

مهندسی نرم افزار

استاد درس: آقای دکتر عبدالله زاده

تهیه کننده: ملیحه هاشمی

USER EXPERIENCE
IN MOBILE

Why User experience?

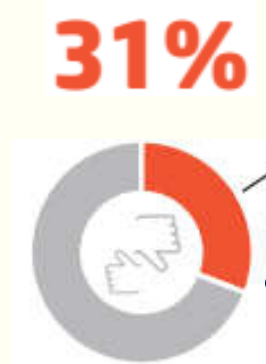
کاربران به یک تجربه موفق و بدون نقص در حین استفاده از موبایل نیاز دارند. در صورتی که این اتفاق نیافتد دیگر از این برنامه‌های کاربردی استفاده نمی‌کنند و یک رتبه ضعیف به آن می‌دهند.



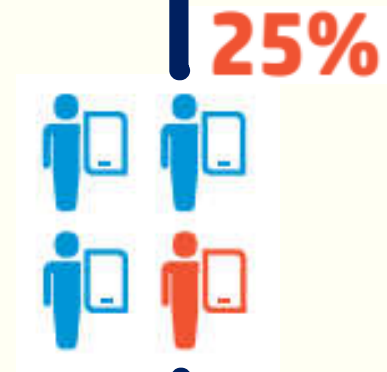
۷۹٪ کاربران در صورتی که یک برنامه کاربردی موبایل در انجام عملیات موردنظر شکست بخورد، تنها یک بار و یا دو بار جهت استفاده مجدد تلاش میکنند.



۳۱٪ افراد تجربه ناموفقی که در رابطه با یک برنامه کاربردی موبایل دارند به دیگران منتقل می‌نمایند.



۲۵٪ کاربران تنها پس از ۳ ثانیه تاخیر در پاسخ گویی، از به کارگیری یک برنامه کاربردی موبایل دست می‌کشند.



Providing User Experience

◆ جهت فراهم سازی تجربه کاربری مناسب باید به نکات گوناگونی در طراحی ابزار موبایل و برنامه کاربردی توجه نمود که در ادامه به برخی از موارد اشاره خواهد شد. این موارد عبارتند از:

◆ تایید و تصدیق (Confirmation & acknowledgement)

◆ کنترل مناسب حالات تهی (empty states)

◆ فراهم سازی روش‌های مناسب جهت پیمایش (navigation)

◆ توجه به Context

تایید و تصدیق (Confirmation & acknowledgement)

❖ زمانی که کاربر عملیاتی در برنامه کاربردی انجام می‌دهد، باید با استفاده از پیغام‌های متنی جهت تایید و تصدیق با کاربر در تعامل بود. این کار موجب کاهش عدم اطمینان درباره‌ی مواردی است که اتفاق افتاده یا خواهند افتاد. همچنین اشتباهات کاربر را نیز کاهش می‌دهد.

❖ تایید (conformation): از کاربر سوال می‌شود آیا مطمئن است که می‌خواهد چنین عملیاتی صورت پذیرد؟

◀ هشدارهای لازم و اطلاعات ضروری در رابطه با انجام عملیات در اختیار کاربر قرار می‌گیرد.

◀ دریافت تایید از کاربر خصوصا در مورد عملیات‌هایی که بازگشت پذیر نیستند و پیامدهای مهم دارند، دارای اهمیت است. در صورتی که عملیات انجام شده یک عملیات روتین است و به راحتی بتوان از این عملیات به وضعیت قبل برگشت، دریافت تایید الزامی نیست.

❖ تصدیق (acknowledgement): باید به صورت متنی کاربر را مطلع کرد که عملیات موردنظر او انجام شده است. در این رابطه ممکن است در برخی از موارد برای کاربر امکان undo عملیات نیز فراهم گردد.

حالات تهی (empty states)

❖ زمانی رخ می‌دهد که امکان نمایش محتوای یک آیتم وجود ندارد. برای مثال زمانی که یک لیست خالی است و یا جست و جو نتیجه‌ای در بر ندارد؛ این حالت بروز می‌کند. اگر چه این حالت‌ها ممکن است به صورت معمول رخ ندهند؛ اما باید از قبل یک طراحی مناسب برای آنها در نظر گرفت. این وضعیت را می‌توان به روش‌های زیر کنترل نمود:

❖ نمایش حالت تهی

❖ خودداری از بروز حالات تهی

◀ ارائه‌ی محتوایی مناسب به کاربر جهت شروع به کارگیری برنامه‌کاربردی

◀ ارائه‌ی محتوایی مناسب جهت آموزش کاربر

◀ ارائه **best match**

✓ هر یک از این روش‌ها در ادامه توضیح داده خواهند شد.

