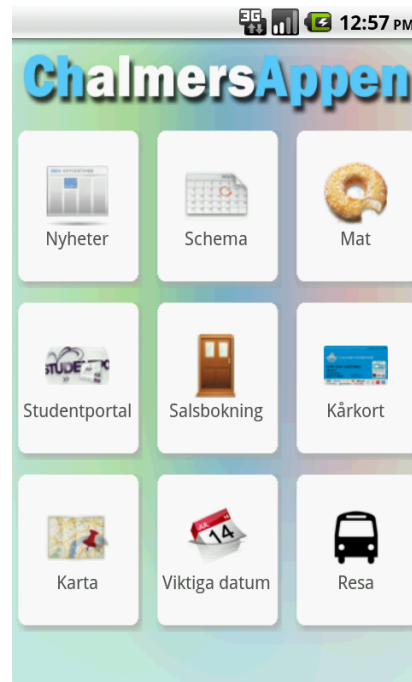


CHALMERS



ChalmersAppen

- en Android-baserad mobilapplikation för studenter vid Chalmers tekniska högskola

Kandidatarbete vid Data- och informationsteknik

KENAN BABIC
MIKAEL FAGERSTRÖM
MICHAEL JASINSKI

KAJSA LINDBERG
MICHAEL RYBACKI
DANIEL TELL

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2012
Kandidatarbete/rapport nr 2012:38

Förord

Rapporten är ett kandidatarbete vid Institutionen för Data- och informationsteknik, Chalmers tekniska högskola, och utfördes våren 2012. Den beskriver utvecklingen av ChalmersAppen, en applikation för smarttelefoner med operativsystemet Android.

Projektgruppen önskar tacka Christer Carlsson för den tid han har spenderat på att handleda kandidatarbetet i rätt riktning, Fia Börjeson för handledning med rapporten och även iChalmers för att de har bidragit med tips och råd.

Abstract

This thesis describes the development of ChalmersAppen, an application for smartphones on the Android operating system. Students at Chalmers University of Technology need access to several websites on the Internet. Given the increase of smartphones among students, the efficiency of their studies may increase with the help of an application that gathers the most relevant information from the most widely used websites, in one place. The goal of the application was to accomplish this task and be user-friendly.

As a basis for the development of the application a survey was handed out to the students. The results of the survey showed that there was an interest for such an application and what features that were of the highest importance. The results gave some directions for the design and implementation of a first version of the application.

The bachelor's thesis further describes the implementation of the features and the result of the application. The complexity factor of the implementation varied and due to some adversities some of the features could not be realized. One of the secondary goals of the bachelor's thesis was to hand over the application to a third party to develop it further and eventually release it on Google Play.

Sammanfattning

Denna rapport beskriver utvecklingen av ChalmersAppen, en applikation för smarttelefoner med operativsystemet Android. Studenter på Chalmers tekniska högskola har behov av att nå flera olika webbsidor på Internet. Med tanke på att smarttelefoner ökar bland studenter kan studietiden underlättas med hjälp av en applikation som samlar den viktigaste informationen från de mest använda webbsidorna på ett ställe. Målet med ChalmersAppen var att den skulle utföra denna uppgift och vara användarvänlig.

Som underlag till framtagandet av applikationen genomfördes en enkätundersökning bland studenterna. Undersökningen visade att det fanns intresse för en sådan applikation och vilka funktioner som var viktigast att ha med. Resultatet användes som riktlinjer för design och implementation av en första version av applikationen.

Kandidatrapporten beskriver vidare hur implementationen av funktionerna har gått till samt hur resultatet av applikationen blev. Svårighetsgraden på implementationen varierade och på grund av en del motgångar kunde ett fåtal funktioner inte slutföras på det sätt som önskades. Ett av delsyftena med kandidatarbetet var att applikationen skulle lämnas över till tredje part för vidareutveckling samt eventuellt lanseras på Google Play.

Ordlista

Activity	En aktivitet, en Android-klass som styr ett fönster med ett användargränssnitt. En aktivitet täcker oftast hela skärmen.
ADT	<i>Android Development Tool</i> , en uppsättning verktyg för utveckling av Android-applikationer.
Android MapView Balloons	Ett bibliotek som används tillsammans med Google Maps API för att visa informationsbubblor på kartor.
Android SDK	<i>Android Software Development Kit</i> , en samling utvecklingsverktyg som möjliggör och underlättar mjukvaruutveckling för Android, med bland annat bibliotek och en emulator.
API	<i>Application Programming Interface</i> , ett gränssnitt som specificerar på vilket sätt kommunikation ska ske mot en viss programvara.
Application context	En abstrakt Android-klass som ger tillgång till applikationsspecifika resurser och klasser.
Attribut	Används i layout-dokument för att definiera grafiska element.
AVD	<i>Android Virtual Device</i> , en emulator som används för att testa applikationer.
Broadcast Receivers	En Android-klass som använts för att ta emot Intent-objekt (se Intent) som sänds ut av andra applikationer eller av operativsystemet.
CID	Chalmers ID, ett unikt användarnamn som används på Chalmers.
Content Providers	Ett gränssnitt i Android som används för att dela data mellan olika applikationer.
Dalvik	Den virtuella maskinen i Android.
DDMS	<i>Dalvik Debug Monitor Server</i> , ett avlusningsverktyg som medföljer Android SDK (se Android SDK).
DOM	<i>Document Object Model</i> , en teknik som används vid parsning av XML-kod och som representerar objekten i form av ett DOM-träd.
Eclipse	En mjukvaruutvecklingsmiljö.
Google Maps API	Ett API som möjliggör för utvecklare att använda Googles implementation av kartor, Google Maps, i deras applikationer och webbsidor.
Google Play	En webbsida som används i distributionen av Android-applikationer, tidigare känt som Android Market.

Gson	Ett bibliotek som används till att parse JSON (se JSON).
HTML	<i>HyperText Markup Language</i> , ett märkspråk som används för att bygga webbsidor.
http	<i>HyperText Transfer Protocol</i> , ett kommunikationsprotokoll som används på Internet, främst för att överföra data mellan webbsidor och klienter.
Intent	Ett Android-objekt med beskrivning av en operation som ska genomföras. Intent-objekt används ofta för att starta aktiviteter.
JSON	<i>JavaScript Object Notation</i> , ett dataöverföringsformat.
Jsoup	Ett bibliotek som används till att parse HTML (se HTML).
Källkod	Programkod skriven i ett märk- eller programmeringsspråk.
Kategori	En uppsättning av en eller fler funktioner som kan nås från huvudmenyn i ChalmersAppen.
Layout-dokument	Ett XML-dokument i Android där grafiskt användargränssnitt kan definieras.
Manifest	En fil vid namn AndroidManifest.xml som måste deklarerar för varje Android-applikation och som innehåller grundläggande information om applikationen och dess aktiviteter.
Parsning	Tolkande av en teckensträng enligt en formell grammatik.
QWERTY	Det vanligaste knappupplägget på tangentbord. Namnet kommer av de sex första bokstäverna på ett sådant tangentbord.
Ramverk	En grund för uppbyggnad av mjukvara, som till exempel en samling färdiga klasser och metoder som kan vidareutvecklas.
REST	<i>Representational State Transfer</i> , en arkitekturstil som används vid utformandet av webbtjänster.
RSS	<i>Really Simple Syndication</i> , ett XML-baserat filformat som används för att publicera nyhetsflöden.
SAX	<i>Simple API for XML</i> , en mekanism som används för sekventiell parsning av XML-kod.
Service	En tjänst, en Android-klass utan ett grafiskt gränssnitt. En tjänst används för att exekvera funktionalitet i bakgrunden.
SQL	<i>Structured Query Language</i> , ett programmeringsspråk som används för att interagera med relationsdatabaser.

SQLite	Ett databashanteringssystem som används i Android och som har implementerats i form av ett C-bibliotek med samma namn.
Stack	En mjukvarulösning bestående av flera abstraktionslager.
Subclipse	En insticksmodul till Eclipse som ger stöd för Subversion (SVN).
SVN	Subversion, ett versionshanteringssystem.
UML	<i>Unified Modelling Language</i> , ett objektorienterat språk använt för att modellera mjukvarusystem.
URI	<i>Uniform Resource Identifier</i> , en sträng med referens till en resurs, som till exempel en webbadress.
XML	<i>Extensible Markup Language</i> , ett märkspråk som bland annat används för att lagra data.

Innehållsförteckning

1. Inledning	1
1.1. Syfte.....	2
1.2. Utmaningar	2
1.3. Rapportens upplägg	2
2. Teknisk bakgrund om Android och applikationsutveckling.....	3
2.1. Smarttelefoner och Android	3
2.1.1. Smarttelefoner	3
2.1.2. Operativsystemet Android	5
2.1.3. Android-versioner	6
2.2. Utveckla Android-applikationer: Verktyg	7
2.2.1. Android SDK	7
2.2.2. AVD	7
2.2.3. Manifest	8
2.2.4. DDMS	8
2.3. Utveckla Android-applikationer: Huvudkomponenter	8
2.3.1. Activities	8
2.3.2. Services	9
2.3.3. Broadcast Receivers	9
2.3.4. Content Providers.....	9
2.3.5. Intents.....	10
2.3.6. Grafiskt användargränssnitt	10
3. Metod och verktyg.....	11
3.1. Enkätundersökning	11
3.2. Utvecklingsmiljö och versionshanteringssystem.....	11
3.3. Programspråk och format	11
3.4. Använda bibliotek.....	12
4. Kravanalys och design	14
4.1. Plattformsversion.....	14
4.2. Fastställande av funktioner och kategorier	14
4.3. Beskrivning av fastställda kategorier	15

4.3.1. Kategori 1, 'Nyheter'	15
4.3.2. Kategori 2, 'Schema'	15
4.3.3. Kategori 3, 'Mat'	15
4.3.4. Kategori 4, 'Studentportal'	15
4.3.5. Kategori 5, 'Salsbokning'	15
4.3.6. Kategori 6, 'Kårkort'	16
4.3.7. Kategori 7, 'Karta'	16
4.3.8. Kategori 8, 'Viktiga datum'	16
4.3.9. Kategori 9, 'Resa'	16
4.4. Kravspecifikation.....	16
4.4.1. Målsättning	16
4.4.2. Funktionella krav	16
4.4.3. Icke-funktionella krav	17
4.4.4. Programmeringsregler.....	17
4.5. Grafisk design.....	17
5. Implementation av kategorier och funktioner.....	19
5.1. Huvudmenyn	19
5.2. Nyheter	19
5.2.1. Parsning av nyhetsflödet.....	20
5.2.2. Parsning av artiklar	20
5.3. Schema.....	21
5.4. Mat.....	21
5.5. Studentportal.....	23
5.6. Salsbokning	23
5.7. Kårkort.....	24
5.8. Karta	24
5.8.1. Sökfunktionen	25
5.8.2. Databasen	26
5.9. Viktiga datum	26
5.9.1. Notifikationer.....	27
5.10. Resa	27
5.11. Menyer.....	28
5.11.1. Om.....	28

5.11.2. Inställningar	28
6. Resultat.....	29
6.1. Nyheter	29
6.2. Schema.....	30
6.3. Mat.....	30
6.4. Studentportal.....	32
6.5. Salsbokning	32
6.6. Kårkort.....	33
6.7. Karta	33
6.8. Viktiga datum	35
6.9. Resa	36
6.10. Notifikationer.....	37
6.11. Den utfällbara menyn	37
7. Diskussion	39
7.1. Problem vid implementation	39
7.2. Förarbete	40
7.3. Relaterade arbeten	40
7.4. Framtida arbete	40
7.4.1. Förslag på utökad funktionalitet	41
8. Slutsats.....	42
Källförteckning.....	43
Appendix	I
A. Enkätundersökning	I
A1. Enkät	I
A2. Resultat av enkätundersökning	II
B. Tänkt funktionalitet.....	III
C. Kravspecifikation	IV

1. Inledning

Mobilanvändandet ökar hela tiden, både för mobila tjänster och mobil data (Gustafsson 2011). Den sista juni 2011 fanns det 13,1 miljoner abonnemang och aktiva kontantkort till mobiltelefoni i Sverige, varav 3,24 miljoner av abonnemangen användes till smarttelefoner.

Smarttelefoner är mer avancerade än vanliga mobiltelefoner. De har bland annat pekskärm och möjlighet till Internetuppkoppling, vilket gör att användarna kan göra sina smarttelefoner mer personliga. Då det finns ett brett utbud av applikationer att ladda ner kan varje telefon anpassas till en användares personliga behov.

Smarttelefoner är mer avancerade än vanliga mobiltelefoner. De har bland annat pekskärm och möjlighet till Internetuppkoppling, vilket gör att användarna kan göra sina smarttelefoner mer personliga. Då det finns ett brett utbud av applikationer att ladda ner kan varje telefon anpassas till en användares personliga behov.

Googles Android är det operativsystem till smarttelefoner som ökar mest, både i världen (Gartner 2012) och i Sverige (StatCounter 2012a) (StatCounter 2012b). Operativsystemet är inte begränsat till att köras på en specifik hårdvaruplattform. Istället har utvecklarna valt att göra det flexibelt så att flera mobiltillverkare kan använda Android till sina smarttelefoner. Detta skiljer sig från exempelvis iOS, Apples operativsystem, som enbart är till för deras egna produkter som till exempel iPhone (Nations 2012).

Android tillhandahåller en öppen plattform för applikationsutveckling både för personligt och kommersiellt bruk (Android 2012a). Genom att ladda upp applikationerna på en specifik marknad, till exempel Google Play, kan andra användare få tillgång till dessa.

Då antalet smarttelefoner ökar finns det potential för applikationer som kan underlätta för studenter genom att samla den viktigaste informationen från de mest använda webbsidorna på ett ställe. Studenter på Chalmers tekniska högskola har dagligen behov av att nå flera olika webbsidor på Internet. Det finns en webbsida för att se scheman, en annan för att boka grupprum, en tredje för att se kurser och så vidare. Då flera av dessa webbsidor är anpassade för datorskärmar medför detta att studenterna dagligen behöver tillgång till en dator. Utbudet av datorer och datorsalar på Chalmers är begränsat, och då troligtvis inte alla studenter har med sig en bärbar dator medför detta att de inte alltid kan komma åt den information som de behöver.

1. Inledning

1.1. Syfte

Syftet med kandidatarbetet är att designa och implementera en första version av en lättanvänd applikation till Android-baserade smarttelefoner. Målet med applikationen är att underlätta studietiden för studenter vid Chalmers tekniska högskola genom att sammanställa viktig information. Vid kandidatarbetets slut ska applikationen kunna lämnas över till en tredje part för vidareutveckling.

1.2. Utmaningar

En av utmaningarna med ChalmersAppen var att få en tilltalande, lättöverskådlig och användarvänlig design. Den andra utmaningen var att få med de funktioner som var mest intressanta och användbara för en student på Chalmers. För att uppnå detta krävdes kunskap om design och användargränssnitt, programmeringskunskaper och vetskap om vad målgruppen var intresserade av. Vidare beskriver rapporten tillvägagångssättet för angripandet av dessa utmaningar.

1.3. Rapportens upplägg

Rapportens upplägg beskrivs övergripande nedan.

Rapportens kapitel 2 beskriver den tekniska bakgrund som ansetts vara relevant. Kapitel 2.1 beskriver övergripligt vad en smarttelefon är, hur operativsystemet Android är uppbyggt och vilka plattformsversioner som finns. Vidare beskriver kapitel 2.2 fyra verktyg som används vid utveckling av Android-applikationer. Efter detta förklaras i kapitel 2.3 sex vanliga huvudkomponenter som används vid Android-utveckling.

Från kapitel 3 till och med kapitel 5 koncentreras rapporten mer på genomförandet av utvecklingen av ChalmersAppen. Kapitel 3 beskriver de metoder och verktyg som har använts vid utvecklingen av ChalmersAppen, där kapitel 3.1 beskriver hur enkätundersökningen gick till och kapitel 3.2 beskriver vilken utvecklingsmiljö och vilket versionshanteringssystem som använts vid utvecklingen. Vidare i kapitel 3.3 och 3.4 följer en beskrivning på vilka språk, format och bibliotek som använts samt förklaring av vad dessa gör.

Kapitel 4 redogör för hur förarbetet gick till, från val av plattformsversion till upprättande av kravspecifikation och design av prototyper. Vidare beskriver kapitel 5 hur implementationen av alla funktionerna gick till.

Den avslutande delen i rapporten består av resultat, diskussion, och slutsats. Här visas resultatet av ChalmersAppen, diskussioner om hela utvecklingsprocessen och även kandidatarbetets slutsats.

2. Teknisk bakgrund om Android och applikationsutveckling

Utveckling av applikationer till Android-baserade smarttelefoner kräver viss teknisk bakgrund. Följande kapitel förklarar grundläggande fakta om smarttelefoner i allmänhet, operativsystemet Androids uppbyggnad samt verktyg och huvudkomponenter vid utveckling av applikationer.

2.1. Smarttelefoner och Android

För att få en bakgrundsförståelse för smarttelefoner och Androids operativsystem presenteras grundläggande fakta om dessa i följande avsnitt.

2.1.1. Smarttelefoner

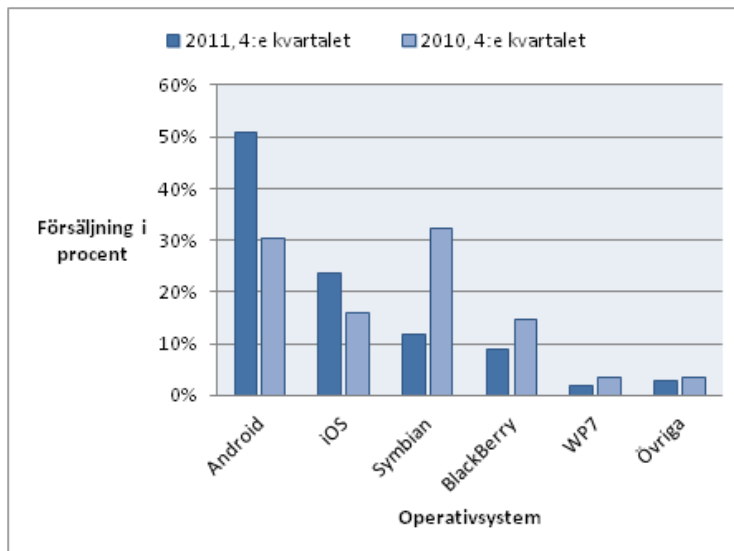
Det finns inte någon allmänt accepterad definition av vad en smarttelefon är, och de definitioner som finns förändras med tiden, men enligt About.com kan en smarttelefon förenklat beskrivas som en kombination av en handdator och en vanlig mobiltelefon (Cassavoy 2012a).

