PANDA

Estrategia de pruebas

1.- Requisitos y especificaciones de la aplicación:

Funcionalidad

La aplicación tiene varias funcionalidades que voy a nombrar, explicar y mostrar un diagrama de flujo para entender más la aplicación.

SPLASHSCREEN

Antes de que cargue la aplicación, hay una pantalla de carga con el logo de la aplicación, que se encuentra esperando a que todo lo necesario de la aplicación cargue correctamente.

Para que esta funcionalidad se valide correctamente, tendremos que comprobar si aparece siempre que se inicia la aplicación y que cuando la siguiente pantalla carga, todo está bien cargado.

LOGIN

Todo empieza con el login, aquí habrá que testear que todos los componentes de ven por pantalla y si cuando el usuario hace click en el boton de login se puede acceder a la siguiente pantalla.

Navigation Bar

El navigation bar es el que conecta las pantallas pricipales cómo son, home(donde puedes acceder a las piezas de ropa), configuración(puedes cambiar de idioma y cerrar sesión) y la carta que aparece lo que has comprado.

De esto tenemos que testear que los tres botones del navbar cargan las pantallas correctas y que no desaparece en las pantallas principales.

Pantalla Home

En esta pantalla aparecen los tres apartados principales con los que poder filtrar la ropa.

Camisetas, pantalones y zapatos.

En esta pantalla lo que habrá que testear, es si aparecen esos tres elementos que son los que nos van a permitir hacer las compras y que, cuando hagamos click en el elemnto deseado, se cargue la pantalla.

Pantallas Donde aparece el listado de la ropa

En esta pantalla, aparecen unas tarjetitas con la imagen, nombre, precio y botón de añadir de la pieza de ropa correspondiente.

Aquí habrá que testear, que todas las piezas de ropa aparezcan, que cuando le des al botón de añadir, se añade a la base de datos interna,que luego se utiliza para que aparezca en la carta.

También habría que intentar añadir la misma pieza en la carta y mirar si no se duplica en la vista.

Configuración

En esta pantalla sale un spinner que te permite cambiar el idioma y un botón, que te permite cerrar sesión.

Aquí habrá que testear, que realmente se cambie el idioma de toda la aplicación y que cuando cierras sesión, se cierre de verdad y cuando vuelvas a hacer login te pida otra vez la cuenta de correo electrónico.

Carta

En esta pantalla aparece la ropa que has añadido a la carta, la carta te permite adquirir más de una unidad de producto y eliminar la pieza.

Habrá que testear que la base de datos devuelva las piezas de ropa correctas y que te permite añadir más de una unidad y eliminar del carrito una pieza.

Y que todo esto se ve reflejado en la base de datos.

También hay un botón que te permite pasar a la pantalla de finalizar compra.

Aquí habrá que testear que este botón hace su funcionalidad, que es pasar a la siguiente pantalla.

Pantalla de finalizar compra

En esta pantalla, aparece un mensaje informando que se está finalizando la compra.

También aparece un botón con un mapa y un botón para finalizar la compra.

Si le damos al botón del mapa se despliega un mapa donde se muestran las ubicaciones de recogida de la ropa y más información.

En esta pantalla hay que testear el correcto funcionamiento de los dos botones y si el mapa se despliega correctamente.