حالات تهی (empty states)



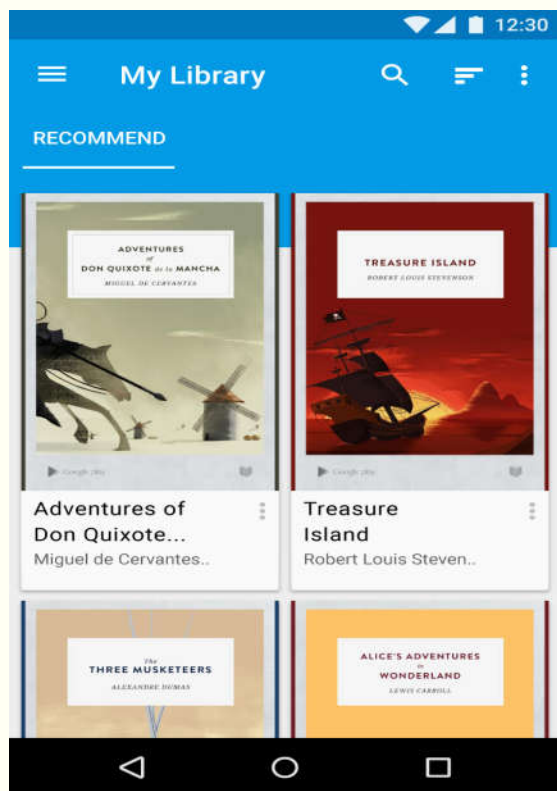
نمایش حالت تهی:

برای مثال شکل زیر نحوه‌ی مناسب نمایش حالت تهی را در رابطه با برنامه کاربردی ایمیل، زمانی که لیست ایمیل‌ها خالی است ارائه می‌دهد.

در این شکل از یک تصویر گویا، متناسب با زمینه و هدف برنامه کاربردی استفاده شده است. متن به کار رفته نیز به صورت کاملاً شفاف وضعیت را مشخص میکند.

این متن و شکل باید به نحوی ارائه شود که کاربر را به درستی از وضعیت با خبر سازد و از انجام عملیات و اقدامات غیرضروری که موجب گمراهی کاربر می‌شود؛ جلوگیری کند.

حالات تهی (empty states)



خودداری از بروز حالت تهی:

◇ ارائه‌ی محتوایی مناسب به کاربر جهت شروع به کارگیری برنامه کاربردی:

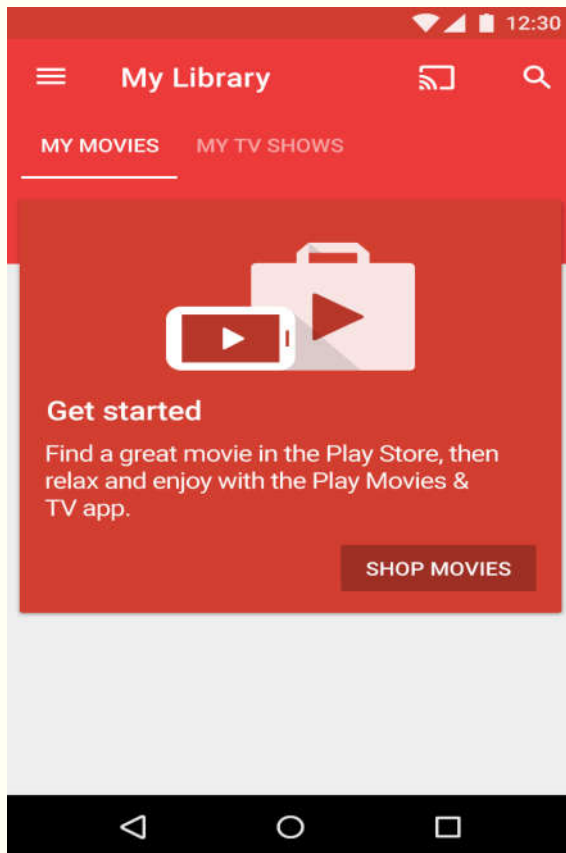
◀ موثرترین راه برای اینکه کاربران به یک برنامه کاربردی علاقمند شوند و آن را یاد بگیرند، هدایت مناسب جهت به کارگیری این برنامه کاربردی است.