Utöver den grundläggande telefonfunktionaliteten har smarttelefoner ett operativsystem som möjliggör personlig anpassning (Cassavoy 2012b). Användaren kan själv välja vilka applikationer han eller hon vill ha. Vidare kan en smarttelefon koppla upp sig mot ett nätverk samt skicka och ta emot e-post (Cassavoy 2012b).

Likt handdatorer har smarttelefoner ett QWERTY-tangentbord (Cassavoy 2012b), det vill säga att tangenterna är placerade på samma sätt som på en vanlig dators tangentbord. Tangentbordet på smarttelefonen kan antingen presenteras för användaren med fysiska knappar eller virtuella knappar på skärmen (Cassavoy 2012b).

För att få en överblick på hur försäljningen ser ut för de olika operativsystemen i smarttelefoner i världen har företaget Gartner under flera år gjort undersökningar på detta (Gartner 2012). Figur 2.1 visar hur försäljningen av smarttelefoner, med avseende på operativsystem, såg ut under det fjärde kvartalet år 2011 jämfört med samma kvartal år 2010.

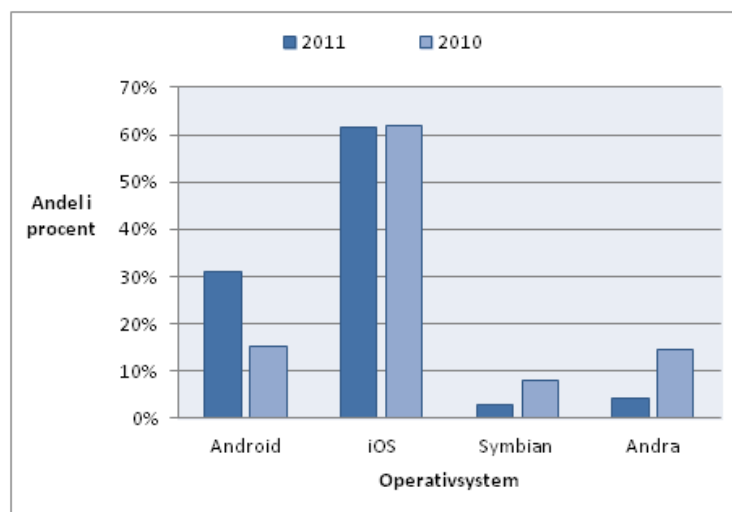
2. Teknisk bakgrund om Android och applikationsutveckling



Figur 2.1: Global försäljning av smarttelefoner till slutanvändare med avseende på operativsystem (Gartner 2012). Bilden visar jämförelse mellan fjärde kvartalet 2011 och fjärde kvartalet 2010 i procent.

Försäljningen av Android-baserade smarttelefoner ökade mest mellan åren 2010 och 2011. Största minskningen syntes hos Symbian, Nokias tidigare operativsystem.

I Sverige ser fördelningen av de olika mobila operativsystemen olika ut. Exempelvis ligger iOS långt för resten av operativsystemen, se figur 2.2.



Figur 2.2: Andel mobila operativsystem i Sverige (StatCounter 2012a) (StatCounter 2012b). Bilden visar fördelningen på de olika operativsystemen i procent och påvisar även skillnaden mellan år 2011 till 2010.

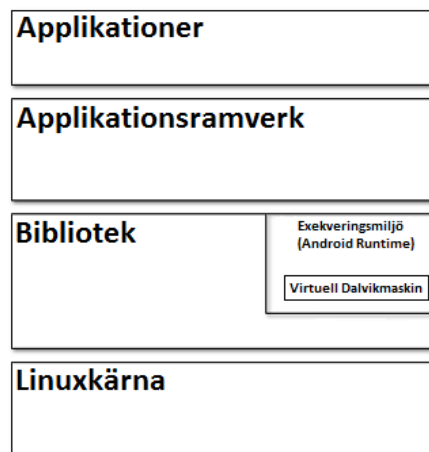
Även om iOS står för största delen av marknaden, har Android fördubblat sin ställning på ett år. Symbian och övriga operativsystem har minskat markant under samma period.

2. Teknisk bakgrund om Android och applikationsutveckling

2.1.2. Operativsystemet Android

Android är ett operativsystem som är anpassat till flera olika användningsområden. Det används till bland annat smarttelefoner, pekplattor och andra enheter (Android 2012b).

Operativsystemets uppbyggnad brukar ofta beskrivas som en stack, se figur 2.3, där varje nytt lager fungerar som en abstraktion av lagret nedanför (Zechner 2011). I botten av stacken ligger själva Linuxkärnan. Ovanpå kärnan finns ett antal grundläggande bibliotek, implementerade i lågnivåspråk, samt systemets exekveringsmiljö, *Android Runtime Environment*, med den virtuella maskinen Dalvik (Zechner 2011).



Figur 2.3: Bilden visar Androids stack: Operativsystemet Androids uppbyggnad från Linuxkärnan till Applikationer (Android Developers 2012a).

Lagret ovanför bibliotekslagret, applikationsramverket, består i sin tur av ett antal bibliotek implementerade i Java (Gargenta 2011a). Mycket av funktionaliteten som hanterar telefonens hårdvara är omslagsklasser i Java som anropar en underliggande lågnivåimplementation. Applikationer som utvecklas för Android arbetar mot applikationsramverket (Gargenta 2011a).

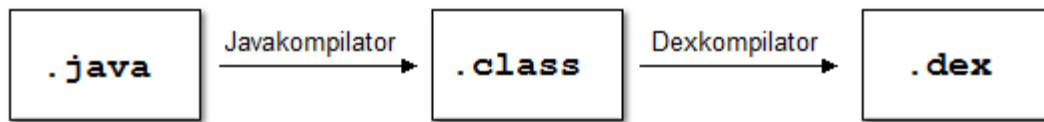
Linuxkärnan fungerar som ett abstraktionslager mellan hårdvaran och systemets underliggande systemtjänster (Gargenta 2011a). Detta ger systemet stor flexibilitet och tillåter att det körs på olika typer av enheter, vilket var huvudanledningen till att Linux valdes som grund för Android. Några andra anledningar till valet var att Linux är ett vältestat och relativt säkert system, med mycket dokumentation samt bra process- och minneshantering. Linux utvecklades även med öppen källkod, vilket också var ett av målen för Android (Gargenta 2011a).

Varje applikation som startas upp av användaren eller internt av systemet tilldelas ett eget Linux-användar-id och körs som en egen Linux-process (Android Developers 2012b). Varje sådan Linux-process körs i sin tur på en egen instans av Dalvik (Android Developers 2012a).

2. Teknisk bakgrund om Android och applikationsutveckling

Dalvik har utvecklats specifikt för mobiltelefoner och har specialanpassats så att varje Android-enhet effektivt kan köra flera instanser av Dalvik samtidigt (Conder 2010).

Dalvik exekverar filer i formatet *Dalvik Executables* med filändelsen *.dex* (Ehringer 2010). Kompileringsprocessen för Android sker i två övergripande steg, se figur 2.4.



Figur 2.4: Förenklad bild av hur kompileringsprocessen för Android går till, från *.java*-filer till *.dex*-filer.

Källkod skriven i Java kompileras först till vanliga Javaklass-filer (filändelsen *.class*), som därefter kompileras till *.dex*-filer (Ehringer 2010). Dessa *.dex*-filer, tillsammans med resurser, så som bilder, strängar och ljudfiler, paketeras därefter till en *Application Package*-fil med filändelsen *.apk* (Android Developers 2012c). Apk-filer är i själva verket en typ av arkivformatet *Zip*. Och det är dessa filer som sedan distribueras och installeras på Android-enheter (Murphy 2008) (Android Developers 2012c).

Android levereras med ett antal lågnivåbibliotek implementerade i C/C++ (Gargenta 2010). Bland dessa återfinns bland annat medie- och grafikbibliotek samt databashanteraren SQLite (Android Developers 2012a). I applikationsramverket finns motsvarigheterna till dessa bibliotek i form av omslagsklasser implementerade i Java (Gargenta 2011a). Applikationsutveckling till Android sker i huvudsak mot detta ramverk, (Zechner 2011) vilket medför att utvecklare endast behöver arbeta mot ett gränssnitt implementerat i Java.

Det översta lagret i Android-stacken innehåller alla applikationer som följer med telefonen, samt de som användaren väljer att installera i efterhand (Gargenta 2011a). Ett kännetecken för Androids API, *Application Programming Interface*, är att alla applikationer som redan är installerade på enheten kan dela ut information till nya applikationer med hjälp av komponenterna *Content Provider*, se vidare kapitel 4.4, och *Intent*, se vidare kapitel 4.5 (Android Developers 2012b). För användaren innebär detta till exempel att applikationer kan starta komponenter av andra applikationer utan att användaren anger detta explicit (Android Developers 2012d). Utvecklare får själva välja vad som får och vad som inte får delas ut (Gargenta 2011a).

2.1.3. Android-versioner

Sedan lanseringen av Android-plattformen har ett antal uppdateringar av systemet släppts. Dessa uppdateringar har åtgärdat buggar och möjliggjort ny funktionalitet (Android Developers 2012e) (Zechner 2011). För applikationsutvecklare har versionerna klassificerats i API-nivåer och större uppdateringar har namngivits efter desserter (Zechner 2011), se tabell 2.1.

2. Teknisk bakgrund om Android och applikationsutveckling

Tabell 2.1: Tabellen sammanställer Android-plattformens största leveransversioner (Android Developers 2012f).

Version	API nivå	Versionsnamn
1.0	1	-
1.1	2	-
1.5	3	Cupcake
1.6	4	Donut
2.0 - 2.1	5 - 7	Eclair
2.2	8	Froyo
2.3	9 - 10	Gingerbread
3.0 - 3.2	11 - 13	Honeycomb
4.0	14 - 15	Ice Cream Sandwich

Version 1.0 var den första kommersiella Android-versionen som släpptes (Android Developers 2012f). Vid lanseringen av *Donut* lades stöd till för olika skärmstorlekar och vidare för *Eclair* kom multitouch-funktionalitet (Zechner 2011). Multitouch innebär att skärmen kan känna av beröring för tre eller fler skärmpunkter samtidigt (Touch Topics 2012). *Honeycomb* introducerade nya användargränssnitt speciellt anpassade för pekplattor och *Ice Cream Sandwich* införde dessa ändringar för smarttelefoner och mindre skärmstorlekar (Android Developers 2012g).

2.2. Utveckla Android-applikationer: Verktyg

För att utveckla applikationer till Android krävs det kunskap om vissa tekniker och utvecklingssystem. I följande avsnitt förklaras en del av dem.

2.2.1. Android SDK

Android SDK, *Android Software Development Kit*, är en uppsättning av utvecklingsverktyg som möjliggör framtagande av applikationer till Android-baserade enheter (Android Developers 2012h). Bland dessa verktyg finns samtliga Android- och Google-API:er som innehåller det som behövs för utveckling mot de olika Android-versionerna (Android Developers 2012a).

2.2.2. AVD

Ett tillägg till SDK:n är AVD:n, *Android Virtual Device*, vilket är en virtuell enhet som emulerar en verklig Android-enhet (Android Developers 2012i). Med detta verktyg kan man utveckla applikationer utan att behöva ha tillgång till en fysisk enhet. En viktig detalj med den virtuella enheten är att den är helt anpassbar, programmeraren kan själv bestämma vilka specifikationer som hans eller hennes virtuella enhet ska ha (Android Developers 2012i).

2. Teknisk bakgrund om Android och applikationsutveckling

2.2.3. Manifest

Vid utveckling av programvara till Android måste alla aktiviteter och krav deklarerats i en *AndroidManifest.xml*-fil (Android Developers 2012j). För att kunna köra en applikation skickar filen Manifest viktig information till systemet. Filen beskriver bland annat de olika delarna som en applikation består av samt vilka behörigheter som den kräver.

2.2.4. DDMS

DDMS, *Dalvik Debug Monitor Server*, är ett avlusningsverktyg som ingår i Androids SDK och det finns som insticksmodul i Eclipse (Android Developers 2012k). Med DDMS kan utvecklaren övervaka enskilda processer och avlusa dessa med hjälp av inbyggda funktioner (Android Developers 2012k).

Ibland behöver utvecklaren veta hur applikationen reagerar mot inkommande samtal och SMS, eller varför applikationen kraschar i vissa situationer. I alla dessa fall erbjuder DDMS effektiva funktioner som gör det lätt för utvecklaren att avlusa (Android Developers 2012k).

2.3. Utveckla Android-applikationer: Huvudkomponenter

Androids API tillhandahåller ett antal grundbyggstenar för att underlätta applikationsutvecklingen (Gargenta 2011b). Dessa byggstenar kan ses som de minsta komponenterna som behövs för att få en fungerande applikation. Komponenterna är underordnade en basklass, vid namn *Application Context*, vilken beskriver hela applikationens process och omgivning som dess komponenter körs i (Gargenta 2011b).

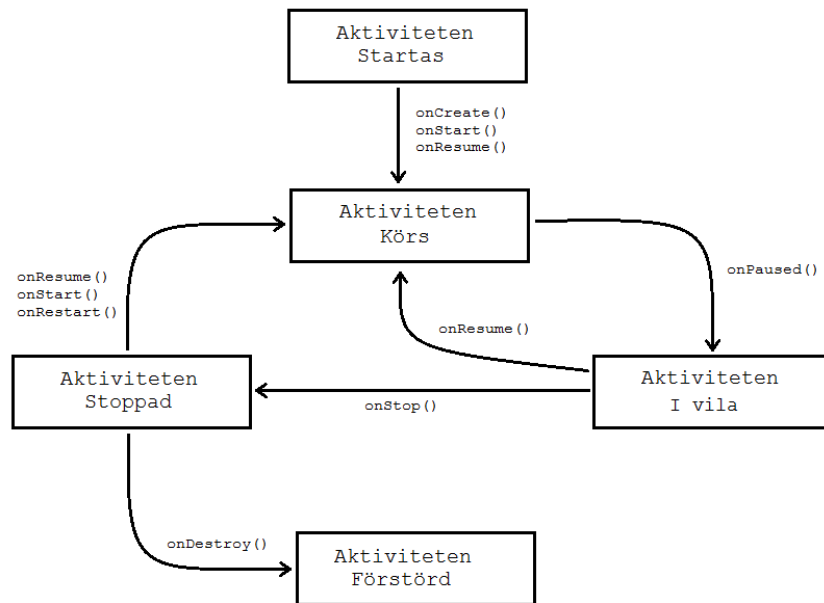
2.3.1. Activities

En aktivitet, *Activity*, är en grundläggande komponent i en Android-applikation (Android Developers 2012l). Vid implementation skapar utvecklaren normalt en ny klass som får ärva från Androids *Activity*-klass (Meier 2009). Därefter överskuggas ett antal ärvda metoder där aktiviteten initieras (Android Developers 2012l).

För att undvika slöseri av resurser vid start av aktiviteter, som är en dyrbar process, har olika tillstånd införts som aktiviteter kan befinna sig i (Gargenta 2011b). Dessa tillstånd hanteras av systemets aktivitetshanterare, och användaren har ingen kontroll över dem. Aktivitetshanteraren sköter därmed en aktivitets livscykel som består utav ett antal tillstånd som aktiviteten kan förflyttas till vid olika tillfällen (Gargenta 2011b).

När användaren startar en ny aktivitet, se figur 2.5, kommer den aktivitet som körs för tillfället att bli vilande och sedan stoppas, medan den nya aktiviteten får fokus "körs" (Gargenta 2011b). När en aktivitet är i tillståndet "stoppad" betyder det inte nödvändigtvis att den är inaktiv, utan endast att den inte syns på skärmen. En aktivitet som är i vila syns i bakgrunden medan fokus ligger på den aktiva.

2. Teknisk bakgrund om Android och applikationsutveckling



Figur 2.5: Bilden visar livscykeln för en aktivitet, från det att aktiviteten startas till dess att den förstörs (Gargenta 2011b).

Med hjälp av tillstånden “i vila” och “stoppad” kan användaren återgå till föregående aktivitet utan att behöva starta upp den på nytt (Gargenta 2011b).

Vid övergången mellan tillstånden anropas återkallningsmetoder såsom `onPause()` eller `onStop()` (Gargenta 2011b). Återkallningsmetoder kan överskuggas av utvecklaren ifall någon funktionalitet ska implementeras vid tillståndsövergången.

2.3.2. Services

En annan viktig grundkomponent i Android är *Service*, eller tjänst (Android Developers 2012b). Tjänster fungerar i princip på samma sätt som aktiviteter. Den väsentliga skillnaden är att tjänster inte kräver något grafiskt gränssnitt för att köras. Tjänster används till exempel för tillfällen då någonting ska exekveras i bakgrunden.

2.3.3. Broadcast Receivers

För att lyssna på olika händelser från operativsystemet eller från andra applikationer används *Broadcast Receivers* (Gargenta 2011b). Exempel på händelser kan vara att telefonen har startats upp eller att batteriet är lågt. Komponenten kan antingen deklarerars i applikationens manifest-fil eller direkt i Javakoden (Android Developers 2012m). Händelserna skapas genom att applikationer startar `Intents` (se vidare avsnitt 4.5).

2.3.4. Content Providers

Eftersom varje applikation körs som en egen Linux-process så måste det finnas ett sätt att dela data mellan dessa (Gargenta 2011b). *Content Providers* fungerar som ett gränssnitt för detta ändamål och exponerar en applikations data samt delar denna på säkert sätt mellan telefonens applikationer (Android Developers 2012n) (Gargenta 2011b).

2. Teknisk bakgrund om Android och applikationsutveckling

2.3.5. Intents

Activities, Services, Broadcast Receivers och *Content Providers* kommunicerar med hjälp av en Android-specifik komponent som kallas *Intent* (Gargenta 2011b). Objekt av klassen *Intent* skapas för att skicka meddelanden antingen inom en applikation eller mellan applikationer. Dessa meddelanden kan vara explicita eller implicita. Explicita meddelanden skickas till en välspecificerad mottagare och de kan till exempel begära att en specifik aktivitet ska startas. Implicita meddelanden talar om vad som ska utföras, men specificerar inte vilken del av systemet som ska hantera begäran (Gargenta 2011b).

