



Mòdul 8
DAM 2

Autor

Unitat formativa

Pràctica

Programació multimedia i
dispositius mòbils

Helena Madrenys Planas

UF1 – Desenvolupament d'aplicacions per
dispositius mòbils

T10.PrFinal – 25/04/2022

Pràctica Final UF1





Índex

Preparació.....	4
nav_header_main.xml.....	4
activity_main_drawer.xml.....	4
mobile_navigation.xml.....	5
MainActivity.java.....	5
Activity About.....	6
activity_about.xml.....	6
AboutActivity.java.....	6
Activity Configuration.....	7
activity_configuration.xml.....	7
ConfigurationActivity.java.....	7
Fragment Home.....	8
fragment_home.xml.....	8
HomeFragment.java.....	8
Fragment Calendar.....	9
fragment_slideshow.xml.....	9
SlideshowFragment.java.....	9
Fragment List.....	10
fragment_gallery.xml.....	10
item_rview.xml.....	10
MovieClass.java.....	10
MovieAdapter.java.....	11



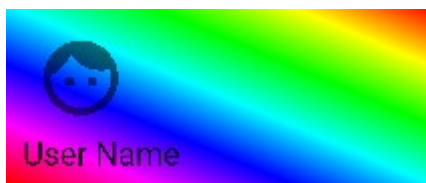
GalleryFragment.java.....	14
Altres.....	15
Idiomes.....	15
Colors.....	15
Imatges/Icones.....	15
Output.....	16
Link Video.....	16
Captures.....	16

Preparació

Abans de tot, al crear el projecte enlloc de crear un «Empty Activity», hem creat un «Navigation Drawer Activity». Aquesta ja ens porta un menú lateral que van a 3 fragments diferents. Abans de començar doncs, haurem de fer que aquest menú porti també a dues activitats més.

nav_header_main.xml

Aquí l'únic que hem fet ha sigut variar una mica el disseny. Hem canviat el fons de color per una imatge, hem canviat la icona de l'usuari i el text que hi havia a sota.

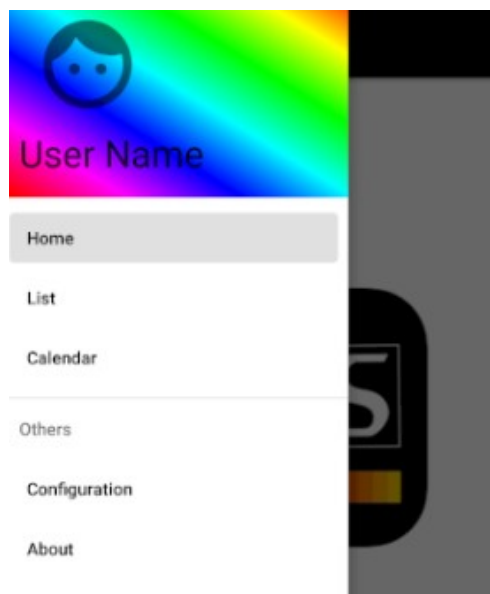


activity_main_drawer.xml

Hem canviat el nom dels elements que hi havien i hem afegit els que ens faltaven. Per a poder posar la separació dels subapartats hem fet servir grups i items amb altres items dins.

```
<group android:checkableBehavior="single">
    <item
        android:id="@+id/nav_home"
        android:title="Home" />
    <item
        android:id="@+id/nav_list"
        android:title="List" />
    <item
        android:id="@+id/nav_calendar"
        android:title="Calendar" />
</group>

<item android:title="Others">
    <menu>
        <group>
            <item android:id="@+id/nav_configuration"
                android:title="Configuration" />
            <item android:id="@+id/nav_about"
                android:title="About" />
        </group>
    </menu>
</item>
```