◇ زمانی که کاربر شروع به کار با یک برنامه کاربردی می‌کند و داده‌ای در این برنامه ندارد (حالت تهی) توصیه زیر قابل استفاده هستند:

◀ از محتوایی استفاده شود که گیرایی گسترده‌ای دارد و ویژگی‌های اصلی برنامه‌های کاربردی را ارائه می‌دهد.

✓ برای مثال در شکل مقابل اگر چه کاربر تاکنون کتابی در برنامه کاربردی کتابخانه ذخیره نکرده است، اما برنامه در شروع گزینه‌هایی را به او را پیشنهاد می‌دهد تا آشنایی بیشتری با ویژگی‌های برنامه کاربردی برای کاربر فراهم شود.

حالات تهی (empty states)



❖ خودداری از بروز حالت تهی:

❖ ارائه‌ی محتوایی مناسب جهت آموزش کاربر:

◀ در صورتی که نمایش حالت تهی از طریق تصویر و متن (گزینه اول) کافی نباشد و موجب سردرگمی کاربر شود؛ می‌توان برای دستیابی به محتوای مناسب به او آموزش داد.

◀ این آموزش باید خلاصه باشد. کاربر به راحتی بتواند آن را skip کند یا در صورت تمایل دنبال کند.

✓ برای مثال شکل روبرو نحوه دستیابی به ویدئوها را به طور خلاصه به کاربر آموزش می‌دهد.

حالات تهی (empty states)

❖ خودداری از بروز حالت تهی:

❖ ارائه **best match**:

◀ زمانی که متن مورد جستجوی کاربر با هیچ یک از پرس و جویهای صورت گرفته در برنامه کاربردی دقیقاً مطابقت ندارد یا نتیجه‌ای در بر ندارد؛ نزدیکترین گزینه به او پیشنهاد گردد.

◀ این کار خصوصاً باعث می‌شود در مواردی که اشتباه نوشتاری در متن جستجو دارد با مشاهده **best match**، کاربر متوجه اشتباه خود شده و سریعتر به نتایج مورد نظر دست یابد.

Navigation

❖ با ارائه مسیرهای پیمایشی مناسب، کاربر به درستی در بخش‌های مختلف برنامه‌ی کاربردی هدایت می‌شود. در این راستا ساختار برنامه کاربردی باید بر مبنای محتوا و task های تعریف شده برای کاربر به نحوی سازماندهی شود که مسیرهای پیمایشی شهودی و قابل پیش بینی باشند.

❖ برای ارائه یک طراحی پیمایشی مناسب توجه به نکات زیر دارای اهمیت می‌باشد:

I. Take Inventory

II. Prioritize


III. Sequence

IV. Deconstruct

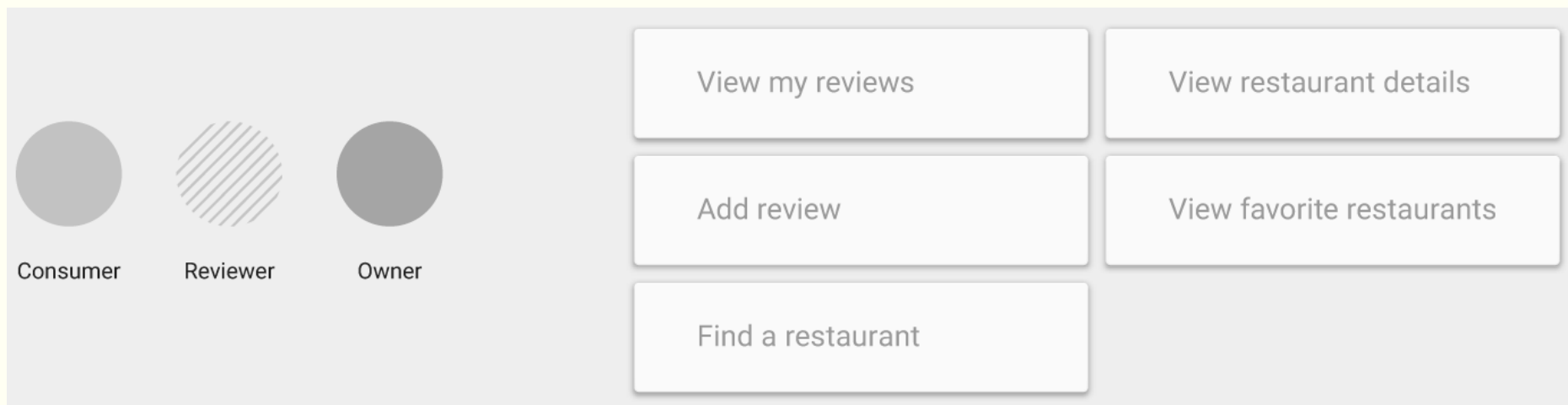
✓ این موارد در اسلایدهای بعدی با ذکر مثال در رابطه با برنامه کاربردی راهنمای رستوران‌ها توضیح داده شده‌اند.