2.3.6. Grafiskt användargränssnitt

Uppbyggnaden av ett grafiskt användargränssnitt kan i Android göras på två olika sätt (Gargenta 2011c). Grafiska komponenter kan deklarerars och definieras direkt i Java, mycket likt det arbetssätt som används vid implementation med Javas bibliotek *Swing* vid traditionell Javautveckling. Det andra alternativet är att separera källkoden som har med gränssnittet att göra från logiken som applikationen byggs upp av. Grafiska komponenter som till exempel knappar och textrutor deklarerars då som element i ett layout-dokument i XML-format och ges därefter attribut (Gargenta 2011c). Exempel på attribut som kan ges till en komponent kan vara storleken i form av höjd och bredd (Android Developers 2012o).

Vid kompilering genereras automatiskt en källkodsfil med namnet *R.java* (Android Developers 2012c). Filen kan ses som en koppling mellan alla resurser, vilket innefattar bland annat XML-filer, bilder samt strängar, och Javakod. *R.java* innehåller enbart pekare som refererar till filerna i resursmappen (Gargenta 2011d).

3. Metod och verktyg

Följande avsnitt beskriver de metoder och verktyg som tillämpats och använts vid utvecklingen av ChalmersAppen.

3.1. Enkätundersökning

För att få en överblick av vad studenter på Chalmers skulle vilja ha i en Chalmersrelaterad applikation genomfördes en enkätundersökning, med 33 slumpmässigt utvalda studenter på Chalmers, se appendix A1. Enkäten frågade om studenterna kunde tänka sig att ladda ner en Chalmersapplikation för Android om det funnits en sådan. Svarade en student ”ja”, fick studenten möjlighet att nämna tre funktioner som han eller hon skulle vilja ha med i applikationen. Ifall svaret blev ett ”nej”, fick studenten möjlighet att förklara varför. Enkätundersökningen sammanställdes och resultatet gav riktlinjer om hur planeringen med ChalmersAppens funktioner skulle läggas upp och prioriteras, se appendix A2.

3.2. Utvecklingsmiljö och versionshanteringssystem

Vid implementationen av ChalmersAppen användes Eclipse, med ett antal olika insticksmoduler, som är en rekommenderad utvecklingsmiljö för Android (Android Developers 2012p). En av insticksmodulerna som användes var ADT, *Android Development Tools*, som möjliggör applikationsutveckling för Android.

En annan insticksmodul som användes var Subclipse som integrerar versionshanteringssystemet SVN, *Subversion*, med Eclipse. SVN valdes tack vare att det erbjuds av Chalmers till studenter för att kunna utföra större programmeringsprojekt (Blåberg 2011). Versionshanteringssystem möjliggör implementation från flera datorer och sparar tidigare versioner av alla filer, historik om vem som ändrat i en fil och vad som ändrades (Pilato 2008).

3.3. Programspråk och format

Vid utvecklandet av ChalmersAppen användes ett flertal olika språk och format. Grundstenarna vid utveckling av Android-applikationer är det objektorienterade programmeringsspråket Java samt märkspråket XML (Android Developers 2012b). När applikationens funktioner implementerades, hanterades det ett antal andra språk och format. Bland språken fanns bland annat HTML och SQL och bland formaten JSON och RSS.

Ett märkspråk är ett språk som använder sig av beskrivningstaggarna för att hjälpa andra program att presentera vad programmet ska visa (Vidgen 2000). Utöver XML har märkspråket HTML använts i den mån att information har hämtats från webbsidor. Skillnaden mellan XML och HTML är att HTML används främst för att presentera information om webbsidor medan XML används till att lagra och överföra data (Vidgen 2000).

3. Metod och verktyg

För databasfunktionaliteten användes språket SQL, *Structured Query Language*. Detta språk är ett standardspråk för att skapa, hämta data från och ändra i relationsdatabaser (W3Schools 2012a).

Formatet JSON, *JavaScript Object Notation*, användes vid utvecklandet av ChalmersAppens kategori 'Resa'. JSON är ett textbaserat dataöverföringsformat och är språkoberoende, vilket betyder att det kan användas vid implementation vid alla typer av programmeringsspråk (JSON 2012). Formatet representerar primitiva och strukturerade datatyper som nyckel-värdepar (Crockford 2006).

Vid implementation av kategorierna 'Nyheter' och 'Mat' användes parsning av RSS, *Really Simple Syndication*. RSS är ett filformat baserat på XML (WhatIsRSS 2012). Formatet används av webbsidor för att leverera relevant information på ett snabbt och enkelt sätt till läsaren. Sedan skapandet av RSS har ett flertal olika versioner utvecklats. RSS-dokument innehåller ett antal standardelement, bland annat `title`-, `link`- och `description`-element (W3Schools 2012b). Bland dessa standardelement finns även `item`-element, som representerar enskilda RSS-artiklar som i sin tur innehåller egna `title`-, `link`- och `description`-element.

3.4. Använda bibliotek

Tre externa bibliotek användes vid utvecklingen av ChalmersAppen, *Jsoup*, *Gson* och *Android MapView Balloons*. Nedan följer en kort beskrivning av biblioteken samt i vilka kategorier de använts.

Jsoup är ett bibliotek som används för att hämta, parsa och bearbeta data i HTML-format (Hedley 2012). Detta bibliotek valdes för parsning i kategorierna 'Nyheter', 'Mat' och 'Viktiga Datum' eftersom det är implementerat i Java och det finns ett lätthanterligt API. På Jsoups webbsida finns även beskrivningar av ett flertal problem lösta med hjälp av biblioteket samt exempel på källkod, vilket möjliggör snabb inläring.

Vidare tillhandahåller *Jsoup* metoder för att hämta HTML-kod från en sträng, fil eller via en webbadress (Hedley 2012). Ytterligare metoder som erbjuds är att specifik information kan parsas ut genom att ange taggar eller attribut samt att specifika element, attribut och text kan manipuleras.

Gson är ett bibliotek som används till att omvandla Javaobjekt till och från JSON (Singh 2012). Till skillnad från det JSON-paket som ingår i standardbiblioteken i Android kan *Gson* returnera objekt av godtyckliga typer, vilket möjliggör ett helt objektorienterat tillvägagångssätt. Detta bibliotek användes vid implementationen av kategorin 'Resa'.

Android MapView Balloons är ett bibliotek som på ett enkelt sätt möjliggör användandet av informationsbubblor i implementationen av kartan (Gilfelt 2012). Informationsbubblor är informationsrutor som visas när användaren trycker på ett objekt på kartan. De finns med i de

3. Metod och verktyg

flesta implementationer av kartor av olika typer (bland annat i Google Maps och Eniro), men eftersom Google Maps API inte stödjer denna lösning som standard, användes ett externt bibliotek.

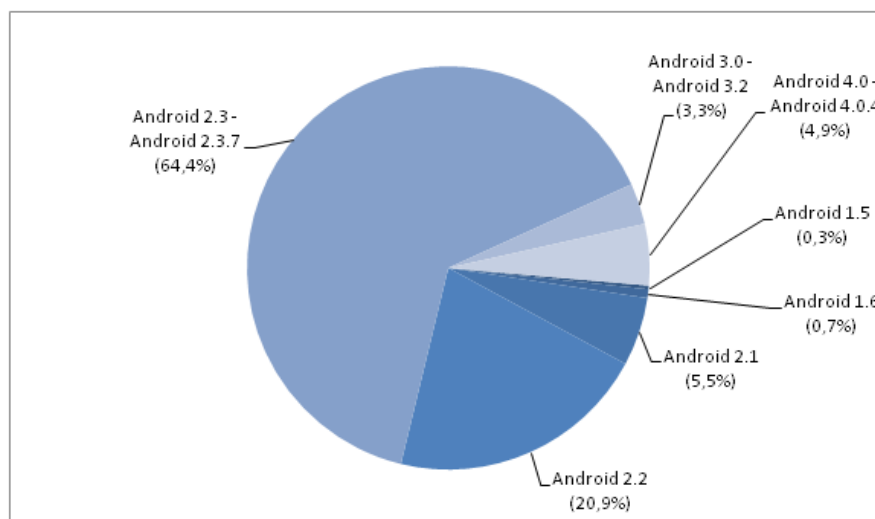
4. Kravanalys och design

Innan implementationen påbörjades av ChalmersAppen genomfördes ett grundligt förarbete. Beslut togs om vilken Android-version som skulle användas samt vilka funktioner som skulle vara med. Vidare upprättades prioriteringar av funktionerna och kategorierna i kravspecifikationen och en designprototyp togs fram. I följande avsnitt beskrivs tillvägagångssätten mer i detalj.

4.1. Plattformsversion

För att täcka större delen av alla Android-baserade telefoner utvecklades ChalmersAppen till plattformsversionen Android 2.1 (även kallad Eclair) med API-nivå 7 (Android Developers 2012q). Androids plattformsversioner är bakåtkompatibla, vilket innebär att applikationer som utvecklas för version 2.1 kan installeras på enheter med 2.1 eller högre.

I början av maj 2012 kunde applikationer, utvecklade för version 2.1, laddas ner och användas av 99 procent av alla användare med en Android-telefon (Android Developers 2012q), se figur 4.1.



Figur 4.1: Bilden visar andel plattformsversioner för de mobiler som har kopplat upp sig mot Google Play under en tvåveckorsperiod, från 18 april till 1 maj 2012, i procent (Android Developers 2012q).

4.2. Fastställande av funktioner och kategorier

För att kunna besluta vilka funktioner och kategorier som skulle ingå i ChalmersAppen följdes i första hand kandidatgruppens önskemål med riktlinjer från enkätundersökningens resultat, se appendix A2. Resultatet visade vilka funktioner som var mest eftertraktade av studenterna och dessa lades till i kravspecifikationen. Denna kompletterades sedan med ytterligare funktioner som fanns med i det ursprungliga förslaget till kandidatprojektet samt de egna önskemålen.

4. Kravanalys och design

Med syfte att organisera liknande funktioner på ett överskådligt sätt delades dessa in i kategorier, som i sin tur även kunde innehålla underkategorier för att skapa en hierarkisk struktur. Grupperingen och sorteringen av funktionerna skedde efter hur de associerades till varandra. Liknande funktioner samlades under samma kategori eller underkategori och de som ansågs mest användbara fick en mer lättillgänglig placering på huvudmenyn.

4.3. Beskrivning av fastställda kategorier

För att ge en överblick på de fastställda kategorierna beskrivs respektive kategori överskådligt. För en mer övergripande struktur, se appendix B.

4.3.1. Kategori 1, 'Nyheter'

Verksamheter på Chalmers publicerar nyhetsflöden på sina webbsidor. Dessa verksamheter är bland annat högskolan, biblioteket och studentkåren. Målet med kategorin 'Nyheter' är att sammanställa dessa flöden på ett lättillgängligt sätt.

4.3.2. Kategori 2, 'Schema'

Chalmers använder systemet TimeEdit för att visa studenters scheman. Dessa scheman presenteras på en webbsida efter det att studenten sökt efter önskade kurser och program. Under kategorin 'Schema' i ChalmersAppen ska användaren kunna söka och se scheman.

4.3.3. Kategori 3, 'Mat'

Det finns flera restauranger och caféer på Chalmers båda campus, där alla har olika menyer och öppettider. För att underlätta val av matställe för användaren av ChalmersAppen ska dessa samlas under kategorin 'Mat'.

4.3.4. Kategori 4, 'Studentportal'

Studenter på Chalmers använder sig av webbsidan Studentportalen för att sköta sina studierelaterade uppgifter. Exempel på uppgifter är bland annat att registrera sig till tentamen, se rapporterade resultat och se aktuella kurser. För att komma åt dessa uppgifter krävs inloggning med hjälp av studentens Chalmers id, hädanefter CID, och lösenord. ChalmersAppens mål med kategorin 'Studentportal' är att plocka ut de mest använda funktionerna från Studentportalen och presentera dessa för användaren på ett lättillgängligt sätt.

4.3.5. Kategori 5, 'Salsbokning'

För att boka grupprum på Chalmers krävs tillgång till olika webbsidor där varje webbsida har sin egen form av inloggning. Vid bokning av grupprum i vad som kallas EDIT-huset krävs inloggning med hjälp av ett CID. För bokning av grupprum i kårhuset krävs användarens personnummer och vid bokning av grupprum i Chalmers bibliotek krävs det ett lånekort. Målet med funktionen 'Salsbokning' är att sammanställa dessa.

4. Kravanalys och design

4.3.6. Kategori 6, 'Kårkort'

Chalmers har ett kårobligatorium, vilket innebär att alla som studerar på Chalmers måste betala kåravgift (Chalmers Studentkår 2010). När denna avgift är betald får studenten ett personligt kårkort. Detta kårkort går att ladda med pengar för att få rabatt vid köp på Chalmers caféer, restauranger samt på bokhandeln Cremona. Chalmers Studentkår har en mobilanpassad webbsida för att se saldo och ladda kårkort med pengar. Kategorin 'Kårkort' ska presentera denna webbsida i ChalmersAppen.

4.3.7. Kategori 7, 'Karta'

Det finns idag kartor över Chalmers båda campus på olika webbsidor, men ingen av dessa är anpassade till mobiltelefoner. Kategorin 'Karta' ska presentera salar, gruppum och andra platser på campusen på en interaktiv karta. Användaren ska se sin aktuella position med hjälp av antingen GPS eller mobila nätverket, och kunna söka platser efter deras namn.

4.3.8. Kategori 8, 'Viktiga datum'

En sammanställning över viktiga datum under ett läsår på Chalmers, såsom läsperioder, tentamensperioder, lov och dylikt, erhålls av Studentportalen. Syftet med kategorin 'Viktiga datum' är att göra denna information tillgänglig för användaren redan på huvudmenyn.

4.3.9. Kategori 9, 'Resa'

Västrafik tillhandahåller information om buss- och spårvagnsavgångar från samtliga hållplatser i Västra Götalandsregionen. Målet med kategorin 'Resa' är att tillgängliggöra denna information för de hållplatser som ligger närmast Chalmers båda campus.

4.4. Kravspecifikation

En kravspecifikation upprättades för utveckling av ChalmersAppen. Nedan ges en beskrivning av huvudrubrikerna. För fullständigt dokument se appendix C.

4.4.1. Målsättning

Under "Målsättning" sammanfattas syftet med applikationen och hur kravspecifikationen ska användas som hjälpmedel för implementationen.

4.4.2. Funktionella krav

De funktionella kraven har ordnats i tabeller som består av kolumner med numrering av kraven, prioritet, lägesrapport och vilka gruppmedlemmar som ansvarar för varje specifikt krav. Tabellerna har uppdaterats under projektets gång med nya statusuppdateringar, omprioriteringar och ansvarsområden. Varje kategori har tilldelats en egen tabell för att på ett lättare sätt dela upp arbetsuppgifter mellan gruppmedlemmarna. I de fall då tabeller blivit för omfattande har dessa delats upp i flera mindre tabeller för att på så sätt underlätta implementationen. De funktioner som fått högst prioritet ansågs vara mest nödvändiga att ha med i applikationen och skulle därför implementeras först. De med lägst prioritet ansågs vara mindre nödvändiga och skulle därför implementeras i mån av tid.

4. Kravanalys och design

4.4.3. Icke-funktionella krav

En lista med icke-funktionella krav upprättades. Applikationen ska till exempel vara användarvänlig och varje funktion ska kunna nås med ett minimalt antal tryck. Applikationen ska även vara relativt säker och inte riskera att användaren förlorar eller delar ut känslig data som till exempel sitt CID med tillhörande lösenord.

4.4.4. Programmeringsregler

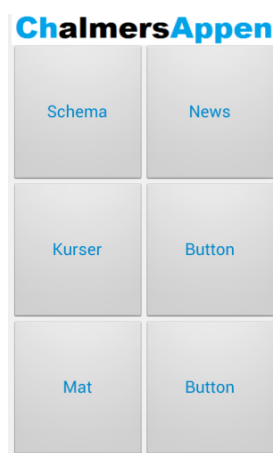
För att underlätta implementationsfasen och få en enhetlig källkodsstruktur, sattes ett antal programmeringsregler upp i kravspecifikationen. Reglerna innefattar hur källkoden skulle se ut och vilka konventioner som skulle följas när det gällde namngivning av klasser, metoder och variabler. Anledningen till att dessa regler skapades var att utvecklare skulle förstå källkoden bättre.

4.5. Grafisk design

Den grafiska designen på en applikation kan vara avgörande för användarupplevelsen. Detta är anledningen till att ett enkelt och stilrent gränssnitt har tagits fram för ChalmersAppen. För att få en så bra design som möjligt har framtagandet genomförts iterativt, från tydliga skisser till funktionella prototyper. Det slutgiltiga gränssnittet beskrivs under respektive kategori i kapitel 6.

Det grafiska gränssnittet utvecklades med syfte att underlätta för användaren att uppnå sina mål. Då ChalmersAppen är utvecklad för smarttelefoner ska det grafiska gränssnittet vara förhållandevis enkelt, med ett fåtal element, och göra alla funktioner nåbara inom ett fåtal klick. Målet blev därför att skapa en överskådlig, lättnavigerad huvudmeny.

I den första designen bestod ChalmersAppens huvudmeny av sex knappar innehållande olika kategorier, se figur 4.2.



Figur 4.2: Bilden visar den ursprungliga designprototypen av ChalmersAppens huvudmeny med sex knappar.

4. Kravanalys och design

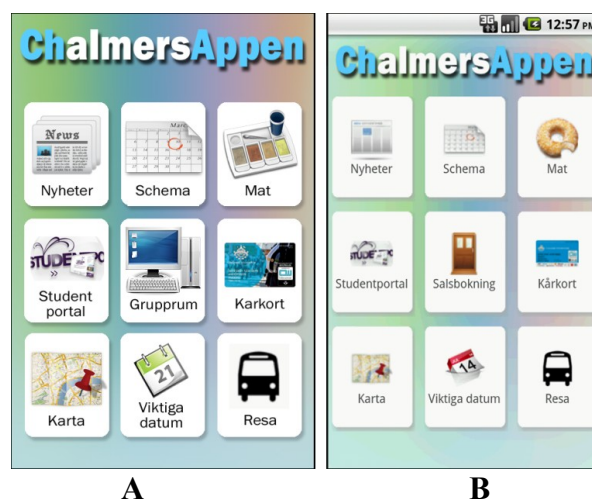
Efter övervägande ändrades dessa sex knappar till nio knappar, eller kategorier. Nio kategorier ansågs vara den bästa lösningen eftersom knapparna då kunde rymma relevant funktionalitet samt att knapparna kunde läggas i ett enhetligt rutnät med jämnt antal element på både höjd och bredd.

Därefter togs en klickbar designprototyp fram, se figur 4.3. Detta gjorde det möjligt att testa och utvärdera denna lösnings utseende och funktionalitet utan hjälp av pappersprototyper.



Figur 4.3: Bilden visar den första klickbara designprototypen i Android med ett enhetligt rutnät med tre rader och tre kolumner.