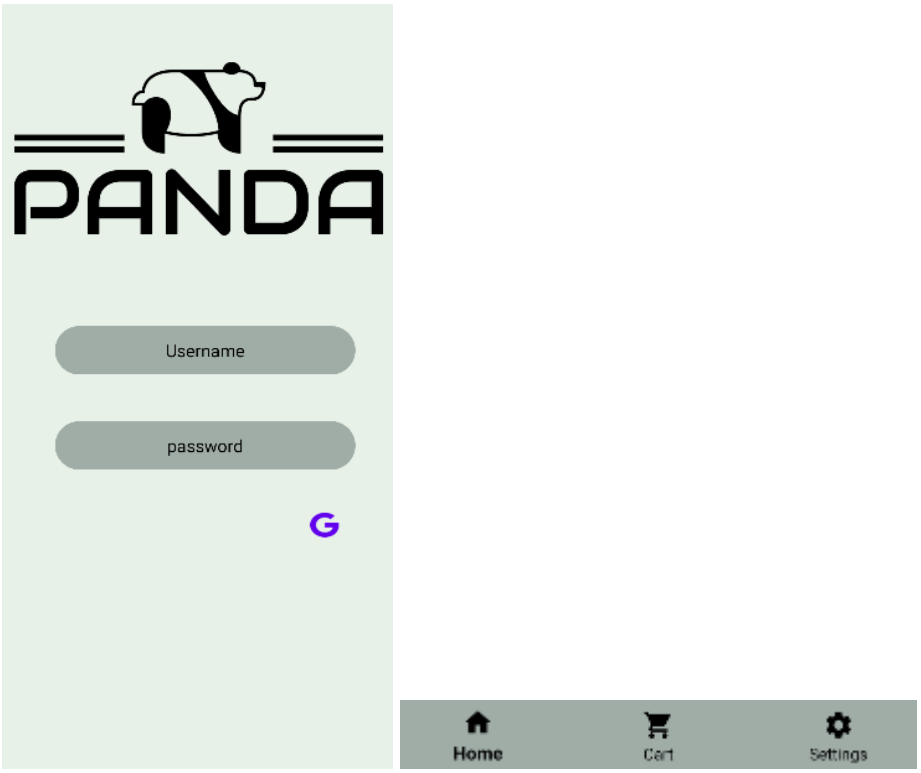
Dentro del mapa, hay que testear que los marcadores(son los puntos de recogidas), están correctamente ubicados.

En el botón de finalizar compra hay que comprobar que borre la tabla dentro de la base de datos, para que se pueda volver a hacer otra compra.

Usabilidad

La aplicación debe de ser fácil de utilizar, para ello, Panda, consta de una interfaz minimalista.

Todas las pantallas son intuitivas de acceder.



En estos ejemplos de navegación, podemos ver como se puede acceder a los diferentes apartados de la app con un solo click, ya sea en imágenes representativas o logos con texto.

Compatibilidad

Nuestra aplicación está hecha para diferentes tamaños de móvil. De momento sólo está enfocada para Android.

Desde el navbar, hasta las tarjetitas dónde se muestra la información són responsive, lo que nos deja seguro que no se deformará dependiendo del móvil.

El mapa de google maps también es responsive, ya que se adapta al ancho y alto de la pantalla.

En todos los demás apartados de la aplicación, no se tiene claro, y por lo tanto habrá que testear que no se deforme en diferentes dispositivos móviles o tablets.

Seguridad

La aplicación consta de un login de google, para proteger la seguridad del usuario, a más a más consta de un login en local, el cual pide usuario y contraseña .

Todas estas medidas están hecha para proteger la app, tanto de ataques de bots, cómo de intentos de robo de cuenta.

Rendimiento

La app, consta de tres procesos que pueden ser más demandantes.

El primero, són las peticiones al firebase, estas pueden tardar ya que tienen que cargar tanto imágenes cómo la información del producto.

El segundo proceso o grupo de procesos, son los que interactúan con la base de datos.

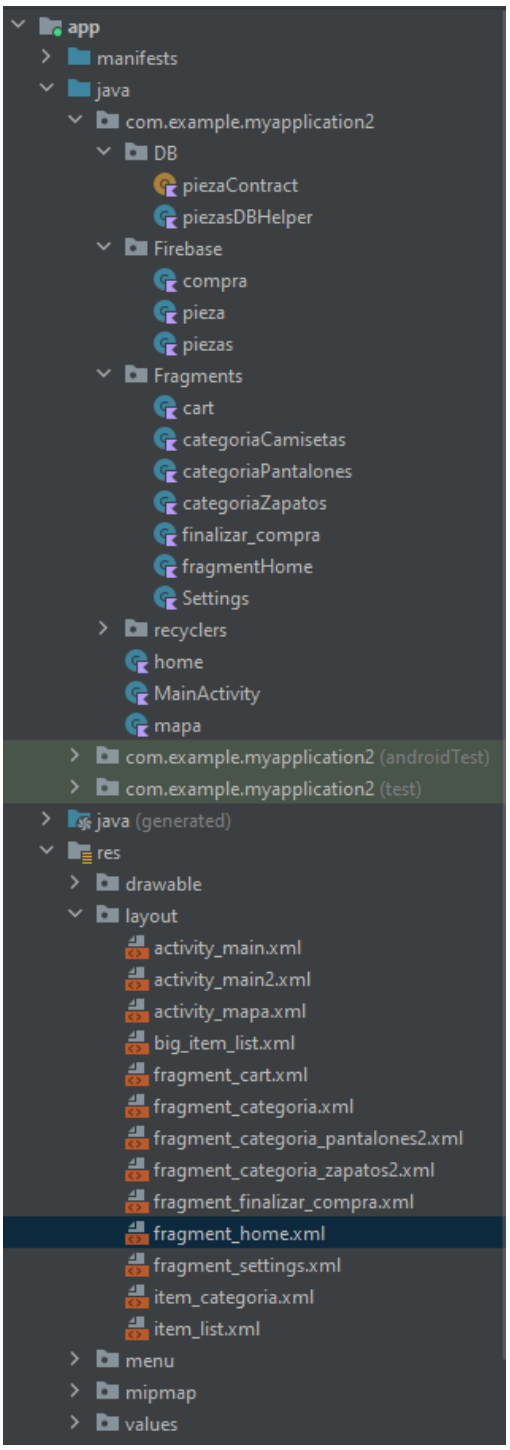
Estos son los de añadir al carrito, eliminar del carrito, y finalizar compra.