Navigation

:Take inventory

شناسایی کاربرهای کلیدی و کارکردها و **task** های مورد نظر آنها 

◀ شکل زیر این مورد را در رابطه با برنامه کاربردی راهنمای رستوران ها نمایش می دهد.

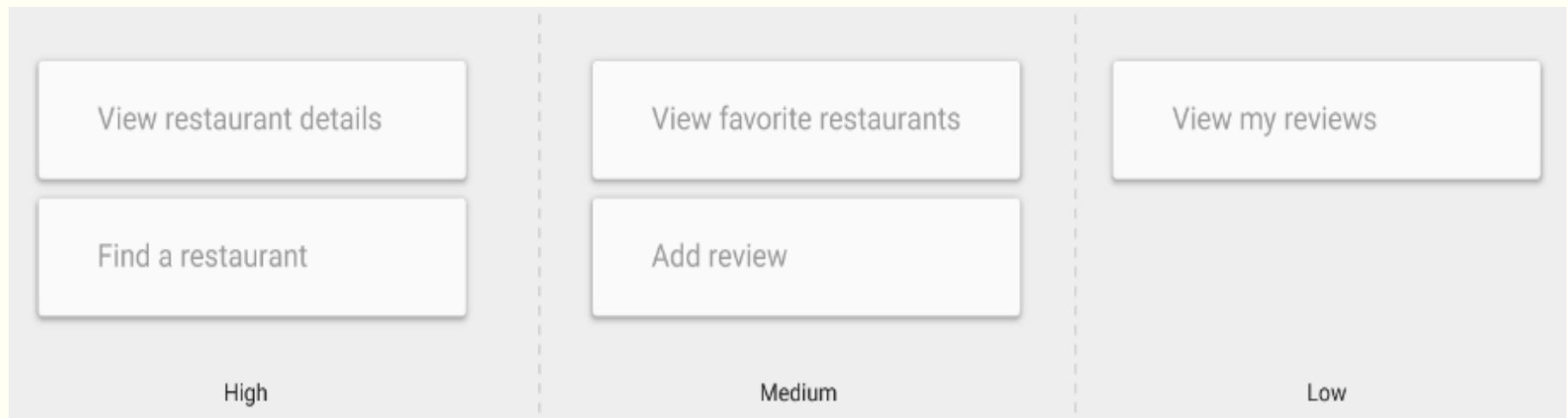


Navigation

:Prioritize

◇ اولویت بندی کارکردها و **task** های شناسایی شده

◇ باید توجه شود که ممکن است این اولویت بندی ها در طول زمان با تغییر نیازمندی های کاربر تغییر یابند.



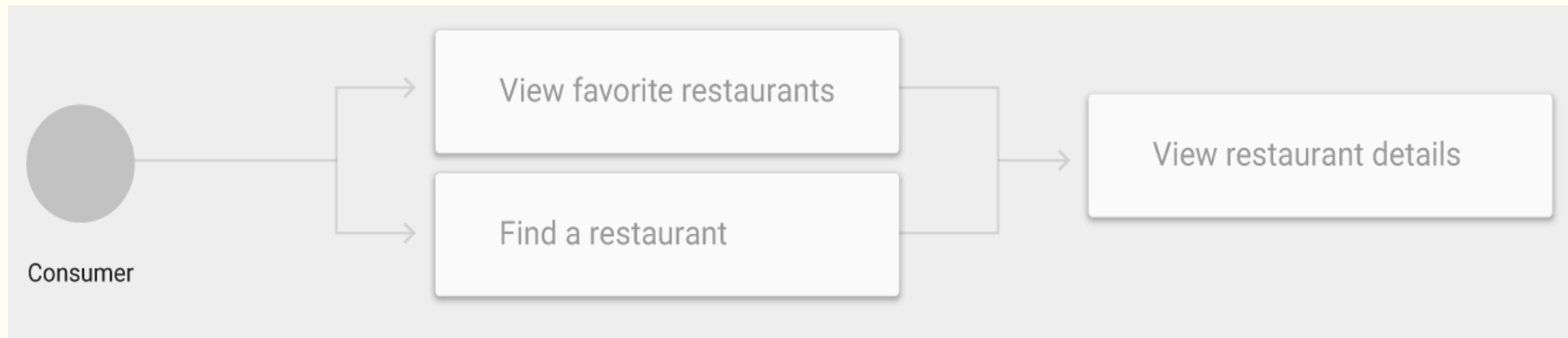
Navigation

:Sequence

◇ باید مسیرهای مختلف که کاربران در طول استفاده از برنامه کاربردی طی می کنند شناسایی شود تا بر مبنای آن ها طراحی پیمایشی صورت گیرد.