mobile_navigation.xml

Aquí hi hem hagut de afegir els dos items nous (les activitys) per a que quedin linkades al menú i que quan s'hi faci click les pugui trobar.

```
<fragment
    android:id="@+id/nav_calendar"
    android:name="com.example.m8uf1t10helenamadrenys.ui.slideshow.SlideshowFragment"
    android:label="Calendar"
    tools:layout="@layout/fragment_slideshow" />

<activity
    android:id="@+id/nav_configuration"
    android:name="com.example.m8uf1t10helenamadrenys.ConfigurationActivity"
    android:label="Configuration"
    tools:layout="@layout/activity_configuration" />

<activity
    android:id="@+id/nav_about"
    android:name="com.example.m8uf1t10helenamadrenys.AboutActivity"
    android:label="About"
    tools:layout="@layout/activity_about" />
</navigation>
```

MainActivity.java

Hem hagut de afegir les dues noves activitys en el binding per a que les pugui trobar al fer click.

```
mAppBarConfiguration = new AppBarConfiguration.Builder(
    R.id.nav_home, R.id.nav_list, R.id.nav_calendar, R.id.nav_configuration, R.id.nav_about)
    .setOpenableLayout(drawer)
    .build();
```

Activity About

activity_about.xml

Hi hem deixat el «TextView» per defecte i hi hem afegit un botó flotant. La imatge ha sigut una icona que hem creat amb «new – imageAsset». Per a que quedés centrada i gran hem hagut de editar les propietats que veiem a continuació:

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="80dp"
    android:layout_height="80dp"
    app:fabCustomSize="80dp"
    android:layout_gravity="end|bottom"
    android:layout_marginEnd="36dp"
    android:layout_marginBottom="36dp"
    android:backgroundTint="@color/white"
    android:contentDescription="DESCRIPTION"
    android:src="@drawable/ic_info"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:maxImageSize="90dp" />
```



AboutActivity.java

Simplement doncs hem de posar un mètode «OnClick» en el botó i fer que ens aparegui un «AlertDialog» amb icona, títol i text.

```
FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //Obrim un alertDialog
        AlertDialog.Builder builder = new AlertDialog.Builder(context: AboutActivity.this);
        builder.setTitle("Information");
        builder.setMessage("Helena Madrenys, 2022");
        builder.setIcon(R.drawable.logo);
        builder.setPositiveButton(text: "OK", listener: null);

        AlertDialog dialog = builder.create();
        dialog.show();
    }
});
```

Activity Configuration

activity_configuration.xml

Hem fet un «ConstraintLayout» i hi hem col·locat 5 botons, 2 «TextViews» i 1 «EditText». Per a fer aquesta composició hem fet ús també de «LinearLayouts» dins del «Layout» principal.

```
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/tvVar1"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:layout_centerVertical="true"
    android:layout_marginTop="24dp"
    android:orientation="horizontal"
    android:padding="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvVar1">

    <Button...>

    <Button...>
```

Background color

LIGHT DARK

Text Size

SMALL LARGE

User Name

CHANGE

ConfigurationActivity.java

Busquem per «ID» tots els elements i després, fent els mètodes «OnClick», podrem canviar la mida de la lletra i el color del fons.

```
//Canvi de color del fons
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // set the color to relative layout
        constraintLayout.setBackgroundResource(R.color.white);
    }
});
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // set the color to relative layout
        constraintLayout.setBackgroundResource(R.color.grey);
    }
});
```

Cal dir que aquests canvis NO quedem traslladats a la resta de aplicació. A la que tornem a la «MainActivity» tot torna a com estava per defecte.

Fragment Home

fragment_home.xml

Aquí doncs començem amb els fragments.

Aquest primer no té res massa complex. Simplement hem afegit un «ImageView» amb l'adreça del logo de l'empresa. El peu de imatge l'hem fet amb el «TextView» que ens venia per defecte.

```
<ImageView
    android:layout_width="231dp"
    android:layout_height="224dp"
    android:src="@drawable/logo"
    app:layout_constraintBottom_toTopOf="@+id/text_home"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

≡ Home



Welcome to Helena-s movie rental!

