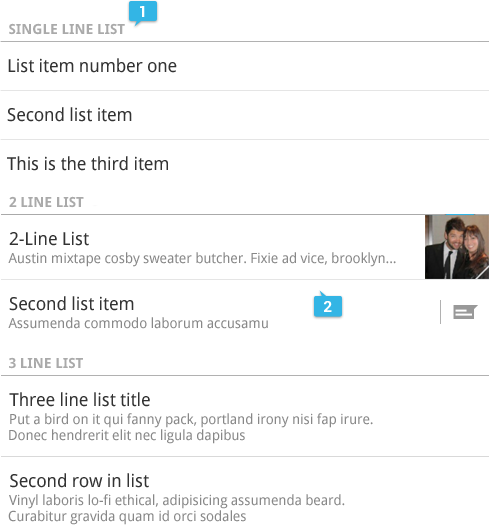
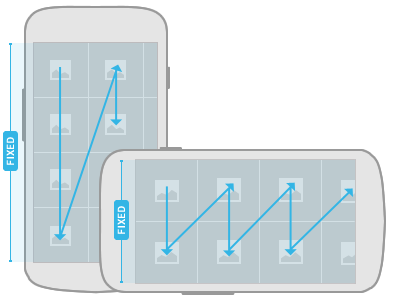
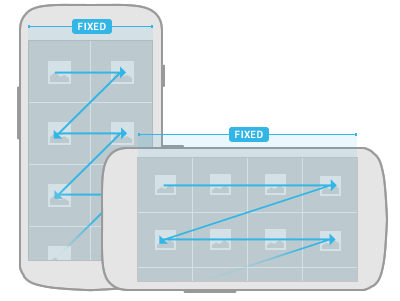
Korištenje listi za prikaz podataka

Prilikom razvoja android aplikacije često je potrebno organizirati podatke kao liste podataka koje korisnik može pretraživati. Liste (eng. Lists) predstavljaju skup podataka u kojemu su podaci organizirani vertikalno. Elementi liste mogu biti različiti tipovi podataka. Element liste može biti jedna ili više linija teksta, tekst u kombinaciji sa slikom ili nekim drugim *widgetom* ( *checkbox* ili *button*) ili nešto treće. Elementi liste mogu se pregledavati vertikalnim *scrollanjem.*



Različite implementacije listi

Alternativa listama su posebne liste koje se zovu „Grid liste“ (eng. Grid lists). Najbolje su za prikazivanje slika. Za razliku od običnih lista, one se mogu *scrollati* vertikalno i horizontalno. Elementi liste sortirani su s lijeva na desno i od gore prema dolje.



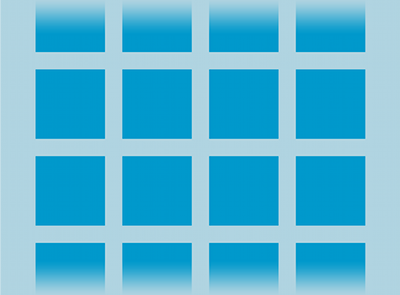
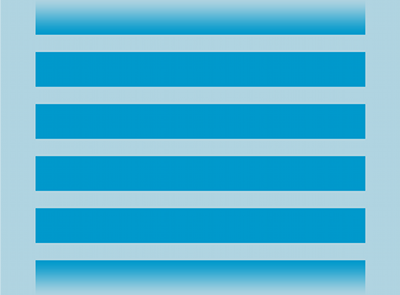
Organiziranje elemenata liste kod vertikalnog i horizontalnog *scrollanja*

Korisničko sučelje gradi se pomoću objekata iz klasa *View* i *ViewGroup*. Elementi na ekranu na koje korisnik može djelovati su objekti iz klase *View*. Objekt iz klase *ViewGroup* sadrži objekte iz *Viewa* i tako određuje raspored elemenata korisničkog sučelja.



Odnost *ViewGroup-View*

*ListView* je *ViewGroup* koji prikazuje listu podataka. Elementi liste ubacuju se u listu automatski korištenjem adaptera. *GridView* je *ViewGroup* koji elemente raspoređuje u dvodimenzionalnu mrežu. Prilikom implementiranja ove aplikacije, *ListView* korišten je za *Wall*, a *GridView* korišten je za galeriju.