◀ مقصدهای نهایی که مسیرهای گوناگونی به صورت مکرر به آن ها ختم می شوند باید برجسته شوند.

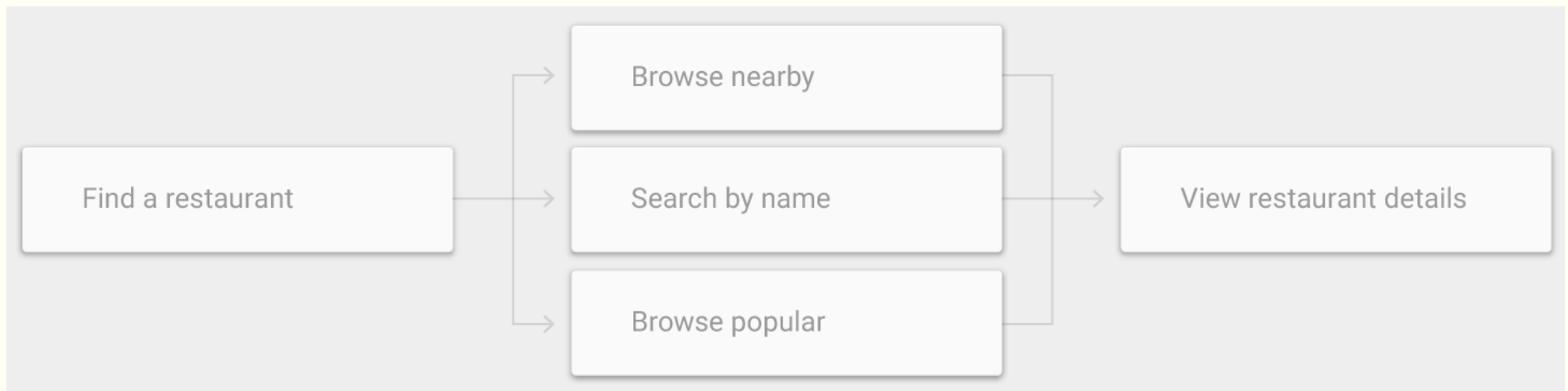
◀ task های مرتبط با هم باید گروه شوند و از آن ها برای ساختاردهی مسیرهای پیمایشی در برنامه کاربردی استفاده شود.



Navigation

:Deconstruct

◆ فعالیت‌های پیچیده، دارای ابهام باید به task های کوچک تر شکسته شود. این وظایف کوچک تر ساده تر درک می‌شوند و از این طریق می‌توان به نحو بهتری اهداف کاربر را برطرف نمود.




Navigation

روش‌های مختلف جهت پیمایش عبارتند از: 