HomeFragment.java

Hem hagut de canviar l'adreça d'on s'obtenia el text del «TextView» comentant el codi que ja hi havia i afegint aquesta línia de codi:

```
final TextView textView = binding.textHome;
//Posem el text en el textview
textView.setText(getString(R.string.text_home));
//homeViewModel.getText().observe(getViewLifecycleOwner(), textView::setText);
return root;
```


Fragment Calendar

El nom del fragment i el seu contingut no quadren (hi ha un calendari però es diu «SlideShow»). Això és perquè hem fet ús dels fragments que hi havia per defecte i no les hi hem canviat el nom. No ho hem fet perquè no ho hem trobat necessari, sols n'hi ha 3. Si fos una aplicació més complexa amb més fragments i activitats sí que potser faria falta un «refactor».

fragment_slideshow.xml

Havíem doncs de afegir un element que no haguéssim vist fins ara. Seguint doncs amb la temàtica de l'aplicació de lloguer de pel·lícules, hem afegit un calendari. Ho fem simplement afegint un «CalendarView».



```
<CalendarView
    android:layout_width="382dp"
    android:layout_height="443dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Calendar



SlideshowFragment.java

Res a destacar. No hem posat funcionalitats al calendari.

Fragment List

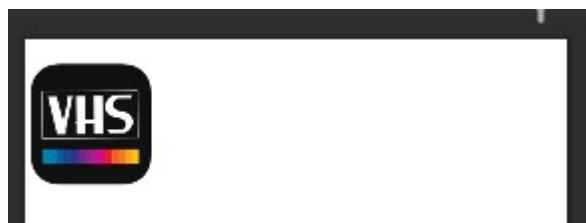
fragment_gallery.xml

Afegim un «RecyclerView». Li donem una id, i li fem «match_parent» tant a l'amplada com a l'alçada.

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recyclerView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

item_rview.xml

Preparem el «Layout» per a cada un dels elements de la llista. Contindrà una imatge i tres textos. Per a que quedin ben alineats farem ús de «LinearLayout» amb orientacions vertical o horitzontal depenent de com convingui.



MovieClass.java

Per a poder crear cadascun dels elements de la llista, hem hagut de crear una classe que pugui crear objectes amb les dades que necessitem. Aquestes serien un «String» títol, un altre «String» per els tags o temàtiques, una «Boolean» per a saber si està reservada o no i un «int» per a posar la ubicació de la imatge.

Creem el constructor i preparem els «get/set» corresponents a cada atribut.

```
public class MovieClass {  
  
    private String title;  
    private String tags;  
    private Boolean reserved;  
    private int imageId;  
  
    public MovieClass(String title, String tags, Boolean reserved, int imageId){  
        this.title = title;  
        this.tags = tags;  
        this.reserved = reserved;  
        this.imageId = imageId;  
    }  
  
    public String getTitle() {return title; }  
    public void setTitle(String title) {this.title = title;}  
  
    public String getTags() {return tags; }  
    public void setTags(String tags) {this.tags = tags;}  
  
    public Boolean getReserved() {return reserved; }  
    public void setReserved(Boolean reserved) {this.reserved = reserved;}  
  
    public int getImageId() {return imageId; }  
    public void setImageId(int imageId) {this.imageId = imageId;}  
}
```

MovieAdapter.java

Hem de crear una classe que ens permeti fer l'enllaç entre el «RecyclerView» i la classe «MovieClass». Així doncs creem el «MovieAdapter.java».

Primer de tot creem dues variables internes: un «ArrayList<MovieClass» anomenat «movieList» i el context. Ja podem fer ara doncs el constructor.

```
ArrayList<MovieClass> movieList;  
Context context;  
  
public MovieAdapter(ArrayList<MovieClass> movieList, Context context){  
    this.movieList = movieList;  
    this.context = context;  
}
```

Li assignem el «Layout» creat anteriorment per els elements de la llista.