Como se ejecutan en una base de datos interna, no tardan nada y son super amenos para el usuario.

El último es el mapa de google maps, al interactuar con la api de google a veces cuando se despliega puede cargar mal o por trozos, habrá que comprobar que todo se muestra correctamente.

Mantenibilidad y Escalabilidad

La aplicación consta de una estructura de ficheros clara y limpia y un código, que en algunas clases es limpio y escalable y en otras no tanto.



En el apartado código habría que empezar a mejorar y escalar el código.

Por ejemplo el crud en firebase habría que meterlo todo en una clase y acceder a esa clase cuando se desee utilizar algo relacionado con firebase.

Ahora mismo se encuentran todos dispersados por muchas classes distintas, cosa que le quita claridad y legibilidad al código.

En el otro lado de la moneda, se encuentra el crud en la base de datos, todo está dentro de una función, lo que simplifica la usabilidad y legibilidad de código.

En igualdad de condiciones se encuentra la clase de googleMaps.

Todas las demás clases son claras de leer y hacen su función.

Integración

En cuanto a integración esta app únicamente depende de que sea desplegada en android.

Y de recibirla información del firebase,lo que tengo que estar pendiente de que no se caduque.

También dependemos de la api de google maps.

Esta app puede ser desplegada en todos los dispositivos Android.

2.- Plan de pruebas

Login

clases que testear: MainActivity.kt

1.- ¿Se muestran todos los componentes por pantalla?

Puntos de aceptación:

- La imagen se encuentra en pantalla al momento de cargar la págins:
- Los placeholders se encuentran en pantalla al momento de cargar la página:
- El botón se encuentra en pantalla al momento de cargar la página:

2- Los placeholders contienen lo que tienen que contener:

Puntos de aceptación:

- El primer placeholder contiene el texto ‘Username’
- El segundo placeholder contiene el texto ‘password’

3- El botón hace su función cuando hacemos login

Puntos de aceptación:

- Cuando introducimos en el campo de username cualquier nombre y la password lo mismo,no nos deja entrar:
- Cuando introducimos en el campo username ‘harun’ y en la password ‘123456’ nos deja entrar:
- Hacemos login con google con una cuenta de gmail cualquiera y nos deja entrar:

Página Home

1- ¿Se encuentran todos los componentes por pantalla?

Puntos de aceptación:

- Se encuentra la imágen cuando carga la pantalla:
- Se encuentran las tres imágenes, camiseta, pantalón y zapatos en este mismo orden:
- Se encuentra en navbar por pantalla:

2-¿Las imágenes de ropa cargan su pantalla correspondiente?

Puntos de aceptación:

- La imagen de una camiseta carga la pantalla fragmentCamisetas:
- La imagen de un pantalón carga la pantalla fragmentPantalones:
- La imagen de un par de zapatos carga la pantalla fragmentZapatos

Configuración

Clase:Settings

1-¿Se cambia el idioma correctamente?

Puntos de aceptación

- Le damos click en idioma, cambiar idioma al inglés y cambiamos de página, aparecen todos los textos en inglés:
- Cuando cambiamos de idioma al Español , cerramos la app y la volvemos a abrir. El idioma se mantiene cambiado:

2.- ¿Se cierra la sesión correctamente?

Puntos de aceptación

- Le damos click al botón de cerrar sesión. Nos envía a la página del login. Cuando volvemos a intentar inciar sesión nos vuelve a pedir el correo electrónico:

Páginas para comprar ropa y Carrito

Clase(s):categoriaCamisetas, categoriaZapatos, categoriaPantalones

Clases anexas:piezasDBHelper

1-.¿Aparecen piezas de ropa?