Embedded navigation 

Tabs 

Bottom navigation bar 

Navigation drawer 

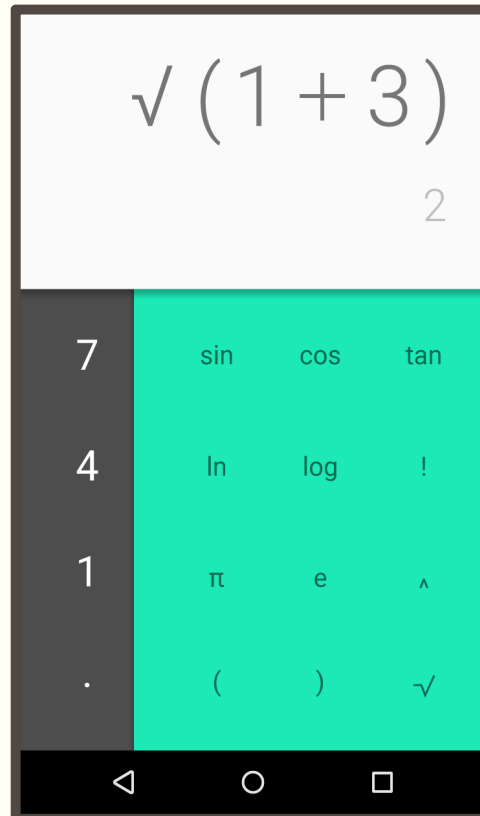
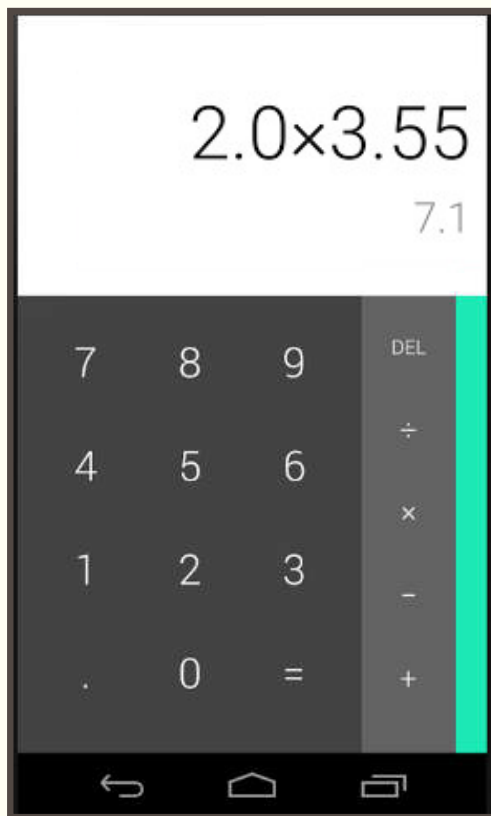
Expanding navigation drawer 

Gestural 

Navigation- Embedded Navigation

- ◆ برنامه‌های کاربردی که از پیمایش ساده‌ای برخوردار هستند، ممکن است بخش‌های مربوط **navigation** را در داخل خود برنامه کاربردی جا داده باشند. در نتیجه‌ی این کار قسمتی از صفحه برنامه کاربردی به این موارد اختصاص پیدا می‌کند و میزان فضای در دسترس برای نمایش محتوای موردنظر کاهش می‌یابد.
- ◇ این نوع روش پیمایشی برای برنامه‌های کاربردی که دارای خصوصیات زیر هستند پیشنهاد میشود:
 - ◁ صفحه اولیه آنها از محتوا و کارکردهای کافی برای ارائه به کاربر برخوردار است و مراجعه به صفحات دیگر آن کم تر رخ می‌دهد.
 - ◁ برنامه‌های کاربردی که کارکردهای مرسوم و مشخصی را در صفحه اولیه خود ارائه می‌دهند.

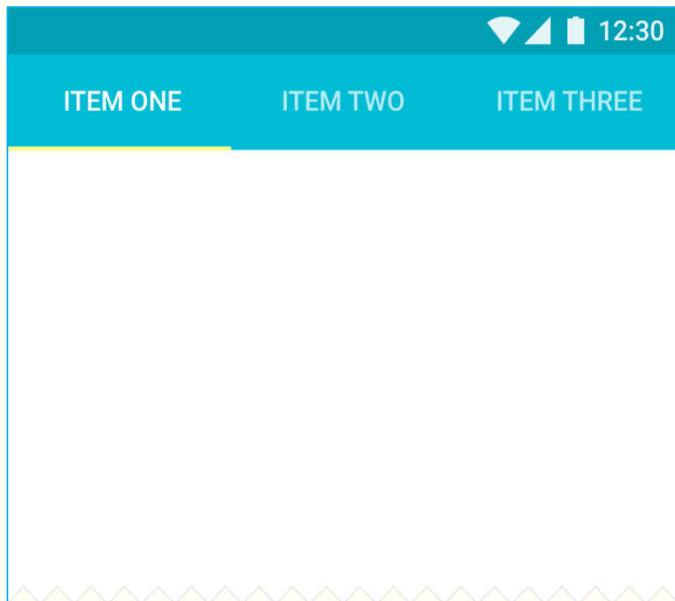
Navigation- Embedded Navigation



◆ مثال: برنامه کاربردی ماشین حساب در موبایل

◆ عملگرهای محاسباتی در همان صفحه اصلی جا داده شده‌اند و کاربر برای دسترسی به آنها نیاز به پیمایش بخصوصی ندارد.