```
@NonNull
@Override
public MovieViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_rview, root: null, attachToRoot: false);
    return new MovieViewHolder(view);
}
```

Tenim «MovieViewHolder», que ens permetrà trobar els elements que busquem per la seva id.

```
public class MovieViewHolder extends RecyclerView.ViewHolder{
    TextView movieTitle, movieTags, movieReserved;
    ImageView movieImage;

    public MovieViewHolder(View itemView){
        super(itemView);
        movieTitle = (TextView) itemView.findViewById(R.id.movieName);
        movieTags = (TextView) itemView.findViewById(R.id.movieTag);
        movieReserved = (TextView) itemView.findViewById(R.id.movieReserved);
        movieImage = (ImageView) itemView.findViewById(R.id.movieImg);
    }
}
```

Hem de reescriure el mètode «getItemCount» per tal que ens retorni la llargada de l'array.

```
@Override
public int getItemCount() { return movieList.size(); }
```

Ara doncs ja podem sobreescriure el mètode «onBindViewHolder». Aquest és l'encarregat de assignar cada valor de l' «ArrayList<MovieClass>» a un valor acceptable per un «RecyclerView».

Cal destacar que hi ha un valor que en la classe és booleà i en el «RecyclerView» és un «String», el «reserved». Per a poder fer-ho compatible haurem de fer un simple «if» i canviar el text a enviar segons si és cert o fals. A més, aprofitarem també per a canviar el color del text en concret.

```
@Override
public void onBindViewHolder (MovieViewHolder holder, int position){
    String title = movieList.get(position).getTitle();
    String tags = movieList.get(position).getTags();
    Boolean reserved = movieList.get(position).getReserved();
    int img = movieList.get(position).getImageId();

    holder.movieTitle.setText(title);
    holder.movieTags.setText(tags);

    if (reserved)
    {
        holder.movieReserved.setText("Reserved");
        holder.movieReserved.setTextColor(context.getColor(R.color.red));
    }else{
        holder.movieReserved.setText("Free");
        holder.movieReserved.setTextColor(context.getColor(R.color.green));
    }
    holder.movieImage.setImageResource(img);
}
```

Finalment ja podem posar el mètode «OnClick()» dins d'aquest mateix mètode. La funcionalitat que li donarem serà la de mostrar un «AlertDialog» amb el mateix «Layout» de tal manera que es mostri les dades de l'element on hem fet click.

```
holder.itemView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //Creem alert dialog
        AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(context);
        View dialogView = View.inflate(context, R.layout.item_rview, root: null);
        dialogBuilder.setView(dialogView);

        //Li passem les dades
        TextView txtTitle = dialogView.findViewById(R.id.movieName);
        txtTitle.setText(title);
        TextView txtTags = dialogView.findViewById(R.id.movieTag);
        txtTags.setText(tags);
        TextView txtReserved = dialogView.findViewById(R.id.movieReserved);
        if (reserved)
        {
            txtReserved.setText("Reserved");
            txtReserved.setTextColor(context.getColor(R.color.red));
        }else{
            txtReserved.setText("Free");
            txtReserved.setTextColor(context.getColor(R.color.green));
        }
        ImageView imageView = dialogView.findViewById(R.id.movieImg);
        imageView.setImageResource(img);

        //El mostrem
        AlertDialog alertDialog = dialogBuilder.create();
        alertDialog.show();
    }
});
```


GalleryFragment.java

Finalment ja podem passar les dades que vulguem al nostre «RecyclerView» mitjançant la funció que hem creat «OmplirLlista()»

Veiem doncs que en el nostre «ArrayList<MovieClass» movieList li passem instàncies de «MovieClass.java».