Av ett antal olika designförslag valdes en version av en slutgiltig designprototyp ut, se figur 4.4. Den skapades med hjälp av bakgrundsbilden nedladdad från Chalmers Studentportal och ett antal gratisikoner nedladdade från webbsidan IconFinder. Prototypens utseende är designat för att efterlikna Chalmers Studentportals färgtema och då även skapa en känsla av igenkänning för användaren. Knapparna var förhållandevis stora för att underlätta för användaren att interagera med ChalmersAppen, både med tanke på olika skärmstorlekar och på att minimera antalet feltryckningar.



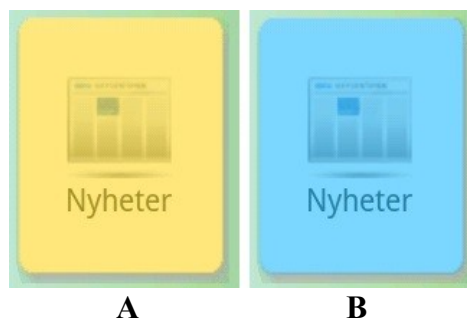
Figur 4.4: Bild A visar den slutgiltiga designprototypen som blev grunden till applikationens huvudmeny som syns på bild B.

5. Implementation av kategorier och funktioner

I följande kapitel beskrivs hur ChalmersAppens funktioner implementerades. Dessa är uppdelade i kategorier och presenteras i samma ordning som de är placerade i ChalmersAppens huvudmeny.

5.1. Huvudmenyn

Huvudmenyn består av en aktivitet som utnyttjar ett `GridView`-element, härnå efter rutnätsvy. En rutnätsvy är en tvådimensionell rullbar lista (Android Developers 2012r). I varje ruta i vyn har en bild placerats för att representera knappar. När användaren trycker på en knapp eller väljer det med hjälp av styrkulan, markeras elementet med ett färgöverlägg. Färgöverlägget implementerades genom att rutnätsvyn fick ett attribut som påverkar dess bakgrund. Attributet kopplade rutnätsvyn med två PNG-filer med olika färger och 50 procents genomskinlighet; en för ”fokus” och en för ”tryck”. Detta gör att knappen som fokus ligger på får ett gult sken, och vid tryck blir skenet turkost, se figur 5.1. Utan det ovannämnda attributet skulle användaren inte få någon grafisk återkoppling vid tryck på ett element i rutnätsvyn.



Figur 5.1: En knapp i huvudmenyn som fokus ligger på får gult sken, som visas i bild A. En knapp som trycks på får turkos sken, som syns i bild B.

5.2. Nyheter

Kategorin ‘Nyheter’ presenterar listor med nyhetsflöden som ursprungligen kommer från en XML-fil, skriven i RSS-format, se kapitel 3.3. RSS-flödena hämtas från Chalmers, Studentkårens och bibliotekets respektive webbsidor. Kategorin implementerades genom parsning i två steg. I det första steget omvandlas filens innehåll från RSS-formatet till ett sträng-format med hjälp av en XML-parser. Artikeltexten i RSS-formatet består i vissa fall endast av sammanfattningar och därför hämtas, i ett andra steg, de fullständiga nyhetsartiklarnas källkod. Dessa parsas med hjälp av en HTML-parser och resultatet visas därefter i ett `WebView`-element, fortsättningsvis webbvy, som används för att visa webbsidor och tolka HTML-kod (Android Developers 2012s).

5. Implementation av kategorier och funktioner

5.2.1. Parsning av nyhetsflödet

Det finns ett flertal tekniker när det gäller XML-parsning, två av dessa har utnyttjats för parsing av nyheterna. De två är DOM, *Document Object Model*, och SAX, *Simple API for XML* (Exforsys Inc. 2007).

DOM parsar XML-dokument genom att bygga upp en trädstruktur av dokumentet i minnet (Franklin 2010). Denna metod kräver mycket minnesutrymme och gör parsningstekniken mindre effektiv än till exempel SAX-parsning (Exforsys Inc. 2007). SAX läser in dokumentet sekventiellt och baserar parsningen på händelser som måste hanteras manuellt, vilket istället gör den mer svårimplementerad än DOM (Franklin 2010). I nyhetsfunktionen har en kombination av båda dessa parsningstekniker använts för att få bästa möjliga resultat.

För att parse RSS-flöden användes Javas inbyggda SAX-parser. Vid implementationen av parsern skrevs en hjälpklass där reglerna för parsningen bestämdes. Vidare skrevs två klasser för lagring av RSS-flöden, varav den ena lagrar en enskild artikel i flödet och den andra innehåller en lista med flera artiklar.

Nyhetsflöden läses in i ett `ListView`-element, fortsättningsvis listvy, som visar objekt i en vertikalt rullbar lista (Android Developers 2012t).

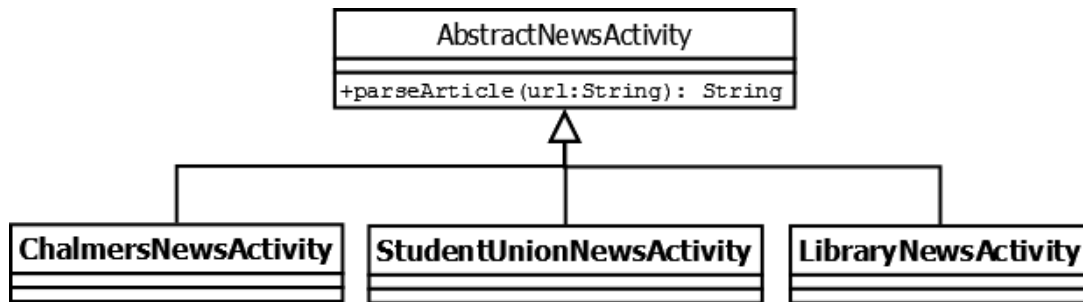
5.2.2. Parsning av artiklar

Vid tryck på ett listelement parsas artikeln med biblioteket Jsoup, se kapitel 3.4, som i sin tur använder sig av DOM, och skickas till en aktivitet som innehåller en webbvvy där den parsade artikeln visas (Hedley 2012). Aktiviteten innehåller även en knapp vars funktion är att öppna artikelns webbadress i telefonens webbläsare.

Till skillnad från RSS-flöden, som följer en standard och därmed kan parsas på samma sätt, publiceras de ursprungliga artiklarna med olika HTML-formateringar på webbsidorna. Dessa skillnader gjorde att Javakällkoden som hanterar parsningen för artiklarna skrevs på olika sätt för webbsidorna.

För att inte källkoden ska dupliceras skapades en abstrakt klass som samlar all gemensam källkod för parsningen av RSS-flödena. Klassen ärvs av en subklass för varje webbsida. Subklasserna implementerar en abstrakt metod som hanterar parsningen av HTML-kod från respektive webbsida, se figur 5.2.

5. Implementation av kategorier och funktioner



Figur 5.2: Det nerskalade UML-diagrammet visar arkitekturen för kategorin 'Nyheter'. De tre subklasserna ChalmersNewsActivity, StudentUnionNewsActivity och LibraryNewsActivity implementerar den abstrakta metoden `parseArticle()`.

5.3. Schema

Kategorin 'Schema' implementerades med en webbvyn som huvudkomponent. Vid tryck på knappen för kategorin utförs ett test som undersöker om användaren har sparat något schema. Om så är fallet laddas webbadressen för detta schema automatiskt från applikationens gemensamma inställningar till webbvyn. Ifall testet inte hittar något sparat schema så laddar webbvyn istället webbsidan för schemasökning.

För att låta användaren spara ett valt schema har tre knappar 'Sök schema', 'Spara schema' och 'Visa sparad schema' placerats ut. När användaren trycker på 'Sök schema' laddas webbsidan för schemasökning i webbvyn. Vid tryck på knappen 'Spara schema' sparas den aktuella webbadressen i applikationens gemensamma inställningar. Knappen 'Visa sparad schema' hämtar den sparade adressen från inställningarna och laddar upp denna i webbvyn.

5.4. Mat

Kategorin 'Mat' består huvudsakligen av två aktiviteter. I den första aktiviteten placerades en vy med flikar för de olika restaurangerna. Den andra aktiviteten innehåller en listvy med veckans lunchmenyer samt två knappar, 'Visa på karta' och 'Mer Information'.

Knappen 'Mer Information' öppnar en webbvyn som visar öppettider och information om restaurangen. Bytet mellan listorna och webbvyn har implementerats med hjälp av ett element som används för att växla mellan två fördefinierade vyer, se figur 5.3.

5. Implementation av kategorier och funktioner

```
<ViewSwitcher
    android:id="@+id/viewFlipperMoreInfo"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_weight="0.15" >

    <ListView
        android:id="@+id/ListViewMenu"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@android:color/transparent"
        android:cacheColorHint="#00000000" >
    </ListView>

    <android.webkit.WebView
        android:id="@+id/webViewLunch"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" >
    </android.webkit.WebView>

</ViewSwitcher>
```

Figur 5.3: Bilden visar källkoden för XML-komponenten ViewSwitcher och dess två underelement: ListView och WebView. ViewSwitcher gör det möjligt att växla mellan underelementen.

Informationen som visas i webbvyn hämtas från webbsidan chalmerskonferens.se. Endast relevant information parsas med Jsoup, se figur 5.4. Parsningen har implementerats på ett likartat sätt eftersom informationen för alla restaurangerna finns på samma webbsida med i princip samma HTML-formatering.



	Ordinarie pris	Kårkort	Gästkort
Classic/Take Away	70 / 63 kr	60 / 54 kr	65 / 59 kr
Veckans soppa/Take Away	55 / 50 kr	45 / 41 kr	50 / 45 kr
Xpress/Take Away	60 / 54 kr	50 / 45 kr	55 / 50 kr
Gröna Väggen/Take Away	60 / 54 kr	50 / 45 kr	55 / 50 kr

Dagens lunchmeny

I vår stora lunchrestaurang Kårrestaurangen erbjuder vi ett brett och välsorterat utbud såsom en kött rätt, fiskrätt, veckans soppa, vårt enklare matalternativ xpress eller Gröna väggen. Vi har något för alla!

Till Classic, xpress och soppan ingår salladsbuffé, smör, bröd, kaffe/te och ett glas valfri dryck.

Till Gröna Väggen ingår en matlåda eller tallrik, salladsbuffén vid väggen, smör, bröd, kaffe/te och ett glas valfri dryck

Mån Fre 11:00 13:30

Chalmers Kårhus

031 772 39 50

bokning@chalmerskonferens.se

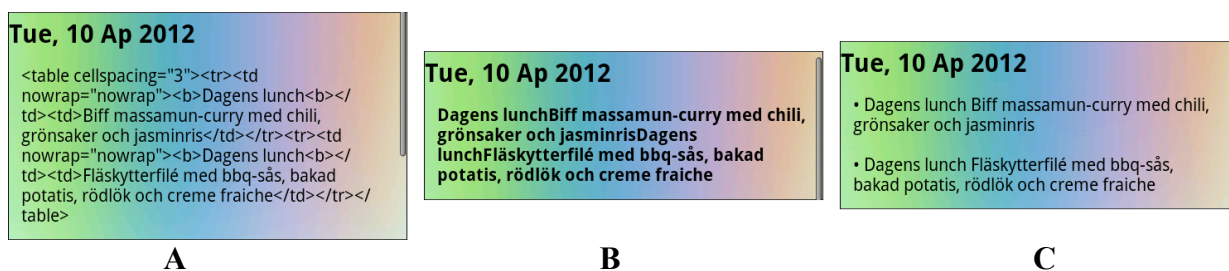
Figur 5.4: Bilden visar parsad information om Kårrestaurangen från webbsidan chalmerskonferens.se.

Webbsidan erbjuder lunchmenyerna från alla restauranger på båda campus i form av RSS-flöden, vilket underlättade implementationen av parsningen. Vid parsningen av lunchmenyerna används samma klasser som för parsningen av nyhetsflödena som beskrevs i kapitel 5.2.1.

5. Implementation av kategorier och funktioner

Implementationen av parsningen för lunchmenyerna genomgick ett antal förbättringar för att göra lunchmenyerna så lättläsliga som möjligt för användaren. Figur 5.5 visar olika resultat som uppnåddes under varje iteration. Första försöket att parsar texten från menyflödena resulterade i oläslig text på grund av att den innehöll HTML-kod, se figur 5.5 A. Därför användes Androids egna `Html`-klass för att ta bort taggarna, se figur 5.5 B.

Den slutgiltiga versionen, som visas i figur 5.5 C, togs fram genom en kombination av SAX-parsning för RSS-flödet samt Jsoup-parsning för att strukturera upp texten och filtrera bort HTML-taggarna. All parsning av lunchmenyerna sker i en separat tråd för att inte blockera användargränssnittet och operativsystemet.



Figur 5.5: Bild A visar första iterationen med HTML-taggarna. Bild B visar andra iterationen. Bild C visar den sista iterationen med strukturerad och lättläslig text.

Kategorin hämtar information om campus från applikationens inställningar och presenterar lunchmenyerna från detta campus för användaren. För att öka användarvänligheten rullar listvyn, som innehåller lunchmenyerna, automatiskt fram till dagens meny. Denna funktionalitet implementerades med hjälp av Javas `Calendar`-klass för att beräkna den aktuella veckodagen.

5.5. Studentportal

Kategorin 'Studentportal' implementerades genom att ladda in webbadressen till Chalmers studentportals inloggningssida i en webbvyn.

5.6. Salsbokning

Kategorin 'Salsbokning' implementerades med hjälp av en vy innehållande tre flikar, på samma sätt som i kategorierna Nyheter och Mat, med bokningssidorna TimeEdit, Aptus (kårhuset) och MRBS (biblioteket). Beroende på vilken flik som är markerad kommer den valda webbsidan att visas i en webbvyn.

5.7. Kårkort

Kategorin ‘Kårkort’ implementerades genom att ladda in webbadressen till Chalmers studentkårs webbsida för kårkort i en webbvvy. Då webbsidan är mobilanpassad länkas användaren enbart till denna.

5.8. Karta

Kategorin ‘Karta’ var den mest omfattande och implementerades genom att skapa en aktivitet som ärver från klassen `MapActivity` i Google Maps API. Detta medför att det mesta av det grafiska gränssnittet, framför allt kartvyn, får ett förutbestämt utseende som är karaktäristiskt för Google Maps. Även en del av interaktionen är fördefinierad, till exempel möjligheten att flytta kartan genom att dra den med fingret. Möjligheten att förstora och förminska bilden finns genom att trycka på skärmen med två fingrar samtidigt och dra isär respektive föra ihop dem. Det är dock möjligt att anpassa kartfunktionen genom att placera valfria element ovanpå kartan samt definiera på vilka sätt en användare kan interagera med kartan.

Vid implementationen av lagerfunktionen på kartan användes den abstrakta klassen `Overlay` från Google Maps API. Subklasser till denna klass används för att visa olika lager på kartan. Lager innehåller geografiska koordinater som kan visas på kartan som punkter. Fyra lager implementerades för kartan, ett för att visa användarens aktuella position, ett för hörsalar, ett för matställen och ett för grupprum.

Platserna som läggs upp på kartan tilldelades ikoner i olika färger, till exempel sattes matställen till röda prickar, grupprum till gula och föreläsningssalar till gröna. Alla element i ett lager har samma ikon, vilket är anledningen till att platser av samma typ har placerats i samma lager.

Varje objekt som kan finnas i de olika lagren innehåller, förutom dess fysiska koordinater, ett namn och en kort beskrivning av platsen. Informationen om varje plats hämtas från databasen, som beskrivs närmare i kapitel 5.8.2.

Ett av målen var att ha klickbara informationsbubblor på kartan, men eftersom Google Maps API inte stödjer denna lösning som standard, användes ett externt bibliotek `Android MapView Balloons` som beskrevs i kapitel 3.4. Genom att använda detta tillvägagångssätt visas en informationsbubbla när användaren trycker på en platsikon, se figur 5.6.



Figur 5.6. Bilden visar en informationsbubbla som visas när användaren trycker på en platsikon på kartan.

5. Implementation av kategorier och funktioner

Kartvyn innehåller som standard två knappar för att förstora och förminska vyn. Knapparna är inte anpassningsbara, vilket är orsaken till att ChalmersAppen istället har två vanliga ikonknappar som tar över funktionaliteten att förstora och förminska.

Användaren kan aktivera och avaktivera olika lager på kartan och visa enbart den platstyp som är av intresse. Möjligheten att aktivera och avaktivera lager implementerades genom att samtliga lager tas bort och de valda ritas om på nytt. Vilka lager som ska vara aktiva respektive inaktiva väljs av användaren i en dialogruta och sparas i applikationens inställningar.

Knappen ‘Min position’, låter användaren centrera vyn på telefonens geografiska position, och placerades i den utfällbara menyn för att inte kartvyn skulle täckas av överflödiga element. Vid tryck på denna knapp görs ett försök för att hämta telefonens position med hjälp av en platstjänst. Positionen hämtas antingen från GPS-satelliter eller genom trådlösa nätverk, beroende på vilken av dessa källor som för tillfället är tillgänglig och vilken som tillhandahåller den mest exakta informationen. Denna information används sedan för att uppdatera lagret med användarens position.

Ifall alla platstjänster är inaktiverade, visas en dialogruta som ger användaren möjlighet att komma åt och ändra inställningarna. Om detta alternativ väljs, startas en aktivitet som låter användaren aktivera och inaktivera platstjänster.

5.8.1. Sökfunktionen

Sökfunktionen för kartan implementerades för att möjliggöra sökning av platser på campus. Detta gjordes genom att ange kartans aktivitet som sökbar i manifestet. Vid tryck på antingen den fysiska eller den virtuella sökknappen visas en sökmeny och ett virtuellt tangentbord på skärmen.

När användaren sedan antingen börjar skriva in en text eller trycker på sökknappen i sökmenyn, skickar systemet ett `Intent`-objekt innehållande en sträng med sökfrasen tillbaka till kartans aktivitet. I vanliga fall skulle detta starta en ny instans av kartans aktivitet. Detta är anledningen till att kartans aktivitet tilldelades uppstartsläget “*singleTop*”, vilket medför att enbart en instans körs åt gången (Android Developers 2012u).

Varje gång användaren skriver eller tar bort ett tecken i söksträngen visas en resultatlista med salar och platser vars namn eller beskrivning stämmer överens med söksträngen. Resultatlistan fylls av element från databasen med hjälp av en subklass till `ContentProvider`, som beskrevs i kapitel 2.3.4, och dess uppgift är att skapa anpassade resultatförslag som stämmer överens med söksträngen (Android Developers 2012v). När användaren sedan klickar på ett resultatförslag, skickas detta objekts id-nummer till kartans aktivitet. Där används id-numret för att anropa databasen och hämta data som behövs för att visa upp objektet på kartan. Därefter centreras kartvyn på detta objekt och ett tryck simuleras på dess position så att en informationsbubbla visas ovanför.