Navigation

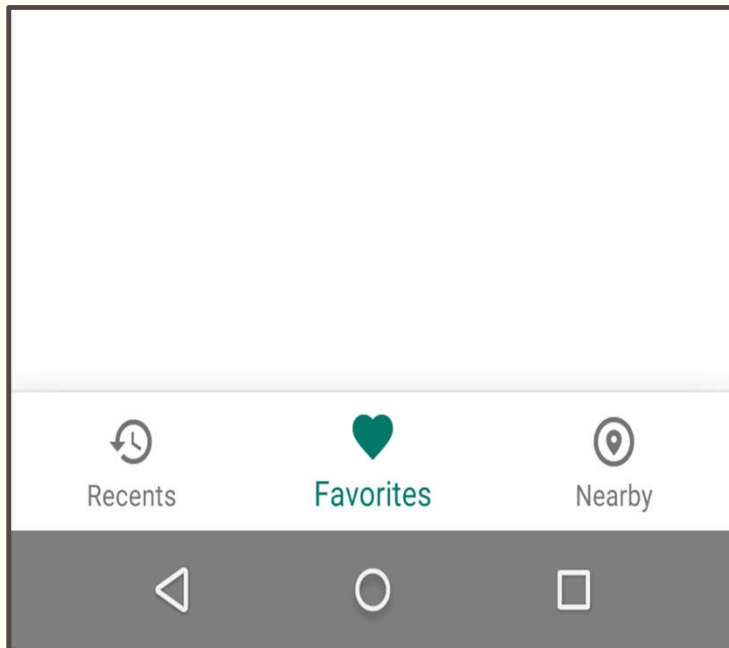


◆ **tab** به کاربران اجازه می‌دهد تا خیلی سریع بین تعداد محدودی از **view** ها مرتبط که دارای اهمیت برابری هستند جا به جا شوند. این روش در زمانی توصیه می‌شود که:

- ◆ بین **view** های مختلف به صورت مکرر سوئیچ می‌شود.
- ◆ آگاه نمودن کاربر از وجود **view** های دیگر
- ◆ برنامه‌های کاربردی که دارای **view** های اصلی محدود و کم می‌باشند.

Navigation- Bottom Navigation bar

مانند  tab عمل می‌کند با این تفاوت که بیشتر در ارائه‌ی محتوا در رابطه با **کارکردهای** مرتبط با هم مورد استفاده می‌شود. مانند شکل زیر:



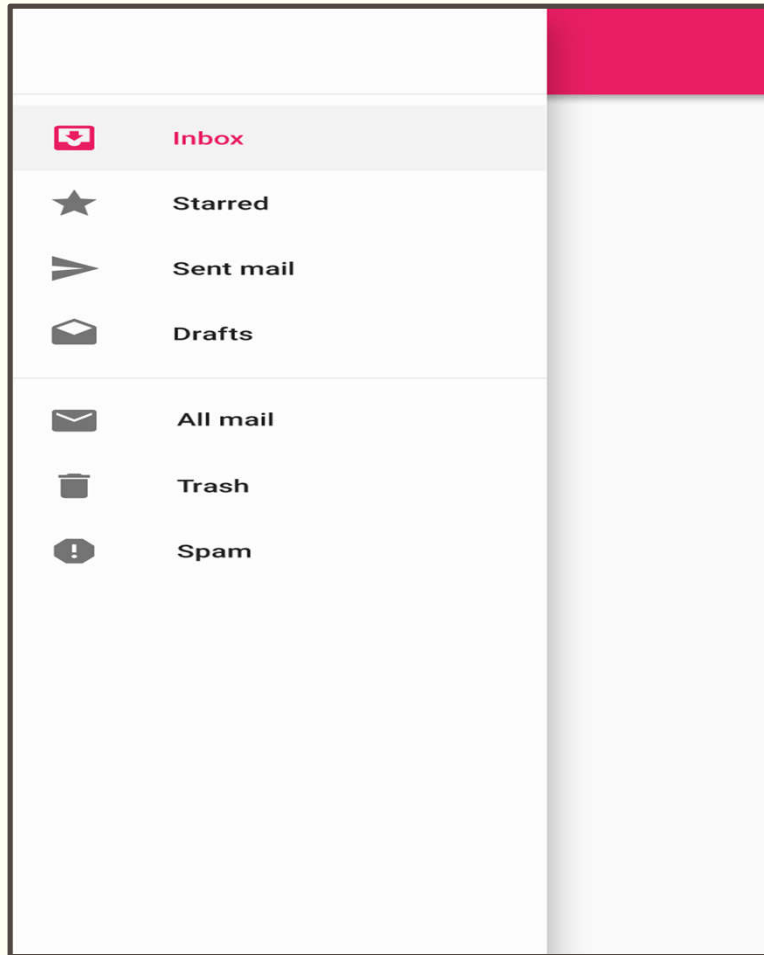
Navigation- Navigation drawer

زمانی که فضای کافی برای استفاده از روش پیمایش مبتنی بر tab وجود ندارد، استفاده از side navigation یک جایگزین مناسب به شمار می‌رود. Side navigation می‌تواند تعدادی از اهداف پیمایشی (navigation targets) را به صورت همزمان به نمایش در آورد. این side navigation به طور ثابت نیست و در نتیجه در صورت عدم لزوم فضای صفحه را اشغال نمی‌کند؛ بلکه به کشویی باز و بسته می‌شود.