```
private void OmplirLlista() {
    movieList.add(new MovieClass("Akira", "Anime" + " " + "Action", reserved: true, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Blade Runner", "Action", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("But I-m a Cheerleader!", "Teen" + " " + "Fun", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Cube", "Horror", reserved: true, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Kill Bill vol.1", "Action", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Kill Bill vol.2", "Action", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Knives Out", "Action", reserved: true, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("The Lego Movie", "Fun", reserved: true, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Mean Girls", "Teen" + " " + "Fun", reserved: true, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Old Boy", "Action" + " " + "Drama", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Perfect Blue", "Anime" + " " + "Horror", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Schindler-s List", "Drama", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Bridge to Terabithia", "Fun", reserved: true, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("Titanic", "Drama", reserved: false, R.drawable.ic_claqueta_foreground));
    movieList.add(new MovieClass("The Lego Movie", "Fun", reserved: false, R.drawable.ic_claqueta_foreground));
}
```

Un cop tenim entrada la taula podem fer servir la classe «MovieAdapter» dins del mètode «OnCreateView» per a mostrar les dades.

```
public View onCreateView(@NonNull LayoutInflater inflater,
                        ViewGroup container, Bundle savedInstanceState) {
    binding = FragmentGalleryBinding.inflate(inflater, container, attachToParent: false);
    View root = binding.getRoot();

    //Creem la llista
    View view = inflater.inflate(R.layout.fragment_gallery, container, attachToRoot: false);
    RecyclerView rv = (RecyclerView) view.findViewById(R.id.recyclerView);
    movieList = new ArrayList<>();
    rv.setLayoutManager(new LinearLayoutManager(getContext()));

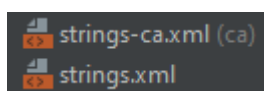
    //Omplim la llista
    OmplirLlista();