5. Implementation av kategorier och funktioner

Om användaren däremot trycker på Enter-knappen, eller sökknappen i sökmenyn, efter att ha skrivit in minst ett tecken i sökfältet söks databasen igenom efter platser som stämmer in på sökkriterierna. Resultatet returneras då i form av en lista.

Om resultatlistan är tom, visas meddelandet “Din sökning gav inga resultat” för användaren. I annat fall skapas ytterligare lager på kartan. Först skapas ett lager för varje plats i resultatlistan, så att samtliga funna platser är synliga på kartan, även ifall något av lagren är avstängt. Sedan skapas ett lager där alla platser som finns med i resultatlistan markeras med hjälp av en markör. Samtidigt beräknas avstånden mellan mitten av kartvyn och punkterna i resultatlistan, och på den punkt som ligger närmast mitten simuleras även i detta fall ett tryck. Dessutom visas en knapp vars funktion är att rensa sökresultat. Vid tryck på denna knapp simuleras ett tryck på den fysiska tillbakaknappen. Då ritas alla aktiva lager om och knappen som rensar sökresultat döljs igen.

5.8.2. Databasen

Databasen utgörs av SQLite-fil. Det är en färdig databasfil med manuellt inmatad data som innehåller information om fysiska platser på Chalmers båda campusområden såsom restauranger och föreläsningssalar. När kartan startas kopieras databasfilen till applikationens interna databasmapp.

Vid anrop av hämtningsfunktioner i databasens gränssnitt, öppnas databasen i skrivskyddat läge och de önskade objekten hämtas från den i form av rådata. Rådatan omvandlas till objekt som sedan returneras till den anropande metoden.

5.9. Viktiga datum

Parsningen för kategorin ‘Viktiga datum’ sker i två steg. Det första steget är att hämta och parsar alla befintliga läsår från webbsidan Läsårstider i Studieportalen, Chalmers äldre version av Studentportalen. Med läsår menas endast de alternativ besökaren till webbsidan kan välja mellan. Det andra steget är att hämta och parsar informationen som tillhör respektive läsår. För att göra kategorin resurssnål hämtas informationen endast för det läsår som användaren har valt. Detta innebär att läsåren med respektive information inte kan hämtas samtidigt och att läsåren måste hämtas först. Det är därför som parsningen sker stegvis.

Implementationen av ‘Viktiga Datum’ använder sig av det externa biblioteket Jsoup för att både ansluta, hämta och parsar läsåren. Dessa är skrivna inom `option`-taggar i HTML-koden för webbsidan, vilket underlättar parsningen. Kategorin ‘Viktiga datum’ har som standard att presentera informationen för det nuvarande läsåret. Denna parsas separat och på liknande sätt som de övriga, men med skillnaden att endast element med attributet `selected` behålls. Av de läsår som finns på webbsidan Läsårstider parsas endast de åtta senaste. Detta för att menyn som låter användaren välja vilket läsår som ska visas inte ska bli för lång.

Vidare används Jsoup för att både ansluta, hämta och parsar informationen för de åtta läsåren. Denna information består av två delar, en löpande text och en tabell. Parsningen av den löpande texten utförs på två olika sätt beroende på vilket läsår det är. Läsåret 2012/2013

5. Implementation av kategorier och funktioner

skiljer sig från de övriga genom att texten är placerad inom en `body`-tagg och inte inom en `p`-tagg. Taggar är standard inom HTML-kod och representerar olika delar av källkod på en webbsida (W3Schools 2012c). Genom ett anrop med respektive tagg som argument kan texten till samtliga läsår parsas. Valet av parsningssätt baseras på om `p`-taggen finns i HTML-koden för webbsidan. Parsningen av tabellen utförs på samma sätt men med `table`-taggen som argument. Slutligen sammanfogas båda delarna till ett komplett HTML-dokument som sedan skrivs ut med all formatering i en webbvvy.

5.9.1. Notifikationer

Notifikationsfunktionen implementerades genom att skapa en klass som ärver från huvudkomponenten `Service` eller tjänst, se kapitel 2.3.2. Tjänsten implementerades så att den startar när enhetens uppstartsprocess är avslutad samt när enhetens knapplås avaktiveras. För att lyssna efter dessa “handlingar” skapades en klass som fick ärva från `BroadcastReceiver`, se kapitel 2.3.3. Klassen kontrollerar också att notifikationerna är påslagna genom att hämta relevanta värden från applikationens inställningar, se kapitel 5.11.2. Tjänsten startar i sin tur en ny tråd som körs i bakgrunden så länge användaren behåller notifikationerna påslagna i inställningarna.

Notifikationerna är inställda att startas en gång per dygn, genom att låta tråden vila i 24 timmar för att därefter startas upp på nytt. Då notifikationerna är aktiva parsas tabellen för läsårstider på Chalmers studentportals webbsida med hjälp av Jsoup. Vid parsningen jämförs det aktuella datumet med alla startdatum i tabellen och om dessa datum överensstämmer kontrolleras därefter om användaren valt att bli påmind om den specifika händelsen. Om så är fallet så skapas ett `Intent`, se kapitel 2.3.5, som skickar en påminnelse till telefonens statusrad.

5.10. Resa

Kategorin ‘Resa’ använder sig av Västtrafiks API, vilket tillhandahåller trafikdata i JSON- och XML-format i enlighet med REST, *Representational State Transfer* (Rodriguez 2008). REST är en uppsättning principer som används för att utforma webbtjänster. Enligt REST ska en klient som vill interagera med en server göra detta genom explicita HTTP-anrop. Vid ett anrop anges en URI, *Uniform Resource Identifier*, som specificerar vilken information som önskas.

ChalmersAppen hämtar trafikdatan i JSON-form, se kapitel 3.3. För att hämta ut avgångar för en viss hållplats krävs det två anrop. Det första använder en hållplats namn för att hämta ut dess id-nummer. Dessa id-nummer är inte beständiga och kan ändras utan förvarning. Utvecklare som använder sig av API:t rekommenderas alltså att inte lagra dessa lokalt under en längre tid (HaCon 2011), i stället lagras hållplatsernas namn. De sparas som strängar i en XML-fil och funktionen kan därför utökas med fler hållplatser utan att någon källkod behöver ändras. Det andra anropet använder det erhållna id-numret för att hämta avgångarna från den aktuella hållplatsen.

5. Implementation av kategorier och funktioner

För att hantera JSON-data skapades en klass som läser in datan och parsar den med hjälp av det öppna biblioteket Gson, se kapitel 3.4. JSON-datan översätts till en lista med vanliga Javaobjekt som sorteras efter den verkliga avgångstiden. Aktiviteten som hanterar den grafiska delen av hållplatsinformationen hålls helt ovetandes om den bakomliggande implementationen.

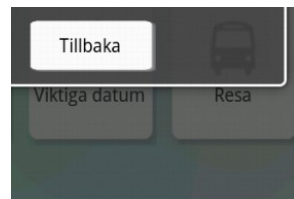
5.11. Menyer

Samtliga Android-telefoner har en menyknapp som normalt ligger utanför pekskärmsytan, och är då en fysisk knapp (Google Inc. 2010). Ramverket för Android-utveckling ger utvecklarna tillgång till denna knapp och dess funktion att öppna en meny.

Menyn byggs normalt upp i XML genom att skapa en komponent med underelement och får automatiskt ett standardutseende om utvecklaren inte väljer att skraddarsy den på egen hand. Menyer har placerats i huvudaktiviteter för varje kategori för att ge användaren snabb åtkomst till viktiga inställningar utan att behöva backa till huvudmenyn.

5.11.1. Om

Aktiviteten som startas vid tryck på menyalternativet ‘Om’ har i applikationens manifest-fil deklarerats med ett attribut som ger dess användargränssnitt ett genomskinligt utseende, se figur 5.7.



Figur 5.7: Bilden visar det transparenta utseendet för menyalternativet ‘Om’.

5.11.2. Inställningar

Androids API erbjuder en variant på huvudkomponenten `Activity` för att underlätta för utvecklare att snabbt kunna skapa standardinställningar i form av listor. Denna funktionalitet har utnyttjats för att underlätta utvecklingen av ChalmersAppen. Inställningar i Android skapas lättast i XML där det finns ett antal olika typer av komponenter att välja mellan. Information som skrivs in i inställningarna sparas automatiskt i applikationens gemensamma inställningar och sparas även då applikationen inte körs.

För att spara CID och lösenord används ett XML-element som låter användaren skriva in och spara text i en textruta (Android Developers 2012w). Lösenordet döljs med hjälp av ett attribut som gör så att prickar visas istället för text.

Möjligheten att sätta på och stänga av notifikationer i inställningarna använder en XML-komponent som hanterar booleska variabler och kan antingen få värdet “true” eller “false”.

6. Resultat

I nästföljande kapitel presenteras resultatet av kandidatarbetet. Först beskrivs resultatet av den utvecklade applikationen, ChalmersAppen, därefter förklaras uppfyllda samt icke-uppfyllda mål.

6.1. Nyheter

Kategorin 'Nyheter' presenterar nyhetsflöden från tre olika källor på Chalmers: Chalmers huvudwebb sida (chalmers.se), kårens webb sida (chs.chalmers.se) och bibliotekets webb sida (lib.chalmers.se).

För att öka användarvänligheten placerades nyhetsflödena i rullningslistor under var sin flik. Ett listelement visar artikelns titel, publiceringsdatum samt en beskrivning av artikeln förkortad till 140 tecken, se figur 6.1 A.

Vid tryck på ett listelement visas artikeln upp i en ny vy. Det som visas är enbart artikeltexten med eventuella medföljande bilder och länkar; allt oväsentligt såsom layout och menyerna på artikelsidan togs bort, se figur 6.1 B. Ovanför artikeltexten placerades en knapp, vars funktion är att öppna telefonens standardwebbläsare och bläddra till artikelns ursprungliga webbadress, se figur 6.1 C.



Figur 6.1: Bild A visar nyhetsflödet från chalmers.se i applikationen där en av nyhetsartiklarna har valts. Bild B visar den parsade nyhetsartikeln från bild A. När knappen nedanför artikeln tryckts, visas artikeln i telefonens webbläsare, som på bild C.

6.2. Schema

Kategorin 'Schema' visar en webbvyn med webbsidan för TimeEdit:s schemasökningsfunktion. Nedanför vyn visas tre knappar: 'Sök schema', 'Spara schema' och 'Visa sparad schema'. Knappen 'Sök schema' laddar upp webbsidan för schemasökning där användaren kan söka på valfria kurser och program på Chalmers, se figur 6.2. 'Spara schema' låter användaren spara det schema som visas för tillfället vid trycket. Den sistnämnda knappen 'Visa sparad schema' laddar in det sparade schemat och visar detta för användaren. Det schema som användaren väljer att spara kommer att laddas upp automatiskt nästa gång användaren startar schema-funktionen.



Figur 6.2: Bilden visar kurssökningen i TimeEdit med kategorin 'Schema'.

6.3. Mat

Kategorin 'Mat' visar lunchmenyer från restauranger som ligger på Chalmers båda campus. Lunchmenyerna hämtas från webbsidan chalmerskonferens.se och presenteras i listor med veckans lunchmenyer från de olika restaurangerna på campus Johanneberg och Lindholmen, se figur 6.3 A.

De restauranger som visas under kategorin 'Mat' är Kårrestaurangen, Linsen, R.F Byssja och Hyllan på campus Johanneberg samt Kokboken, L's Kitchen och Restaurang R på campus Lindholmen. För varje restaurang finns det en knapp "Mer information" som låter användaren läsa en kort beskrivning samt se öppettider för restaurangen. Figur 6.3 B visar exempel på hur informationen för Kårrestaurangen presenteras.

6. Resultat



Figur 6.3: Bild A visar lunchmenyerna för Kårrestaurangen. Bild B visar funktionen ‘Mer information’.

Ett tryck på knappen ‘Visa på Kartan’ byter vy till applikationens kartfunktion och markerar den aktuella restaurangens position med en informationsbubbla.

Restaurangerna visas i var sin flik och menyerna för respektive restaurang visas i rullningslistor. För att öka användarvänligheten, centreras vyn automatiskt på dagens lunchmeny som även belyses med vit bakgrundsfärg, se figur 6.3 A.

För att presentera så relevant information som möjligt tillhandahåller ChalmersAppen inställningar som låter användaren ange vilket campus han eller hon studerar på. Utifrån detta visas de mest relevanta lunchmenyerna först. Om användaren väljer att visa menyer för det andra campuset, finns det en möjlighet att byta campus tillfälligt genom att trycka på telefonens menyknapp och välja detta alternativ. Figur 6.4 visar menyalternativet ‘Lindholmen’ som växlar till Lindholmens menyer.

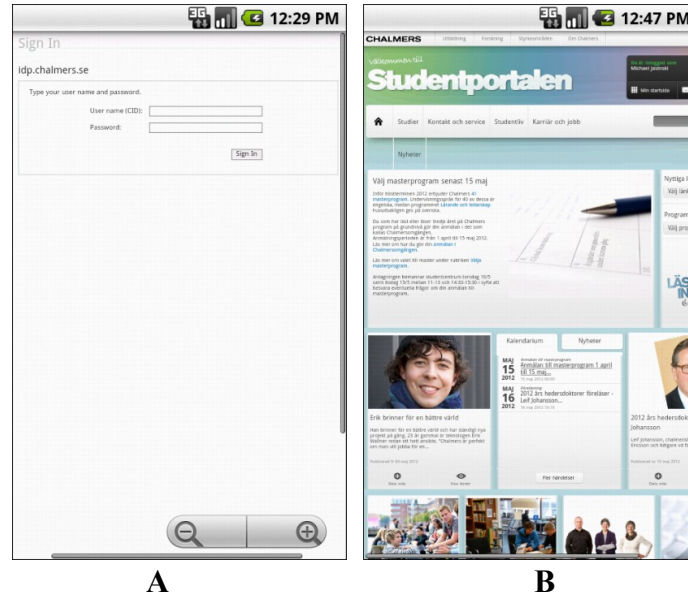


Figur 6.4: Bilden visar den utfällbara menyn under kategorin ‘Mat’.

6. Resultat

6.4. Studentportal

Kategorin ‘Studentportal’ visar Chalmers Studentportals inloggningssida i en webbvvy, se figur 6.5 A, varifrån användaren kan komma åt samtliga funktioner och tjänster i Studentportalen, se figur 6.5 B.



Figur 6.5: Bild A visar hur inloggningen till Studentportalen ser ut. På bild B visas en webbvvy av Studentportalen.

6.5. Salsbokning

Kategorin ‘Salsbokning’ visar flikar med vilka användaren kan byta mellan de system som finns för att boka salar och grupprum på Chalmers båda campus. De alternativ som ges är ‘Chalmers’, ‘Kårhuset’ och ‘Biblioteket’, se figur 6.6. När användaren väljer ett av dessa, laddas respektive webbsida upp och användaren kan logga in för att boka sal eller grupprum.



Figur 6.6: Bilden visar kategorin ‘Salsbokning’ med ‘Kårhuset’ som valt alternativ.

6. Resultat

6.6. Kårkort

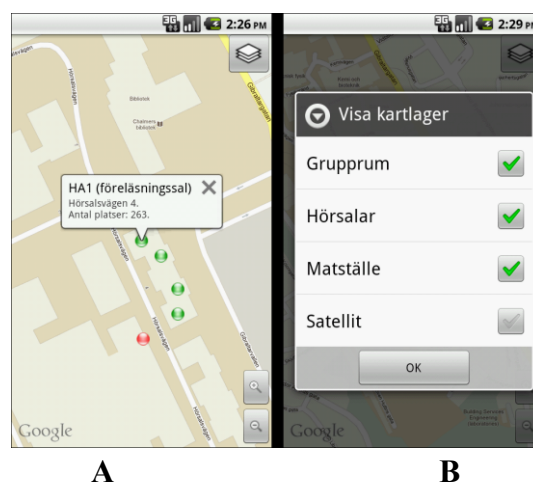
I kategorin 'Kårkort' visas en mobilanpassad webbsida som tillhandahålls av Chalmers Studentkår, där användaren väljer vilken typ av kort som han eller hon innehar. Därefter öppnas respektive webbsida där användaren kan se saldo och ladda kårkortet med pengar, se figur 6.7.



Figur 6.7: Bilden visar Chalmers Studentkårs webbsida där användaren kan välja vilken typ av kårkort som ska laddas.

6.7. Karta

Kategorin 'Karta' visar en karta där platser placeras ut på Chalmers båda campusområden. Syftet med detta är främst att underlätta för nya studenter att orientera sig på sitt campus, men även äldre studenter kan ha nytta av funktionen. Platser som placeras ut på kartan i form av platsikoner är föreläsningssalar, grupprum och matställen. När användaren trycker på en platsikon på kartan, visas en informationsbubbla med grundläggande information om platsen, se figur 6.8 A.



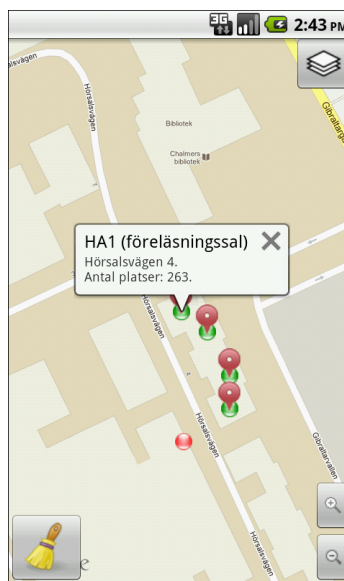
Figur 6.8. Bild A visar en informationsbubbla vid tryck på en platsikon i kartvyn. Bild B visar dialogrutan som låter användaren välja vilka lager ska synas på kartan.

6. Resultat

Användargränssnittet består av tre knappar: två knappar i det nedre högra hörnet som hanterar förstörings- och förminskningsfunktionerna, samt en knapp i det övre högra hörnet där användaren kan välja kartlager, se figur 6.8 B. Med denna funktion kan användaren aktivera eller inaktivera visning av lager med olika platstyper, exempelvis matställen. Tack vare detta kan användaren lättare få överblick över de platserna som han eller hon är intresserad av.

