	Medium	Done
Som användare vill jag inte behöva ta bort en gammal sökning för att göra en ny	RM2-100	Medium	Done
Separera sortering från EventCollection-implementationerna för att undvika rörighet	RM2-98	Medium	Done
Ändra filter så att kriterier staplas samt versaler/gemener är obetydliga	RM2-99	Medium	Done
Som användare vill jag enkelt kunna sortera aktiviteternas kategorier via färger	RM2-89	Medium	Done
Som nyanländ vill jag kunna byta till arabisk text på knappar och menyer	RM2-59	Medium	Done
Bugfix: Sökfunktionen målas inte upp rätt i listvyn	RM2-92	Medium	Done
Knapp för att visa på kartan	RM2-75	Medium	Done
flytta "kameran" till platsen	RM2-73	Medium	Done
Skicka med plats till detaljvyn	RM2-72	Medium	Done
Som nyanländ vill jag kunna klicka på en "visa på karta"-knapp för aktiviteterna i detaljvyn	RM2-17	Medium	Done
Som nyanländ vill jag att kartan inte skall laddas om när jag byter vy.	RM2-56	Medium	Done
Optimera kartans livscykel så att den ej förstörs vid byte av vy.	RM2-77	Medium	Done
Som nyanländ vill jag kunna sortera aktiviteter baserat på namn (i bokstavsordning, från båda ändar)	RM2-83	Medium	Done
Implementera sparning av aktiviteter i en lokal databas	RM2-62	Medium	Done
Som nyanländ vill jag kunna spara en aktivitet till senare	RM2-12	Medium	Done
Som nyanländ vill jag kunna filtrera baserat på namn	RM2-16	Medium	Done

Som nyanländ vill jag kunna filtrera baserat på taggar	RM2-67	Medium	Done
Som nyanländ vill jag se hur långt jag har till en vald aktivitet	RM2-3	Medium	Done
Lägg till en singleton-klass som länkar "databasdelen" (i 'model.db'-paketet) till "appdelen"	RM2-85	Medium	Done
Som nyanländ vill jag kunna bli guidad av appen till en vald aktivitet	RM2-10	Medium	Done
Lägg till hämtning av aktiviteter via index samt sökning i 'EventCollection' och dess implementationer	RM2-82	Medium	Done
Hämta in databasen via .json filer	RM2-80	Medium	Done
Skapa instans av EventList och iterera igenom hela listan av events. Kalla på motsvarande Placemark metod	RM2-79	Medium	Done
Skapa instans av Database & kalla på getAllEvents	RM2-78	Medium	Done
Implementera filtrering baserat på namn och taggar gemensamt	RM2-70	Medium	Done
Implementera filtrering baserat på taggar i databasen	RM2-69	Medium	Done
Kartan ska ej centreras varje gång telefonen skickar gps-signal	RM2-55	Medium	Done
Som nyanländ vill jag klicka på en aktivitet i listan för att se dess detaljvy för att få ytterligare information om aktiviteten.	RM2-46	Medium	Done
Kryssa ner/tillbaka för att komma tillbaka till karta/lista	RM2-28	Medium	Done
Som nyanländ vill jag klicka på en markör för att se aktivitetens namn och adress och få övergripande info om aktivitet.	RM2-41	Medium	Done
Skapa en detaljvy för aktivitet	RM2-33	Medium	Done
Visa markörer i olika färger / med olika ikoner	RM2-22	Medium	Done
Skapa snygg knapp	RM2-48	Medium	Done
Skapa ful knapp på första bästa ställe för vybyten	RM2-47	Medium	Done
Som nyanländ vill jag att kartan visar min nuvarande position med en markör, så att jag lättare kan hitta till aktiviteter.	RM2-43	Medium	Done
List-värden ska innehålla hårdkodade aktiviteter	RM2-38	Medium	Done
Som nyanländ vill jag se en karta för att kunna lokalisera mig.	RM2-44	Medium	Done
Visa markörer på kartan	RM2-21	Medium	Done
Skapa ListView med 1-line dummy värden	RM2-37	Medium	Done
Skapa ListView fragment	RM2-36	Medium	Done
Som nyanländ vill jag dela med mig av en aktivitet till en vän	RM2-11	Low	
Som nyanländ vill jag kunna betygsätta en aktivitet	RM2-8	Low	
Som användare vill jag kunna rapportera en aktivitet/organisation	RM2-18	Low	
Som företag vill jag kunna logga in, via en webbvvy, för att kunna redigera mina uppgifter	RM2-19	Low	
Som företag vill jag kunna lägga till mina aktiviteter i databasen via en webbvvy så att de syns på kartan	RM2-1	Low	
Som företag vill jag kunna be om hjälp från integrerade svenskar	RM2-14	Low	Done