Puntos de aceptación

- Mirar en Firebase si hay piezas de ropa subidas, si las hay, tienen que estar todas plasmadas en el frontal de nuestra aplicación.

2.- ¿Se añaden piezas de ropa al carrito?

Puntos de aceptación

- Le damos al botón de añadir una pieza, primero de todo tiene que aparecer como introducido en la base de datos, después tiene que aparecer un mensaje por pantalla que hemos introducido eso correctamente y por último tiene que aparecer en el frontal del carrito:

- Intentamos comprar la misma pieza de ropa dos veces, le damos al botón de más por primera vez, hacemos las comprobaciones anteriores, le volvemos a dar al botón de más. No se ha vuelto a introducir en la base de datos, tampoco aparece por duplicado en el frontal:

3.-¿Se eliminan piezas de ropa del carrito?

Puntos de aceptación

- Hacemos scroll hacia la derecha del botón que deseamos eliminar. No aparece un mensaje diciendo si queremos o no. Ponemos ok. Se ha eliminado tanto del frontal del carrito como de la base de datos:

- Hacemos scroll hacia la derecha del botón que deseamos eliminar. No aparece un mensaje diciendo si queremos o no. Ponemos cancel.No se ha eliminado ni de la base de datos ni del frontal:

4.-¿Añadir o disminuir unidades de ropa en el carrito?

Criterios de aceptación

- Le damos al botón más y aumenta una unidad:
- No podemos poner más de 99 unidades:
- Le damos al botón de menos y disminuye una unidad:
- No puede ser menos de una unidad:

Google Maps

Clase: mapa

1.- ¿Se despliega correctamente?

Criterios de aceptación

- Desde la página de finalización de compra le damos al botón del mapa. El mapa se despliega con todo el alto y ancho de la pantalla y con una animación se muestra nuestra ubicación actual:

- Cuando le damos al botón del mapa por primera vez, pide permisos para detectar nuestra ubicación actual:

- Creamos unos marcadores, desde la función createMarker, aparece en nuestro mapa el marcador con color rojo.

- Dentro del mapa, le damos al botón que nos aparece arriba a la derecha y nos lleva a la ubicación actual:

Finalizar Compra

Clase:finalizar_compra

Clases anexas:piezasDBHelper

1.-¿Finalizamos compra correctamente?

- Cuando le damos al botón de abajo a la derecha dentro de la pantalla de finalizar compra, nos aparece un mensaje de que nuestra compra ha sido finalizada con éxito. Comprobamos y la base de datos ha borrado todo el contenido, así mismo el frontal también está vacío-

3.-Herramientas para testear

Las herramientas que utilizaremos en nuestro proyecto serán las siguientes:

- JUnit
- Espresso

JUnit:

¿Qué es?

Es un framework de pruebas unitarias para el lenguaje de programación Java (en nuestro caso Kotlin), que se utiliza para probar de manera automatizada la funcionalidad de las diferentes partes de nuestro código.

Beneficios y funcionalidades de JUnit en Android Studio:

- Pruebas unitarias:
  - JUnit permite escribir pruebas unitarias para verificar el comportamiento esperado de nuestro código, como métodos, clases.
- Automatización:
  - Se pueden crear conjuntos de pruebas automatizadas que se ejecutan automáticamente cada vez que realizas cambios en tu código. Esto te ayuda a detectar rápidamente errores.
- Integración:
  - Con Android Studio, JUnit está integrado de manera nativa en Android Studio.

Espresso

¿Qué es?

Espresso es una herramienta de prueba automatizada en Android Studio que se utiliza específicamente para realizar pruebas de interfaz de usuario (UI) en aplicaciones Android.

Funcionalidades y beneficios:

Pruebas de interfaz de usuario:

- Espresso está diseñado para facilitar la escritura y ejecución de pruebas automatizadas de UI. Permite simular acciones del

usuario, como hacer clic en botones, escribir texto, desplazarse por vistas y realizar gestos táctiles.

- Sintaxis clara y concisa: Espresso utiliza una sintaxis simple y legible, lo que facilita la escritura y comprensión de las pruebas.

- Integración con Android Studio: Espresso está estrechamente integrado en Android Studio, lo que permite ejecutar las pruebas directamente desde el IDE.

4.-VIDEO EXPLICACIÓN Y FUNCIONAMIENTO DE TESTING

link yt: https://www.youtube.com/watch?v=-mzbYccpBvc