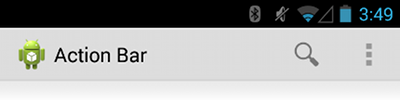
*ListView* i *GridView*

Adapter dohvaća podatke s nekog izvora, npr. polje ili baza podataka, i postavlja taj podatak u listu. Odgovoran je za stvaranje potrebnog objekta za prikaz, dodavanje podataka koji se trebaju prikazati u za taj objekt te prijenos tog objekta u *ListView*. Adapter je zapravo objekt klase *Adapter*.

Za implementiranje prilagođenih elemenata liste potrebno je napraviti novi *View* u XML-u koji definira kako će element liste izgledati. Isto tako, potrebno je napraviti potklasu klase *Adapter* u kojoj se definira kako napraviti, popuniti podacima i vratiti *View* definiran u novom *layoutu*. U potklasi *Adaptera* treba nadjačati metodu getView() koja vraća *view* iz prilagođenog *layouta*. Novi layout za *view* elementa liste kreira se jednako kao i *view* za aktivnost ili fragment. Prilikom implementiranja galerije i kreiranja novog *layouta* za elemente liste nismo koristili *ImageView* nego smo implementirali svoju potklasu *SquareImage*. Tako smo riješili problem koji se pojavljivao prilikom različitih veličina slika. *Layout* za element *Walla* sastoji se od objekata *TextViewa* i *ImageViewa*. Tako smo dobili element liste koji predstavlja post koji se sastoji od naslova, teksta posta, slike itd.

Anroid app navigacija

Vrlo važna stavka svake Android aplikacije je navigacija korisnika u aplikaciji. U ovoj aplikaciji koristi se traka akcija (eng. Action Bar) i navigacijske ladice (eng. Navigation Drawer). Traka akcija korisniku pruža informaciju o njegovom položaju unutar aplikacije. To je jednostavno sučelje koje se mijenja ovisno o aktivnosti ili fragmentu koji je trenutno u prvom planu. Traka akcijaosigurava prostor za naslov aplikacije i oznaku položaja korisnika unutar aplikacije. Uz to, često su prisutne ikone za određenju akciju (npr. pretraga, eng. Search) i *drop-down* liste. Ova aplikacija sadrži traku akcija u kojoj se prikazuje trenutni položaj korisnika (News, Messages, Wall, My Profile, Gallery), a prilikom otvaranja navigacijskih ladica, u traci akcija prisutan je naslov aplikacije (Ferbook) i gumb *logout* kojim se korisnici odjavljuju iz aplikacije.



Primjer Action Bar-a

Navigacijske ladice prikazuju glavne navigacijske opcije aplikacije. Pojavljuju se kada korisnik dotakne ikonicu aplikacije u traci akcija ili kada povuče prstom od lijevog prema desnom rubu. Kako bi se dodale navigacijske ladice u aplikaciju, potrebno je definirati korisničko sučelje tako da *DrawerLayout* bude korijenski element *layouta*. Zatim se ugnježđuje jedan *view* koji će sadržavati podatke dok je navigacijska ladica skrivena i *view* koji sadrži listu navigacijskih ladica. Prilikom dodira na određenu navigacijsku ladicu, fragment koji je određen za tu akciju prikazuje se na ekranu korisnika. U ovoj aplikaciji postoji pet navigacijskih ladica za koje su definirani fragmenti (News, Messager, Wall, My Profile, Gallery).



Primjer Navigacijskih ladica

Literatura:

1. http://developer.android.com/design/building-blocks/lists.html
2. http://developer.android.com/guide/topics/ui/layout/listview.html
3. http://developer.android.com/design/building-blocks/grid-lists.html
4. http://developer.android.com/guide/topics/ui/layout/gridview.html
5. http://developer.android.com/guide/topics/ui/actionbar.html
6. http://developer.android.com/training/basics/actionbar/adding-buttons.html
7. http://developer.android.com/training/implementing-navigation/nav-drawer.html
8. http://www.androiduipatterns.com/2013/05/the-new-navigation-drawer-pattern.html