För att vidare underlätta för användaren att hitta sökta platser skapades en sökfunktion som aktiveras via den utfällbara menyn, eller med en fysisk sökknapp om en sådan existerar. Användaren ombeds att skriva in namnet på den sökta platsen, till exempel "HA1" som är en föreläsningssal på campus Johanneberg, i sökfältet. Efter att ha skrivit in minst en bokstav, får han eller hon en lista med matchande platser på båda campusområden. Vid tryck på ett av resultaten i listan centreras kartvyn på den platsen, och en informationsbubbla visas ovanför.

Efter att ha skrivit in någon eller några bokstäver i sökfältet kan användaren, istället för att trycka på ett av de matchande resultaten i listan, trycka på Enter-knappen eller "sök"-knappen i sökrutan. Då visas markörer över platser vars namn matchar söksträngen, och en informationsbubbla visas över den plats som ligger närmast den position som kartvyn är centrerad på, se figur 6.9. Om inga namn matchar söksträngen, visas ett kort meddelande upp för användaren.



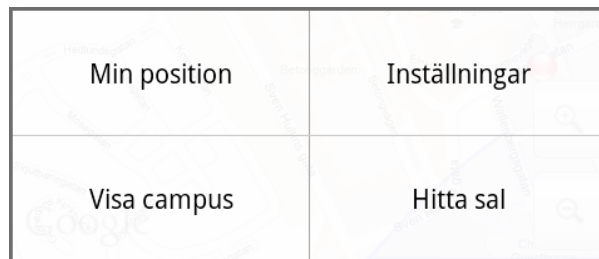
Figur 6.9: Bilden visar resultat av sökning med söksträngen "HA", där alla föreläsningssalar med "HA" i namn markeras med platsmarkörer, och en informationsbubbla visas över den hittade platsen som ligger närmast mitten av kartvyn. I nedre vänstra hörnet på skärmen syns knappen vars funktion är att rensa sökresultat.

När sökmarkörerna är synliga visas ytterligare en knapp i det nedre vänstra hörnet på skärmen, med funktion att rensa sökresultat. När användaren har tryckt på denna knapp, försvinner alla sökmarkörer och även själva knappen. Så länge knappen är synlig har även den

6. Resultat

fysiska tillbakaknappen denna funktion, det vill säga den tar bort sökmarkörerna och 'Rensa sökresultat'-knappen istället för att stänga ner kartan.

Kartfunktionen innehåller även en utfällbar meny som fälls ut med hjälp av den fysiska menyknappen, se figur 6.10. Menyn i kartvyn innehåller fyra alternativ: 'Min position', 'Inställningar', 'Visa campus' och 'Hitta sal'. Det första alternativet centrerar kartvyn på användarens position, och 'Visa campus' centrerar vyn på det campus som har valts i inställningarna. Alternativet 'Inställningar' öppnar aktiviteten Preferences som beskrivs närmare i kapitel 6.11, medan det sista alternativet aktiverar sökfunktionen.



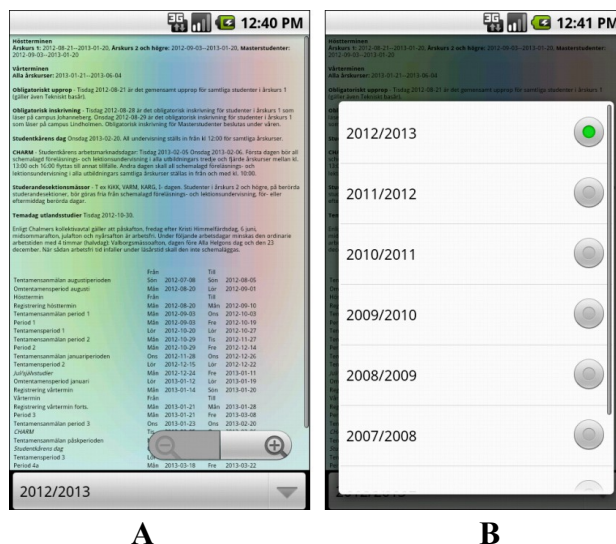
Figur 6.10: Bilden visar knapparna i den utfällbara menyn som aktiveras vid tryck på den fysiska menyknappen i kategorin 'Karta'.

Ifall platstjänster är avstängda och telefonens geografiska position inte är känd, visar ett tryck på knappen 'Min position' upp en informationsruta där användaren meddelas om att platstjänsterna är avstängda. Därefter ges användaren möjligheten att ändra platstjänsternas inställningar, vilket bläddrar vyn till systemets inställningsmeny för platstjänster. Där kan användaren aktivera eller inaktivera hämtning av telefonens position via mobila nätverk och GPS.

6.8. Viktiga datum

Kategorin 'Viktiga datum' presenterar information om läsårstider på Chalmers som hämtas från Studieportalens webbsida Läsårstider. Funktionen visar information som består av två delar, en löpande text och en tabell, med bibehållen formatering, se figur 6.11 A. Allt irrelevant som till exempel meny och länkar till andra webbsidor filtreras bort.

6. Resultat

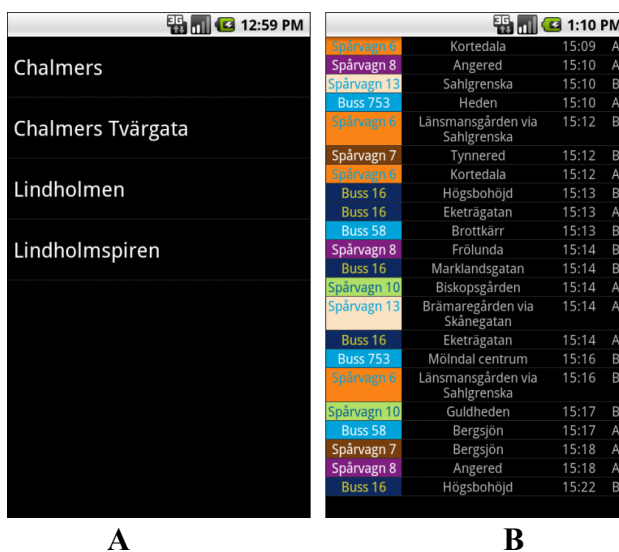


Figur 6.11: Bild A visar viktig information för läsåret 2012/2013 och bild B visar menyn med vilka läsår som kan väljas.

Funktionen erbjuder, precis som Läsårstider på Studieportalen, möjligheten att kunna välja vilket läsår som ska visas, se figur 6.11 B. Om inget val görs visas nuvarande läsår automatiskt.

6.9. Resa

När användaren trycker på kategorin 'Resa' listas fyra hållplatser som ligger på eller i närheten av Chalmers två campus. Hållplatserna som listas är Chalmers, Chalmers Tvärgata, Lindholmen och Lindholmospiren, se figur 6.12 A. När hållplats har valts visas de tjugo närmaste avgångarna med linjenamn, destination, avgångstid samt läge, se figur 6.12 B. Avgångstiderna som visas är realtidsdata i de fall som denna finns tillgänglig, i annat fall används tidtabellen.

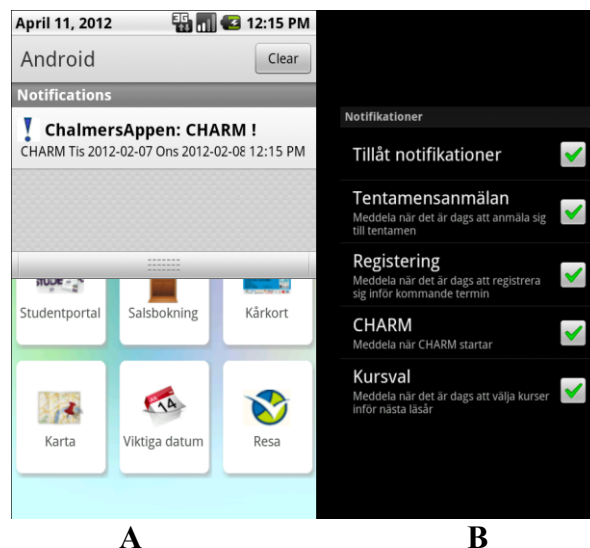


Figur 6.12: Bild A visar de olika hållplatserna som kan väljas i kategorin 'Resa'. Bild B visar hur det kan se ut när Chalmers är vald som hållplats.

6.10. Notifikationer

I telefonens statusfält meddelas användaren om viktiga händelser från Studieportalens (student.chalmers.se) tabell för läsårtider. Vid inkommande påminnelser visas texten "ChalmersAppen" för att indikera vilken applikation de kommer från. Där visas även vilken typ av påminnelse det är samt dagens datum, se figur 6.13 A. Vid tryck på meddelandet i statusfältet försvinner påminnelsen.

För att sätta på notifikationstjänsten kan användaren välja att markera kryssrutan 'Tillåt notifikationer' i ChalmersAppens inställningar, se figur 6.13 B. Användaren kan även välja vilka viktiga händelser som han eller hon vill bli påmind om. Dessa händelser är tentamensanmälan, terminsregistrering, CHARM och kursval inför höstterminen. För att spara på telefonens resurser kan notifikationstjänsten stängas av helt genom att avmarkera kryssrutan 'Tillåt notifikationer'. Då stängs tjänsten av och inga händelser kontrolleras oavsett om dessa är markerade eller inte.



Figur 6.13: Bild A visar telefonens statusfält med en notifikation. Bild B visar applikationens inställningar där användaren kan välja vilka påminnelser han eller hon är intresserad av.

6.11. Den utfällbara menyn

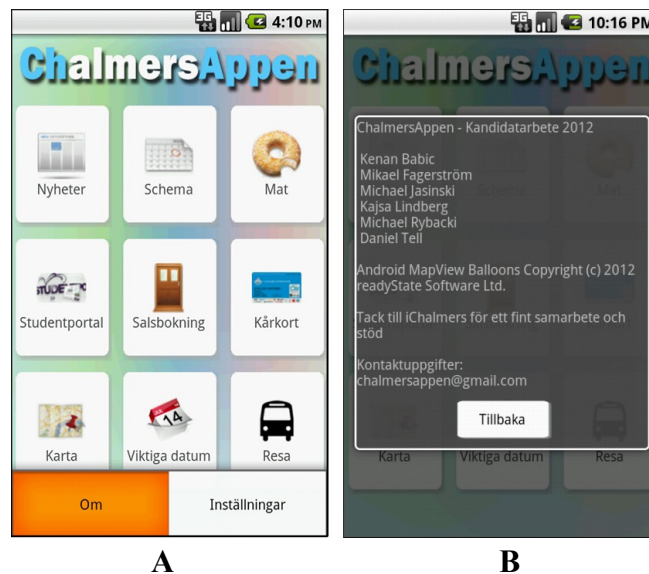
Under alla kategorier i ChalmersAppen finns det en meny som fälls ut vid tryck på telefonens fysiska menyknapp. Från menyn kan användaren nå applikationens inställningar genom att välja 'Inställningar'. Där kan användaren ställa in vilket campus som ska användas som standard, ange CID med tillhörande lösenord samt ändra inställningar för notifikationsfunktionen, se figur 6.14.

6. Resultat



Figur 6.14: Bilden visar applikationens inställningar.

Applikationens huvudmeny innehåller, förutom 'Inställningar', menyalternativet 'Om', se figur 6.15 A. Detta alternativ visar en vy som presenterar ChalmersAppen och dess utvecklare, se figur 6.15 B. Längst ner i vyn visas även kontaktinformation till utvecklarna i form av en gemensam e-postadress, som är tänkt att användas för återkoppling från applikationens användare. Denna återkoppling kan innefatta buggrapportering samt förslag på framtida uppdateringar. Vyn visar även en knapp 'Tillbaka' som tar användaren tillbaka till applikationens huvudmeny.



Figur 6.15: Bild A visar huvudmenyn med valt menyalternativ 'Om' i den utfällbara menyn. Vid tryck på den presenteras utvecklarna av applikationen som visas på bild B.

7. Diskussion

Den första versionen av ChalmersAppen skulle vara lättanvänd och ha en tilltalande grafisk design. Detta mål uppnåddes genom att applikationens alla kategorier kunde förse med funktionalitet samt att designen anses ge ett sådant intryck.

Innan implementationen av applikationen påbörjades bestämdes ett antal mål som beskrev slutresultatet av varje kategori och funktion. Under utvecklingens gång realiserades de flesta målen. Vissa kategorier och funktioner blev bättre än vad som förväntades. En av dessa är kategorin 'Karta'. Den innehåller informationsbubblor som visas när användaren trycker på platsikoner och en fullkomligt implementerad sökfunktion. Platsikoner som placerades ut på kartvyn kommer att underlätta för användare att hitta platsernas positioner på Chalmers båda campus.

En funktion som inte fanns med bland de ursprungliga målen var notifikationerna. Funktionen förväntas underlätta studietiden för användarna eftersom de kommer att få automatiska påminnelser om händelser som de i vanliga fall måste hålla reda på själva.

Trots en fullt fungerande applikation finns det fortfarande utrymme för förbättringar och vidareutveckling av mer funktionalitet. Under utvecklingens gång uppstod också vissa problem som gjorde att all funktionalitet inte kunde realiseras fullt ut. Dessa beskrivs närmare i nästföljande kapitel.

7.1. Problem vid implementation

Ett av de större problemen med utvecklingen av ChalmersAppen var att implementera kategorin 'Studentportal'. Den planerades att vara den mest omfattande av kategorin. Målet var att logga in användaren med CID och lösenord och sedan visa utvald information, parsad från Studentportalen. Kategorin kunde dock inte slutföras på grund av implementationssvårigheter som uppstod. Om den planerade funktionaliteten under 'Studentportal' hade lyckats hade applikationen blivit mer personligt anpassad. Mycket tid spenderades på att angripa denna utmaning men trots det kunde målet inte uppnås. Under försöken implementerades två fungerande lösningar i ett vanligt Javaprojekt, men dessa gick inte att överföra till Android. Orsaken till detta är fortfarande okänd, dock kan det ha berott på att ett av de använda biblioteken inte var kompatibelt med Android.

En lyckad inloggningsfunktionalitet hade medfört att användarens personliga information hade kunnat visas på ett sätt som vore bättre anpassat för en mobilskärm än vad Chalmers Studentportal är. Kategorin skulle ha bestått av tre underkategorier: 'Tentor', 'Kurser' och 'Resultat' med tillhörande underfunktioner, se appendix B. Ytterligare fördelar hade varit att med hjälp av de parsade kursresultaten hade studentens betygsmedelvärde kunnat beräknas och applikationen hade kunnat skicka notifikationer vid resultatregistrering. Vidare hade de registrerade kurserna för pågående termin kunnat användas till att visa ett personligt anpassat

7. Diskussion

schema för varje student. En nackdel med detta sätt vore att användaren skulle bli begränsad till dessa utvalda funktioner.

7.2. Förarbete

Trots mycket förarbete hade detta kunnat vara ännu mer utförligt bland annat i form av en tydligare och mer riktad enkät med ett större urval respondenter. Fördelen med detta hade varit att få en mer övergripande överblick på vilka funktioner som faktiskt eftertraktades av studenterna på Chalmers. Däremot fick de 33 respondenterna, som svarade på enkäten ChalmersAppen använde, stor frihet att välja vilka funktioner de ville ha med, vilket kan vara både en fördel och en nackdel. En fördel kan vara att mer variation och idérikedom kan fångas upp medan en nackdel kan vara att de svarande inte vet vilka funktioner som skulle kunna tänkas ingå i en applikation.

Även en mer detaljerad kravspecifikation hade kunnat förenkla arbetet och utförandet av ChalmersApplikationen, dels för att det hade varit lättare att följa de olika iterationer som utfördes i utvecklandet, dels för att det hade gett ett bättre upplägg på källkoden. Trots att detta hade förenklat arbetet har ChalmersAppen ändå blivit förhållandevis bra, eftersom vissa delar blev bättre än förväntat medan andra inte riktigt blev som de var tänkta från början utan fick förenklas.

7.3. Relaterade arbeten

Innan kandidatarbetets början fanns det vetskap om en Chalmersrelaterad applikation till smarttelefoner vid namn iChalmers. Denna applikation var endast riktad till operativsystemet iOS, men efter kandidatarbetets början framkom det att det även fanns en Chalmersapplikation till Android, vid namn Studentport. Detta kom att påverka arbetet med ChalmersAppen som var inriktad på samma operativsystem. ChalmersAppen behövde leverera mer och bättre funktionalitet för att ha möjlighet att konkurrera med Studentport. Detta kunde uppfyllas på ett bättre sätt än vad som från början ansågs möjligt. Dock har både Studentport och iChalmers funnits på marknaden sedan 2011 respektive början av 2012 vilket har gett dem ett stort försprång i funktionalitet, så som att de båda har lyckats med inloggning till Chalmers studentportal.

7.4. Framtida arbete

Ett delsyfte med kandidatarbetet var, som tidigare nämnt, att lämna över applikationen till tredje part för vidareutveckling. Under arbetets gång har det diskuterats ifall applikationen ska lämnas över direkt eller om den ska fortsätta att utvecklas. Dessa diskussioner har uppstått efter att ett flertal förbättringsförslag och idéer har uppkommit.

En nackdel med att lämna över applikationen skulle vara att dessa idéer inte skulle realiseras lika snabbt, eller att de inte skulle förmedlas vidare på rätt sätt. En tredje part är inte heller

7. Diskussion

lika insatt i källkodens upplägg och har inte funderat lika mycket på vilken typ av funktionalitet som skulle kunna ingå.

En fördel med att överlämna applikationen skulle däremot kunna vara om utvecklarna har tillgång till databaser och intern information som skulle kunna effektivisera och förbättra befintlig funktionalitet.

7.4.1. Förslag på utökad funktionalitet

Det finns mycket funktionalitet som det inte har varit möjligt att implementera i denna version av ChalmersAppen. Nedanför presenteras ytterligare funktionalitet som skulle kunna implementeras i framtida versioner av applikationen.

Funktionaliteten för schemat skulle kunna utökas genom att visa schemat för användarens pågående kurser eller att låta användaren spara flera olika scheman. Schemat skulle även kunna sparas på telefonens minneskort för att kunna nås utan nätverksåtkomst. Vidare skulle applikationen kunna tillhandahålla möjligheten att flyttas till SD-kortet, vilket är av betydelse ifall användarens smarttelefon inte har tillräckligt internminne. Ytterligare skulle schemat kunna sammanlänkas med kategorin 'Karta' och visa den aktuella dagens föreläsningar och salar på kartan och även få notifikationer om dessa.

För kategorin 'Salsbokning' skulle det vara möjligt att skapa ett grafiskt användargränssnitt för att genomföra salsbokningar. Detta skulle öka användarvänligheten då webbsidorna som visas inte är mobilanpassade. Även kategorin 'Viktiga datum' skulle gynnas av en liknande lösning. Kategorin 'Kårkort' visar en mobilanpassad webbsida, men användarupplevelsen skulle ökas ifall funktionaliteten hanterades helt inom applikationen.