استفاده از این روش در موارد زیر توصیه می‌شود:

- برنامه‌ی کاربردی تعداد زیادی view مهم و اصلی دارد.
- نیاز به پیمایش سریع بین view های نامرتبط به هم وجود دارد.
- ساختار پیمایشی طراحی شده دارای عمق زیادی می‌باشد.
- مسیرهای پیمایشی غیرمکرر در قسمت‌های پایین این لیست کشویی قرار می‌گیرند و کم‌تر توجه کاربر را جلب می‌کنند.

Navigation- Navigation drawer



مثال  :

Navigation- Expanding navigation drawer

World	▼	World	^
U.S.	▼	Africa	
Politics	▼	Americas	
Business	▼	Asia Pacific	
Tech	▼	Europe	
Sports	▼	Middle East	
Arts	▼	U.S.	▼
		Politics	▼
		Business	▼
		Tech	▼
		Sports	▼
		Arts	▼

❖ زمانی که سلسله مراتبی پیمایشی دارای عمق زیادی می‌باشد، می‌توان سلسله مراتب آیتم‌های یک **navigation drawer** را توسعه داد. به عبارتی هر آیتم در سطوح زیرین خود تعدادی موارد را گنجانده است. همچنین این سلسله مراتب می‌تواند در سطوح زیرین نیز وجود داشته باشد. این سلسله مراتب در صورتی که کاربر مایل باشد امکان مخفی شدن و در صورت نیاز او امکان نمایش دارند.

Navigation- Gestural

◈ پیمایش با استفاده از حرکات دست به کاربر این امکان را می‌دهد که به راحتی بین view های متوالی و یا سلسله مراتبی به راحتی جا به جا شود. حرکات پشتیبانی شده در این زمینه عبارتند از :

◈ لمس صفحه و کشیدن آن به صورت افقی به سمت چپ و راست (مثلا در برنامه کاربردی ایمیل امکان جابه جایی بین ایمیل های متوالی در یک پوشه فراهم می شود).

◈ لمس صفحه و کشیدن آن به صورت عمودی به سمت بالا و پایین

◈ زمانی توصیه می شود که:

◈ رابطه ترتیبی بین view های مختلف وجود دارد (برای مثال صفحات متوالی)

◈ View از لحاظ نوع محتوا یکسان هستند . (مثلا همه ی ایمیل های موجود در یک پوشه)

Context در محیط موبایل

تعریف context

هر اطلاعاتی که مشخص کننده‌ی موقعیت یک موجودیت است و در ارتباط بین کاربر و یک برنامه کاربردی تاثیرگذار و مرتبط است و موجب تغییر رفتار آن برنامه کاربردی می‌گردد.

هدف اصلی از برنامه‌های کاربردی تحت موبایل آگاه از زمینه (context) این است که ابزارهای موبایل ویژگیهای محیط خود را تشخیص و تفسیر نمایند تا پاسخ هوشمندانه‌تر ارائه نمایند.

استفاده از برنامه‌های کاربردی تطبیق‌پذیر و آگاه از زمینه یک راه مناسب برای پاسخ‌گویی به محدودیت‌های ابزار است. مثلا سایز صفحه و حافظه.

مثال:

برنامه کاربردی راهنمای تور آگاه از مکان که براساس موقعیت و محل کاربر راهنمایی مناسب به او ارائه می‌دهد.

Context در محیط موبایل

context در محیط موبایل را می توان به ۵ دسته تقسیم کرد: ◀

User context ◊

◀ تنظیمات کاربر (user preferences) ، نیازهای اطلاعاتی (information needs) ، برنامه ی زمانی شخصی (personal time schedule)

Device context ◊

◀ سیستم عامل ، ساینز صفحه ، پروتکل های شبکه قابل تعبیه شده در دستگاه (embedded network protocol)

Activity context ◊

◀ نوع Task ها و زمان انجام یک task

Network context ◊

◀ تنظیمات شبکه و اتصالات

External physical environment context ◊

◀ میزان نویز در محیط ، روشنایی ، دما