    //Preparem adaptador
    MovieAdapter adapter = new MovieAdapter(movieList, getContext());
    rv.setAdapter(adapter);
    return view;
}
```

Altres

Idiomes

Des del començament hem estat col·locat tots els «Strings» dins del fitxer «values/strings.xml». Si creem una nova carpeta i fitxer «values-ca/strings-ca.xml» i hi copiem tots els textos i els traduïm, automàticament ens canviarà l'idioma depenent de les preferències del telèfon.



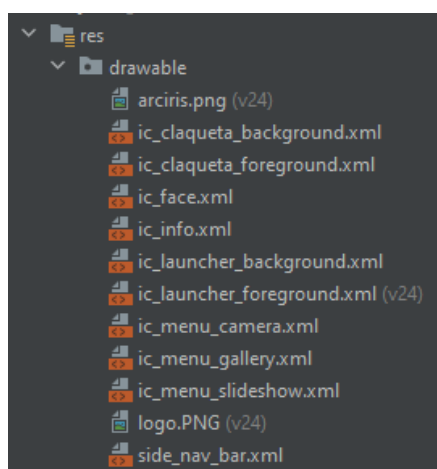
Colors

En aquesta aplicació hem fet servir alguns colors que no venien per defecte. Per a fer-ho, hem obert el fitxer «values/colors.xml» i hi afegim els colors amb codi RGB que creiem oportuns.



Imatges/Icones

Per a fer servir imatges personalitzades hem tant copiat i enganxat directament imatges a la carpeta «res/drawable» o n'hem creat de noves amb «new-ImageAsset-ClipArt».



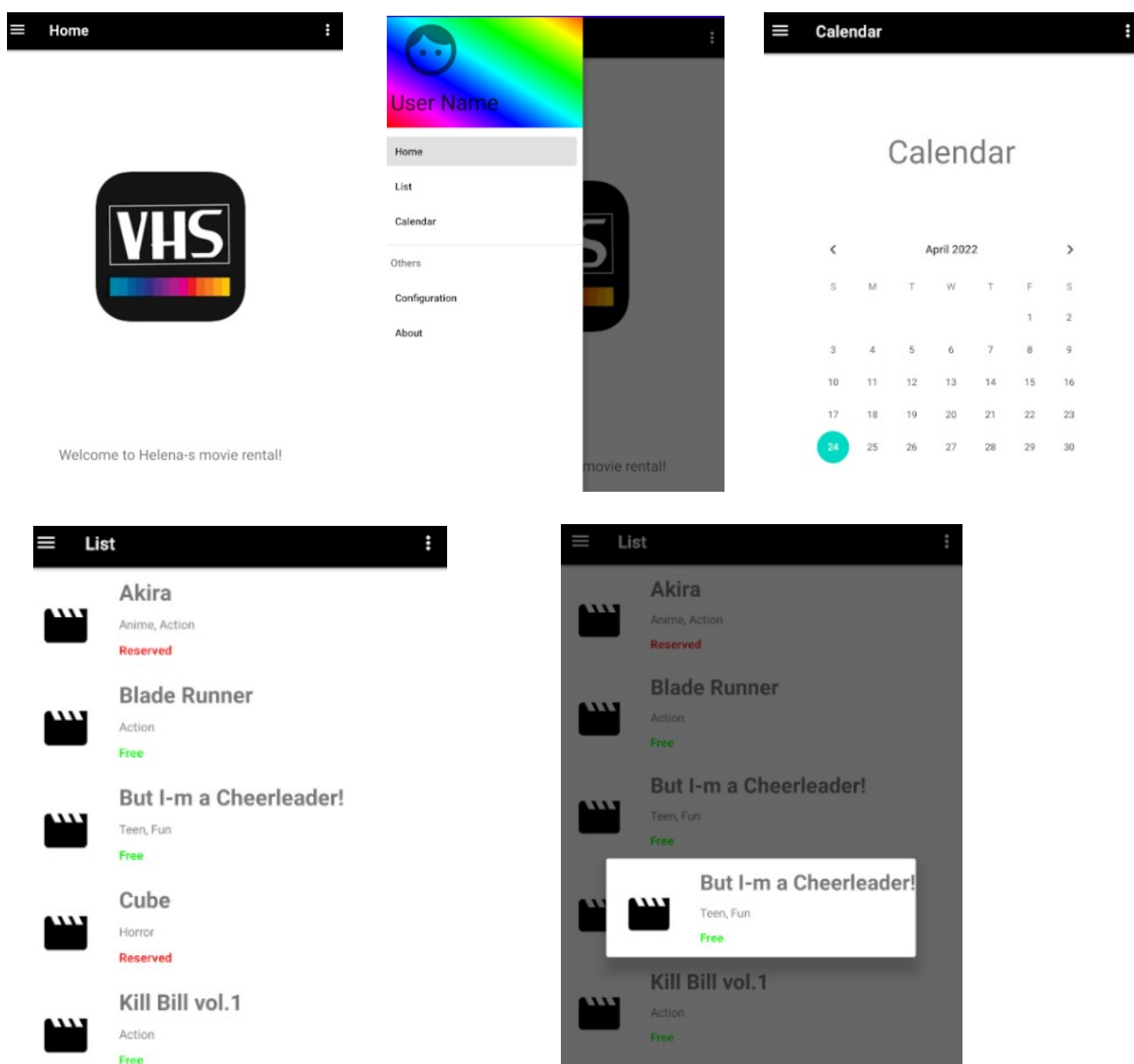


Output

Link Video

<https://drive.google.com/file/d/1HCOT-YeYB94vs20B6CSsPg5PsE696DSU/view?usp=sharing>

Captures





Configuration

Background color

LIGHTDARK

Text Size

SMALLLARGE

User Name

CHANGE

Configuration

Background color

LIGHTDARK

Text Size

SMALLLARGE

User Name

CHANGE

Configuration

Background color

LIGHTDARK

Text Size

SMALLLARGE

User Name

CHANGE

About

If you have any questions contact us at
helenaRental@gmail.com

i

About

Information

Helena Madrenys, 2022

OK

i