För att minska antalet knapptryckningar som krävs för att nå avgångarna i kategorin 'Resa' skulle möjligheten att spara en standardhållplats kunna införas. Fördelen med detta hade varit att användaren skulle nå den sökta funktionen fortare, vilket ökar användarvänligheten.

8. Slutsats

Den första versionen av ChalmersAppen färdigställdes och fungerar till stor del som planerat. Mycket av den planerade funktionaliteten har implementerats, och det kan konstateras att stora delar av denna kan underlätta studietiden för studenter enligt kandidatarbetets delmål.

De flesta utvecklingsmålen uppfylldes. Den främsta anledningen till att somliga mål inte uppfylldes var brist på tid. Framför allt saknades tid till att lösa de tekniska problem som uppstod vid utvecklingen av kategorin ‘Studentportal’, där svårigheterna vid implementation av inloggningsfunktionen förhindrade realisering av funktionen.

Slutligen planeras överlämning av ChalmersAppen till tredje part för vidareutveckling, om intresse finns.

Källförteckning

Android (2012a) *Develop Apps*. <http://www.android.com/> (2012-05-08).

Android (2012b) *Discover Android*. <http://www.android.com/> (2012-05-08).

Android Developers (2012a) What is Android?. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012b) Application Fundamentals. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012c) Building and Running. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012d) Tasks and Back Stack. *Dev Guide*. <http://developer.android.com/> (2012-05-10).

Android Developers (2012e) Bug fixes. *SDK*. <http://developer.android.com/> (2012-05-10).

Android Developers (2012f) Android API Levels. *Dev Guide*. <http://developer.android.com/> (2012-05-10).

Android Developers (2012g) Android 3.0 Platform Highlights. *SDK*. <http://developer.android.com/> (2012-05-10).

Android Developers (2012h) Introduction. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012i) Managing Virtual Devices. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012j) The AndroidManifest.xml File. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012k) Using DDMS. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012l) Activity. *Reference*. <http://developer.android.com/> (2012-05-09).

Android Developers (2012m) BroadcastReceiver. *Reference*. <http://developer.android.com/> (2012-05-09).

Källförteckning

Android Developers (2012n) Content Provider. *Reference*. <http://developer.android.com/> (2012-05-09).

Android Developers (2012o) Construct the UI. *Reference*. <http://developer.android.com/> (2012-05-09).

Android Developers (2012p) Android Developer Tools. *Dev Guide*. <http://developer.android.com/> (2012-05-08).

Android Developers (2012q) Platform Versions. *Resources*. <http://developer.android.com/> (2012-05-03).

Android Developers (2012r) public class GridView. *Reference*. <http://developer.android.com/> (2012-05-12).

Android Developers (2012s) public class WebView. *Reference*. <http://developer.android.com/> (2012-05-12).

Android Developers (2012t) public class ListView. *Reference*. <http://developer.android.com/> (2012-05-13).

Android Developers (2012u) The AndroidManifest.xml File: <activity>. *Dev Guide*. <http://developer.android.com/> (2012-05-13).

Android Developers (2012v) Adding Custom Suggestions. *Dev Guide*. <http://developer.android.com/> (2012-04-22).

Android Developers (2012w) public class CheckBoxPreference. *Reference*. <http://developer.android.com/> (2012-05-13).

Blåberg, E.B. (2011) Versionshantering. *Chalmers Insidan*. <http://www.chalmers.se/insidan/> (2012-05-10).

Cassavoy, L. (2012a) What makes a smartphone smart?. *About.com*. http://cellphones.about.com/od/smartphonebasics/a/what_is_smart.htm (2012-05-08).

Cassavoy, L. (2012b) What is a smartphone?. *About.com*. http://cellphones.about.com/od/glossary/g/smart_defined.htm (2012-05-08).

Chalmers Studentkår (2010) Kåravgift. <http://www.chs.chalmers.se/sv/Detta-aer-kaaren/Medlemskap/Kaaravgift> (2012-05-04).

Conder, S. och Darcey, L. (2010) *Android Wireless Application Development*. Boston, MA: Pearson Education.

Källförteckning

- Crockford, D. (2006) The application/json Media Type for JavaScript Object Notation (JSON). *Network Working Group*. <http://tools.ietf.org/html/rfc4627> (2012-05-12).
- Ehringer, D. (2010) The Dalvik Virtual Machine Arcitecture. [Elektronisk]. http://davehringer.com/software/android/The_Dalvik_Virtual_Machine.pdf (2012-05-10).
- Exforsys Inc. (2007) XML Parsing. <http://www.exforsys.com/tutorials/xml/xml-parsing.html> (2012-05-10).
- Franklin, S. (2010) XML Parsers: DOM and SAX Put to the Test. *devx*. <http://www.devx.com/xml/Article/16922/1954> (2012-05-10).
- Gargenta, A. (2011a) Android Builders Summit - A Walk through the Android Stack. [YouTube]. http://www.youtube.com/watch?v=ygP3_1Lzrj4 (2012-05-09).
- Gargenta, M. (2010) Learn about Android Internals and NDK. [YouTube]. <http://www.youtube.com/watch?v=hcXIchfpHfU> (2012-05-10).
- Gargenta, M. (2011b) *Learning Android*. Sebastopol, CA: O'Reilly Media.
- Gargenta, M. (2011c) AnDevCon: Android for Java Developers - Marko Gargenta, Pt. 2. [YouTube]. <http://www.youtube.com/watch?v=hcXIchfpHfU> (2012-05-09).
- Gargenta, M. (2011d) AnDevCon: Android for Java Developers - Marko Gargenta, Pt. 1. [YouTube]. <http://www.youtube.com/watch?v=hcXIchfpHfU> (2012-05-09).
- Gartner (2012) Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth. *Gartner Newsroom*. <http://www.gartner.com/it/page.jsp?id=1924314> (2012-05-08).
- Gilfelt, J. (2012) Android MapView Ballons. *github*. <https://github.com/jgilfelt/android-mapviewballoons>. (2012-05-09).
- Google Inc. (2010) Android 2.1 Compatibility Definition. *Android compatibility program*. <http://developer.android.com/> (2012-05-12).
- Gustafsson Kojo, B., Davidsson, P. och Fransén, K (2011). *Svensk Telemarknad första halvåret 2011*. Stockholm: Post- och telestyrelsen. (Rapportnummer: PTS-ER-2011:21).
- HaCon (2011) *HAFAS Public Access Gateway: Easy access to HAFAS journey planner systems*. Hannover: HaCon Ingenieurgesellschaft mbH. (Specifikation: Version 1.04).
- Hedley, J. (2012) jsoup: Java HTML Parser. *jsoup*. <http://jsoup.org/> (2012-05-12).

Källförteckning

JSON (2012) *Introducing JSON*. <http://www.json.org/> (2012-05-09).

Meier, R. (2009) *Professional Android™ Application Development*. Indianapolis, IN: Wiley Publishing.

Murphy, M. L. (2008) *The Busy Coder's Guide to Android Development*. [Elektronisk] CommonsWare LLC.

Nations, D. (2012) What is iOS?. *About.com*. <http://ipad.about.com/od/iPad-Glossary/g/What-Is-iOS.htm> (2012-05-08).

Pilato, C. M., Collins-Sussman, B. och Fitzpatrick, B. W. (2008) *Version Control with Subversion*. Upplaga 2. Sebastopol: O'Reilly Media.

Rodriguez, A. (2008) RESTful Web services: The basics. *developerWorks*. <https://www.ibm.com/> (2012-05-09).

Singh, I., Leitch, J. och Wilson, J. (2012) google-gson. <http://code.google.com/p/google-gson/> (2012-05-12).

StatCounter (2012a) Top 8 Mobile Operating Systems in Sweden from Jan to Dec 2010. *StatCounter Global Stats*. http://gs.statcounter.com/#mobile_os-SE-monthly-201001-201012-bar (2012-05-08).

StatCounter (2012b) Top 8 Mobile Operating Systems in Sweden from Jan to Dec 2011. *StatCounter Global Stats*. http://gs.statcounter.com/#mobile_os-SE-monthly-201101-201112-bar (2012-05-08).

Touch Topics (2012) What is Multi-touch?. *Helping Define Technologies, Terms, and Solutions for the Touch Industri*. http://solutions.3m.com/wps/portal/3M/en_US/TouchTopics/Home/Terminology/WhatIsMultiTouch/ (2012-05-10).

Vidgen, R. och Goodwin, S. (2000) XML: what is it good for?. *Computing & Control Engineering Journal*, vol. 11, nr 3, ss. 119-124.

W3Schools (2012a) Introduction to SQL. *w3schools.com*. http://www.w3schools.com/sql/sql_intro.asp (2012-05-09).

W3Schools (2012b) RSS Syntax. *w3schools.com*. http://www.w3schools.com/rss/rss_syntax.asp (2012-05-10).

W3Schools (2012c) HTML 4.01 / XHTML 1.0 Reference. *w3schools.com*. <http://www.w3schools.com/tags/default.asp> (2012-05-31).

Källförteckning

WhatIsRSS (2012) RSS Primer: One Page Quick Introduction to RSS. *What is RSS? RSS Explained*. <http://www.whatissrss.com/> (2012-05-10).

Zechner, M. (2011) *Beginning Android Games*. Berkeley, CA: Apress.

Appendix

A. Enkätundersökning

A1. Enkät

ChalmersAppen - undersökning

1. Skulle du kunna tänka dig att ladda ner en Chalmers app för Android om det fanns en sådan?

☐ Ja

☐ Vet ej

☐ Nej

☐ Det beror på appens utformning (kommentera gärna)

Ev. kommentar:

2. Om JA på fråga 1: Nämn 3 stycken funktioner som du skulle vilja ha i en sådan app (i prioritetsordning):

a.

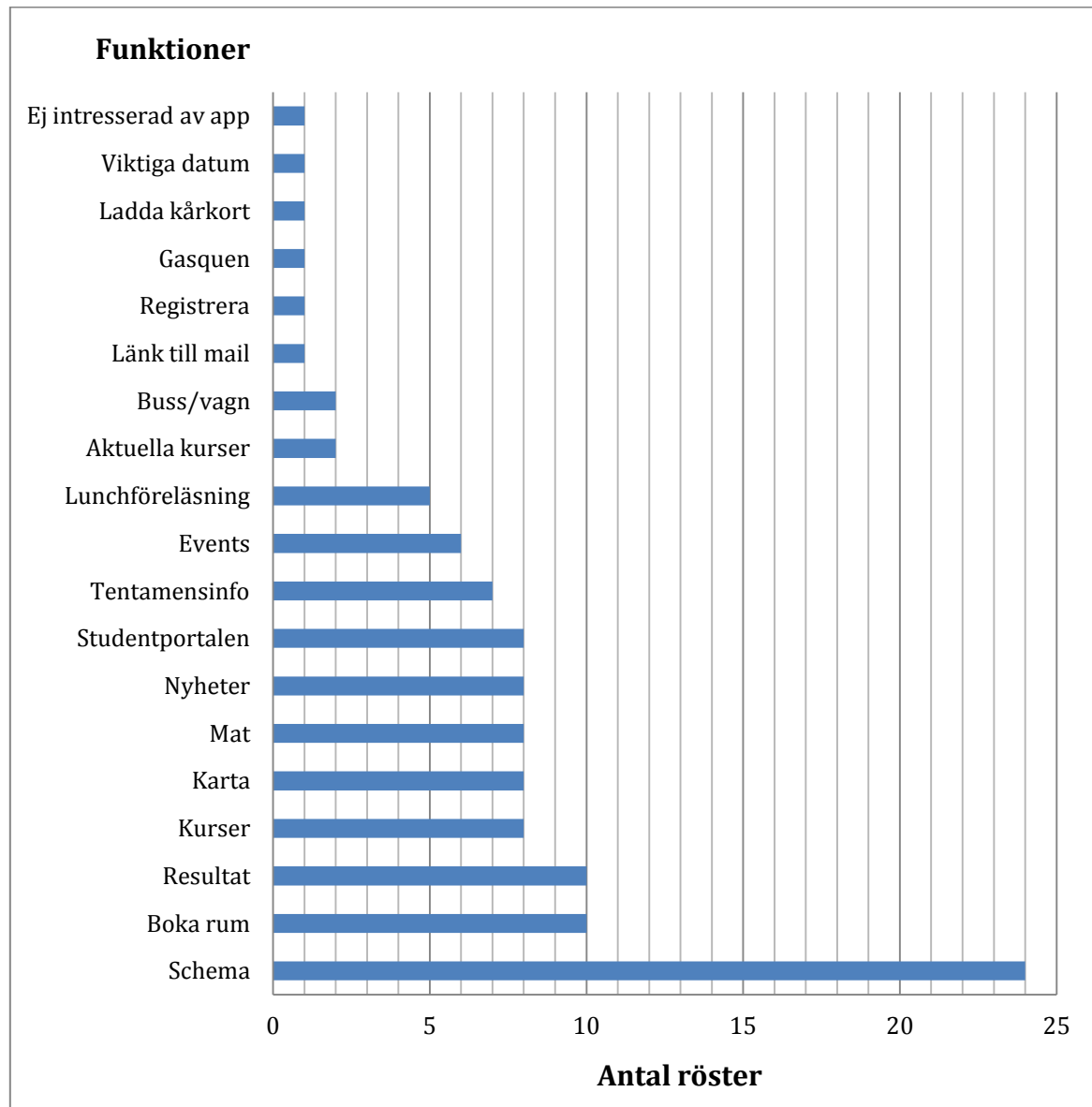
b.

c.

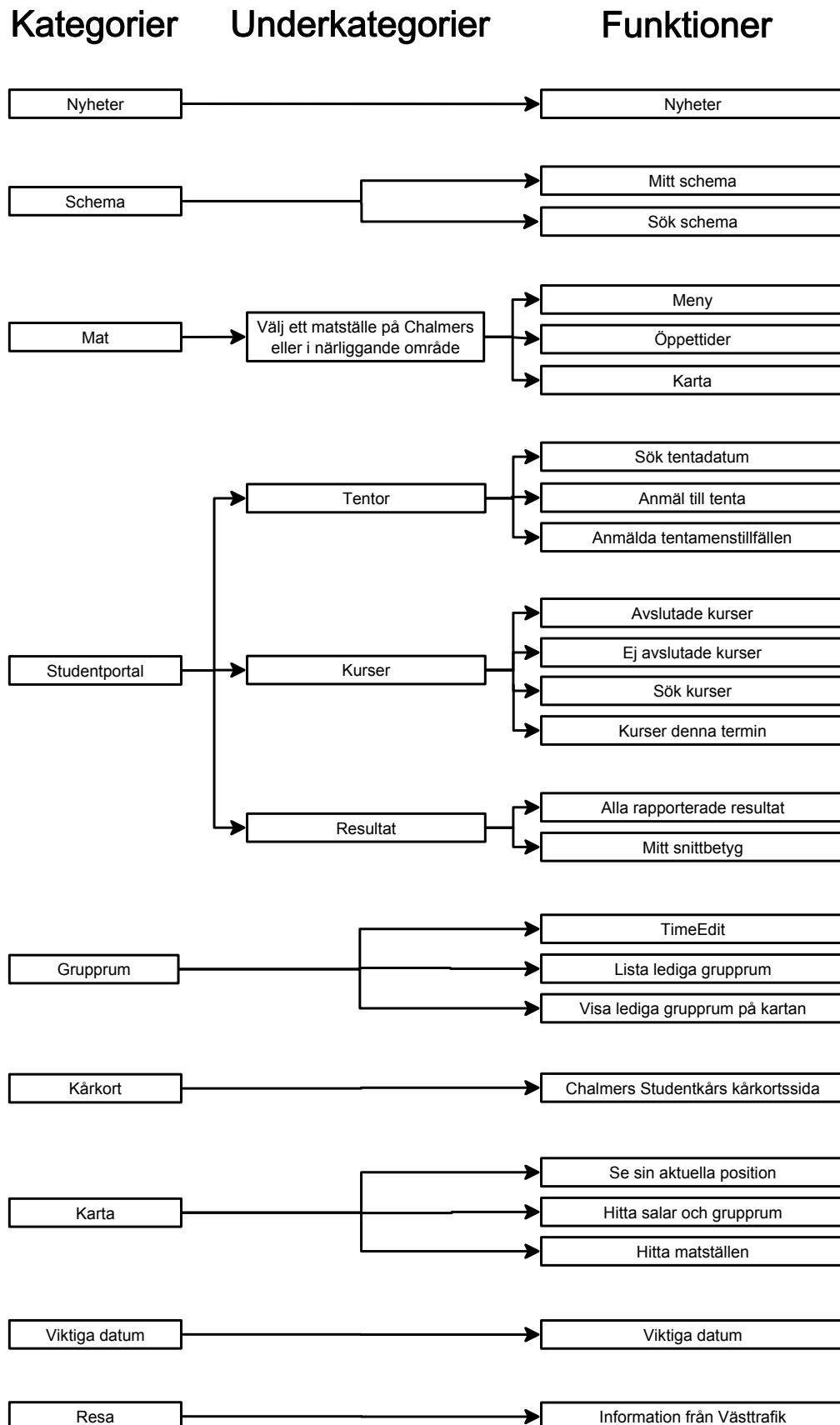
3. Om NEJ på fråga 1: Varför skulle du inte vilja ha en sådan app?

TACK FÖR MEDVERKAN ! 😊

A2. Resultat av enkätundersökning



B. Tänkt funktionalitet



C. Kravspecifikation

Innehållsförteckning

Målsättning

Funktionella krav

Förklaring av prioritering

1. HomeActivity - ChalmersAppen startsida
2. NewsActivity - Nyheter
3. ScheduleActivity - Schema
4. LunchActivity - Mat
5. StudentPortalActivity - Studentportal
6. BookingActivity - Salsbokning
7. UnionCardActivity - Kårkort
8. MapViewActivity - Karta
9. ImportantDatesActivity - Viktiga Datum
10. TrafficActivity - Resa
11. AboutUsActivity - Om
12. Preferences - Inställningar
13. LoginActivity - Inloggning
- I. Implementationskrav

Icke-funktionella krav

Programmeringsregler

Målsättning

Målsättningen är att skapa en applikation för Androidbaserade smarta telefoner som ska underlätta studierna för studenter vid Chalmers Tekniska Högskola. Syftet med denna kravspecifikation är att beskriva hur applikationens funktionalitet samt grafiska gränssnitt ska utformas. Applikationen ska följa de funktionella kraven som specificeras i kravspecifikationen. Dessa kan komma att uppdateras med tiden beroende på hur mycket som gruppen hinner med att implementera.

Funktionella krav

Förklaring av prioritering

- 1: Definitivt måste göras innan kandidatarbetets slut
- 2: Måste göras innan appen släpps ut
- 3: Görs endast i mån av tid
- 4: Görs eventuellt när allting med prio 3 är klart

1. HomeActivity - ChalmersAppen start sida

Nr	Kravbeskrivning	Prio	Status	Ansvar
1.1	Startsidan ska bestå av en rubrik "ChalmersAppen" och nio stycken knappar: Nyheter, Schema, Mat, Studentportal, Salsbokning, Kårkort, Karta, Viktiga Datum och Resa. Rubriken ska bestå av en bild som ska placeras överst och vara centrerad. Knapparna ska radas upp från vänster till höger i ett 3x3-rutnät i den ordningen som de presenteras ovan. Knapparna ska vara vita och bestå av en ikon och ett namn. Menyknappen som finns på Androidtelefoner ska visa en utfällbar meny. Interfacet i huvudmenyn ska använda GridView.	1	Klart	Jasinski och Rybacki
1.2	Knappen "Nyheter" ska starta en ny aktivitet som visar upp RSS-flöden från www.chalmers.se , www.chs.chalmers.se (Studentkårens hemsida) och www.lib.chalmers.se (Chalmers bibliotek).	1	Klart	Jasinski och Kenan
1.3	Knappen "Schema" ska starta en ny aktivitet med knapparna "Mitt Schema" och "Sök Schema".	1	Ej genomfört till version 1.0	Fagerström och Kajsa
1.4	Version 4: Knappen "Mat" ska starta en ny aktivitet som visar matställena "Kårrestaurangen", "R.F Byssja", "Hyllan" och "Linsen" om det valda campuset är Johanneberg. Om Lindholmen är valt, ska matställena "L's Kitchen", "Restaurang R" och "Kokboken" visas.	2	Klart	Jasinski
1.5	Knappen "Studentportal" ska starta en ny aktivitet som visar en lista med alternativen "Tentor", "Kurser" och "Resultat". Om användaren inte är inloggad, ska användaren ombes ange sina inloggningssuppgifter.	1	Ej genomfört till version 1.0	Jasinski
1.6	InloggningsActivity ska visa två fält, "CID" och "lösenord", samt en kryssruta, "spara lösenord" och en login-knapp. Om kryssrutan kryssas, ska lösenordet (krypterat) sparas i mobilen, dock borde användaren förvarnas om medföljande säkerhetsproblem genom en informationsruta. Om systemet godtar användarens CID och lösenord, omdirigeras han eller hon till aktiviteten som nämndes ovan. Annars ska en informationsruta informera användaren om att inloggningen misslyckades. Därefter ombedes han eller hon att försöka igen.	1	Ej genomfört till version 1.0	Jasinski och Kenan
1.7	Knappen "Grupprum" ska starta en ny aktivitet som länkar till TimeEdit:s bokningssystem med hjälp av en WebView.	2	Klart	Kenan och Kajsa

Appendix

1.8	Knappen "Kårkort" ska starta en ny aktivitet som länkar till den mobila hemsidan för kårkortsinformation http://kortladdning.chalmerskonferens.se/mobil.htm med hjälp av en WebView.	2	Klart	Kenan
1.9	Knappen "Karta" ska starta en ny aktivitet där kartvyn presenteras. Kartan ska implementeras med Google Maps API och innehålla samtliga salar på Chalmers samt låta användaren söka salar.	1	Klart , dock är inte alla salar på Chalmers tillagda	Rybacki
1.10	Knappen "Viktiga Datum" ska starta en ny aktivitet som länkar till adressen https://www.student.chalmers.se/sp/academic_year_list med hjälp av en WebView.	1	Klart	Fagerström och Kajsa
1.11	Knappen "Resa" ska starta en ny aktivitet som visar alla spårvagns och buss -avgångar från hållplatserna "Chalmers", "Chalmers Tvärgata" och "Lindholmen". Detta ska implementeras med Västtrafiks API.	3	Klart	Tell
1.12	Den utfällbara menyn ska visa alternativen "Om", "Språk", "Inställningar" och "Tema". Alternativet "Om" ska visa en ruta där kandidatgruppen samt annan relevant information presenteras kortfattat. Alternativet "Inställningar" ska starta en PreferenceActivity där användaren kan välja att spara sitt lösenord och/eller användarnamn för automatisk inloggning. "Tema" ska låta användaren välja bakgrund för applikationen.	1	Kvar: - Språk - Tema	Jasinski
1.13	Man ska kunna ställa in notifications så att dom visas när man t.ex. har en kommande tenta eller nåt sånt. Användaren ska kunna sätta på och stänga av notifikationer av tentamensanmälan, terminsregistrering, CHARM och kursval inför hösten. Informationen plockas från viktiga datum.	3	KLART	Jasinski

2. NewsActivity - Nyheter

Nr	Kravbeskrivning	Prio	Status	Ansvar
2.1	Nyheter ska presenteras i en Tab Layout, där fokus vid start ges till fliken med nyheterna från chalmers.se.	1	Klart	Jasinski och Kenan
2.2	Nyheterna hämtade från RSS-flöden ska visas i en lista.	1	Klart	Jasinski och Kenan
2.3	Man ska kunna se RSS-flöden från Chalmers (http://www.chalmers.se/sv/nyheter//_layouts/Chalmers.Core.UI/RSS.ashx?listId={ec38b007-b740-438d-a6de-4b8b8f4ea8f3}), från kåren (http://www.chs.chalmers.se/) samt från biblioteket (http://www.lib.chalmers.se/rss.xml)	1	Klart	Jasinski och Kenan
2.4	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

3. ScheduleActivity - Schema

Nr	Kravbeskrivning	Prio	Status	Ansvar
3.1	Schemat ska visas i ett WebView-element	1	Klart	Jasinski
3.2	Det ska finnas en knapp "Sök schema" som länkar till " https://web.timeedit.se/chalmers_se/db1/public/r.html?sid=3&h=t " i samma WebView som schemat visas i.	1	Klart	Jasinski
3.3	Det ska finnas en knapp 'Spara schema' som låter användaren spara det aktuella schemat som visas i WebView-elementet.	1	Klart	Jasinski
3.4	Det ska finnas en knapp 'Visa sparad schema' som visar det sparade schemat i WebView-elementet.	1	Klart	Jasinski
3.5	När användaren startar kategorin ska det sparade schemat visas automatiskt.	1	Klart	Jasinski
3.6	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

4. LunchActivity - Mat

Nr	Kravbeskrivning	Prio	Status	Ansvar
4.1	En Tab Layout med alternativen "Kårrestaurangen", "R.F Byssja", "Hyllan" och "Linsen" ska visas när det valda campuset i inställningarna är Johanneberg. För valt campus Lindholmen ska "L's Kitchen", "Restaurang R" och "Kokboken" visas.	2	Klart 2012-03-17	Jasinski
4.2	Varje flik ska visa RSS-flödet från respektive restaurang/matställe i en lista.	2	Klart	Jasinski
4.3	En lunchmeny för det valda matstället ska visas, beroende på matstället, i WebView eller som PDF. Under listan ska det finnas en knapp som leder till kartan. Det valda matstället ska visas på kartan, genom att applikationen går över till MapActivity, där kartvyn centreras på matstället, och en informationsbubbla visas ovanför matställets ikon.	3	Knappen existerar men gör inget för tillfället. Måste synkas med MapActivity	Jasinski
4.4	Öppettider för det valda matstället ska visas. Under samma lista ska det finnas en annan knapp som länkar till webbsidan för respektive matställe och visar priser och öppettider. Informationen finns på http://chalmerskonferens.se/	2	Klart 2012-03-17	Jasinski
4.5	Vid tryck på den fysiska menyknappen ska en meny visas med valmöjligheten att visa menyer för campus Johanneberg eller Lindholmen. Menyn ska även länka till applikationens huvudinställningar.	2	Klart	Jasinski

5. StudentPortalActivity - Studentportal

Nr	Kravbeskrivning	Prio	Status	Ansvar
5.1	Alternativen "Tentor", "Kurser" och "Resultat" ska ordnas i en ExpandableView som expanderar en ny meny med underlistorna enligt följande krav.	1	Påbörjad	Tell & Fagerström
5.2	Under "Tentor" ska alternativen "Sök tentamensdatum", "Tentamensanmälan" och "Anmälda tentamentillfällen".	1	Påbörjad	Tell & Fagerström
5.3	Under "Kurser" ska alternativen "Kurser denna termin" "Ej avslutade kurser", "Avslutade kurser" och "Sök kurser" presenteras.	1	Påbörjad	Tell & Fagerström
5.4	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

6. BookingActivity - Salsbokning

Nr	Kravbeskrivning	Prio	Status	Ansvar
6.1	När man klickat på knappen 'Salsbokning' ska man länkas till TimeEdits hemsida (webview), först med webbookning för student för att boka grupprum. https://web.timeedit.se/chalmers_se/db1/b1/ Helst ska rullmenyn redan vara vald som "Student (central inloggning)"	2	Första utkast klart. En WebView visas.	Kajsa och Kenan
6.2	En lista ska visas med alla rum som är lediga just nu.	4	Ej genomfört till version 1.0	Kajsa och Kenan
6.3	Lediga rum ska visas på kartan.	4	Ej genomfört till version 1.0	Kajsa och Kenan
6.4	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

7. UnionCardActivity - Kårkort

Nr	Kravbeskrivning	Prio	Status	Ansvar
7.1	Denna aktivitet ska använda en WebView för att länka vidare till http://kortladdning.chalmerskonferens.se/mobil.htm som är en mobilversion av Chalmers Studentkårs webbsida för kårkortsladdning.	2	Klart 2012-03-16	Kenan
7.2	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

8. MapViewActivity - Karta

Nr	Kravbeskrivning	Prio	Status	Ansvar
8.1	En karta ska visas med hjälp av MapActivity från Google Maps API.	1	Klart 2012-03-14	Rybacki
8.2	Kartvyn ska innehålla en centreringsknapp som vid klick centrerar vyn på användarens nuvarande position (knappen kan även placeras utanför kartvyn, t.ex. i en tab längst ner eller i en utfällbar meny).	1	Klar 2012-03-16	Rybacki
8.3	Samtliga salar/rum på Johanneberg och Lindholmen ska läggas upp på kartan som klickbara platsikoner (objekt av klassen OverlayItem).	2	Påbörjad, kommer inte vara färdig i denna version på grund av det stora antalet salar.	Rybacki
8.4	Vid klick på en platsikon ska en informationsbubbla visas ovanför ikonen innehållande platsens detaljer.	2	Klart 2012-03-22	Rybacki
8.5	Kartan ska innehålla en sökfunktion som ska aktiveras med hjälp av den fysiska sökknappen eller vara nåbar från antingen ett sökfält ovanför kartvyn eller från en knapp i den utfällbara menyn. Vid aktivering ska sökfält och virtuellt tangentbord visas för användaren. Medan man skriver i sökfältet ska en lista med resultat visas, t.ex. om användaren skriver "37", visas alla salar/rum vars namn innehåller strängen "37".	1	Klart	Rybacki
8.6	När användaren trycker på ett element i resultatlistan ska kartvyn centreras på den valda platsen och en informationsbubbla visas direkt ovanför platsikonen.	1	Klart	Rybacki
8.7	När användaren skriver en text i sökfältet och trycker på sökknappen ska de matchande platser markeras med en platsmarkör och en informationsbubbla ska visas ovanför platsikonen som tillhör resultatet som ligger närmast mitten av kartvyn.	1	Klart	Rybacki
8.8	Det ska gå att välja olika visningsläge, exempelvis att visa enbart	2	Klart	Rybacki

Appendix

	hörsalar. En knapp ska visas där användaren kommer att kunna välja vilka lager ska visas, ungefär såsom i Google Maps-applikation för Android.		2012-03-20	
8.9	Det ska gå att navigera till olika ställen (vägbeskrivning).	4	Ej genomfört till version 1.0	Rybacki
8.10	Det ska gå att visa sina kommande föreläsningar, övningar etc. för dagen på kartan med angiven tid	3	Ej genomfört till version 1.0	Rybacki
8.11	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Rybacki

9. ImportantDatesActivity - Viktiga Datum

Nr	Kravbeskrivning	Prio	Status	Ansvar
9.1	Funktionen ska visa en tabell med viktiga datum hämtade från <webbadress>. Denna sida ska med hjälp av ett WebView-element länka vidare till (https://www.student.chalmers.se/sp/academic_year_list) som visar läsårstider på Chalmers.	1	Påbörjad	Kajsa & Fagerström
9.2	Användaren ska ha möjlighet att välja år för tabellen i enlighet med det som visas på webbsidan	1	Klart	Kajsa & Fagerström
9.3	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

10. TrafficActivity - Resa

Nr	Kravbeskrivning	Prio	Status	Ansvar
10.1	Kategorin "Resa" ska visa en lista på alla avgångar (bussar och vagnar) från hållplatser nära Chalmers. Listan ska visa information för hållplatserna Chalmers, Chalmers Tvärgata och Lindholmen med hjälp av Västtrafiks API.	3	Klart	Tell
10.2	Listan ska visas i form av en tabell med rubriker buss/spårvagnslinje, riktning (namnet på ändhållplats), nästa avgång (i minuter), avgång därefter (i minuter).	3	Klart	Tell
10.3	Listan ska vara sorterbar beroende på olika fält. Vid tryck på fältets namn, sorteras listan stigande/fallande enligt detta fält.	4	Ej genomförd	Tell
10.4	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

11. AboutUsActivity - Om

Nr	Kravbeskrivning	Prio	Status	Ansvar
11.1	Funktionen ska presentera kandidatprojektets gruppmedlemmar samt tacka iChalmers för samarbetet.	1	Klart	Jasinski

12. Preferences - Inställningar

Nr	Kravbeskrivning	Prio	Status	Ansvar
12.1	Inställningarna ska låta användaren välja campus, sätta på/stänga av påminnelser, spara sitt CID och lösenord (krypterat).	1	Klart	Jasinski och Rybacki

13. LoginActivity - Inloggning

Nr	Kravbeskrivning	Prio	Status	Ansvar
13.1	Aktiviteten LoginActivity ska presentera två texttrutor där användaren kan fylla i sina inloggningsuppgifter.	1	Påbörjad	Tell och Fagerström
13.2	Den utfällbara menyn ska visa alternativet "Inställningar". Vid tryck på "Inställningar" ska en PreferenceActivity startas där användaren ska kunna ändra applikationens inställningar.	1	Klart	Jasinski

I. Implementationskrav

Nr	Kravbeskrivning	Prio	Status	Ansvar
I-1	En SQLite-databas innehållande alla platser som placeras ut på kartan ska skapas. Datan ska alltså inte finnas någonstans i koden.	1	Klart 2012-03-21	Rybacki
I-2	När nätverksanslutning är inaktiv ska användaren få ett felmeddelande och alternativ att ändra sina nätverksinställningar.	1	Klart	Rybacki
I-3	<MapViewActivity> När platstjänsterna är avaktiverade, användaren ska få ett felmeddelande vid försök att se telefonens position på kartan i MapViewActivity.	1	Klart	Rybacki
I-4	<LoginActivity> Lösenordsrutan i LoginActivity ska dölja lösenordet genom att visa prickar istället för bokstäver	1	Påbörjad	Tell och Fagerström

Icke-funktionella krav

- **Användarvänlighet**
 - Varje funktion ska kunna nås inom ett fåtal knapptryck från huvudmenyn.
 - Användargränssnittet ska följa praxis för Android-applikationer.
- **Kapacitet**
 - Eftersom det inte kommer att finnas någon central databas ska applikationen klara av lika många användare som de underliggande webbplatserna.
- **Säkerhet**
 - Systemet ska kryptera användaruppgifter (login, lösenord) som används för att logga in på Studentportalen.
- **Underhållbarhet**
 - Applikationens kod ska vara lätt att sätta sig in, vilket innebär att hänsyn ska tas till programmeringsreglerna.
 - Applikationen ska vara i sådant skick att den ska kunna överlämnas för vidareutveckling.
- **Tillgänglighet**
 - Applikationen ska kunna visa upp viss data utan att Internetuppkopplingen är på. Ett exempel på detta är platser på kartan och sparade schema.
- **Prestanda**
 - Start av aktiviteter ska inte innebära långa väntetider för användaren.
- **Kompatibilitet**
 - Applikationen ska vara kompatibel med alla Android-telefoner med Android-version 2.1 och högre.
 - Applikationens layout ska anpassas till alla möjliga skärmstorlekar.

Programmeringsregler

För att underlätta utvecklingen ges nedan ett antal regler som bör följas vid applikationsutvecklingen. Reglerna ska fungera som stöd vid läsning av kod. De ska även vara ett stöd för den institution som eventuellt tar sig an uppgiften att ta över och underhålla applikationen efter kandidatarbetets slut.

Paket:

- Varje aktivitet och dess tillhörande klasser skall placeras i var sitt paket för att strukturera applikationen och separera dess olika delar.

Klasser:

- Aktiviteter (klasser av typen `Activity`) döps till "`<namn>Activity`", till exempel `HomeScreenActivity`. Undantaget är `Map` som döps till `MapViewActivity` eftersom det finns en klass vid namn `MapActivity` inne i Google Maps API.
- Klasser som inte är av typen `Activity` ska ges ett tydligt och lättförståeligt namn.

XML:

- Layout-dokument i XML-format tillhörande en aktivitet bör döpas till samma namn fast utan ändelsen "`Activity`" och med enbart små bokstäver. Exempelvis layout-dokumentet motsvarande `HomeActivity` bör döpas till "`home.xml`".
- Alla texter som används i applikationen ska deklarerars i `strings.xml` och får **inte** hårdkodas.

Views / Objekt:

- Variabelnamn döps till "`<typ><Namn>`". Ex: Ett `Button`-objekt kan t.ex. ha namnet `buttonNews` och ett `TextView`-objekt `textViewNews`.
- Alla ID för `View`-objekt ska ha samma namn som variabeln som pekar på `View`-objektet i Javakoden. Till exempel, en knapp vars variabel i Javakoden heter `buttonNews` ska ha ID:t `android:id="@+id/buttonNews`.
- Alla objekt ska ha engelska namn, det vill säga `buttonNews` och inte `buttonNyheter`.

Kommentarer:

- Alla metoder, publika och privata, skall innehålla Javadoc-kommentarer. Detta underlättar för övriga gruppmedlemmar samt andra att förstå koden.
- Kommentarer skall skrivas i koden, dock endast där det är nödvändigt, det vill säga när koden är svår att förstå vid första anblick. Operationer som anses vara elementära bör därmed inte förklaras.