|  |  |
| --- | --- |
| **Project Case** |  |
| ISYS6203003  Mobile Application Development |
| **Computer Science** | **O212-ISYS6203-SO02-00** |
| ***Valid on*** *Odd Semester Year 2021/2022* | **Revision 00** |

1. Seluruh kelompok tidak diperkenankan untuk:

*The whole group is not allowed to:*

* + - Melihat sebagian atau seluruh proyek kelompok lain,

*Seeing a part or the whole project from other groups*

* + - Menyadur sebagian maupun seluruh proyek dari buku,

*Adapted a part or the whole project from the book*

* + - Mendownload sebagian maupun seluruh proyek dari internet,

*Downloading a part or the whole project from the internet,*

* + - Mengerjakan soal yang tidak sesuai dengan tema yang ada di soal proyek,

*Working with another theme which is not in accordance with the existing theme in the matter of the project,*

* + - Melakukan tindakan kecurangan lainnya,

*Committing other dishonest actions,*

* + - Secara sengaja maupun tidak sengaja melakukan segala tindakan kelalaian yang menyebabkan hasil karyanya berhasil dicontek oleh orang lain / kelompok lain.

*Accidentally or intentionally conduct any failure action that cause the results of the project was copied by someone else / other groups.*

1. Jika kelompok terbukti melakukan tindakan seperti yang dijelaskan butir 1 di atas, maka **nilai kelompok** yang melakukan kecurangan (menyontek maupun dicontek) akan di – **NOL** – kan.

*If the group is proved to the actions described in point 1 above, the score of the group which committed dishonest acts (cheating or being cheated) will be “Zero”*

1. Perhatikan jadwal pengumpulan proyek, segala jenis pengumpulan proyek di luar jadwal tidak dilayani.

*Pay attention to the submission schedule for the project, all kinds of submission outside the project schedule will not be accepted*

1. Jangan lupa untuk melihat kriteria penilaian proyek yang ditempel di papan pengumuman, atau tanya asisten anda.

*Don’t forget to look at the project assessment criteria that posted on the announcement board, or ask your teaching assistant.*

1. Persentase penilaiaan untuk matakuliah ini adalah sebagai berikut:

*Marking percentage for this subject is described as follows:*

|  |  |  |
| --- | --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* | **UAP**  *Final Exam* |
| 40% | 60% | - |

1. Software yang digunakan pada matakuliah ini adalah sebagai berikut:

*Software will be used in this subject are described as follows:*

|  |
| --- |
| **Software**  *Software* |
| Android Studio 3  Android SDK 23  ADT (Android Development Tools) |

## Ekstensi file yang harus disertakan dalam pengumpulan tugas mandiri dan proyek untuk matakuliah ini adalah sebagai berikut:

*File extensions should be included in assignment and project collection for this subject are described as follows:*

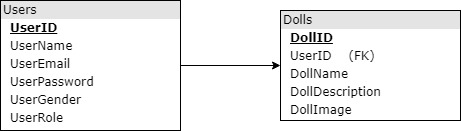
|  |  |
| --- | --- |
| **Tugas Mandiri**  *Assignment* | **Proyek**  *Project* |
| Folder Project (JAVA, CLASS, PROJECT, CLASSPATH, APK, DEX, CACHE, XML, JAR, image files (JPG / PNG / GIF), TXT) | Folder Project (JAVA, CLASS, PROJECT, CLASSPATH, APK, DEX, CACHE, XML, image files (JPG / PNG / GIF), JAR, TXT) |

## Soal

*Case*

**BlueDoll**

**BlueDoll** is one of the most famous company which provides high quality dolls worldwide. As it stands, currently **BlueDoll** is trying to reach even more customer by creating a mobile application based on **Android Operating System** which may be downloaded by customers, thus increasing **BlueDoll** popularity. The application provides features such as **user** **authentication**, **view** **all dolls**, **share** the joy of dolls, and check for **BlueDoll** stores location via **Google Map**.

****

**Figure 1. ERD of the application’s database**

As stated above, this application will then need several forms to fulfill its purpose. The breakdown of those forms described below:

1. **Application Header**

This application has a header that is **displayed whenever the user managed to login into the application**. This header consists of several parts which are:

* **Navigation Drawer**:
  + This menu is represented by a **hamburger icon** located in the **left side of the header**. This menu contains:
    - **View all dolls** : Direct the user to the **view all dolls form**.
    - **Insert dolls** : Direct the user to the **modify/insert dolls form**.
    - **Logout** : Direct the user to the **login form**.
* **Options Menu**:
  + This menu is represented by a **kebab icon** located in the **right side of the header**. This menu contains:
    - **Our location** : Direct the user to the **About Us form.**

This header can only be seen on **several pages after the user has logged in** to the application. Those forms are:

* **View all dolls form**
* **About us form**

1. **Login Form**

This form will be used to **authenticate users** to the application. The authentication process requires the user to **input email** and **password** which has been **registered in the application’s database** beforehand. The page contains several components and validation which are:

* Use **EditText** for **email** and **password** input field. Adjust the component to suit the right purpose.
* Use **Button** for the user to log in.

This button will be used to **validate and authenticate user** to the application. The **login** processes are:

* + Validate that **email must be filled**.
  + Validate that **password must be filled**.
  + Validate that **email and password** is **registered** **in the application’s database**.
* **If any of the validation fails**, the application will **show an error message** according to the failure using **Toast**.
* **Redirect the logged in user** to the **View All Dolls Form**.
* Use **TextView** that contains “**Register here!**”. This component will **direct users** to the **Registration Form** whenever clicked by the user.

1. **Registration Form**

This form will be used to **register a** **new user** which will then be **stored** in the **application’s database**. The **registration** process **requires** **email**, **name**, **password**, **password confirmation**,and **terms and conditions agreement**. The registered user’s default role will be **member**. The requirements for this form are:

* Use **EditText** for **email, name, password,** and **password confirmation** input field. Components must be **adjusted to suit the right purpose**.
* Use **RadioButton** to select gender between **male** or **female**.
* Use **DatePicker** to select user **birthday**.
* Use **CheckBox** for **terms and conditions agreement**.
* Use **Button** for **Register**.

This button will be used to **validate and register user** to the application. The **registration** processes are:

* + Validate that **email** must be filled with a **valid email format**.
  + Validate that **name must be filled**.
  + Validate that **password must be filled, contains at least 6 characters**.
  + Validate that **password confirmation must be filled** and **match with password**.
  + Validate that one **gender** must be chosen by the user.
  + Validate that registrant **has** **agreed on the terms and conditions**.
* Validate that **email must be unique** and **not registered by other users**.
* **User ID** will be **generated** with following format.

|  |
| --- |
| **USXXX** |

**Note** : **X** is a number obtained **incrementally** between **0-9**.

(**Last** **userID** **+ 1**)

**Example** : **US024**

* + **If any of the validation fails,** the application will **show an error message** according to the failure using **Toast**.
  + **Otherwise, register** the **given information** and **store it** in the **application’s database**. User will then be directed to the **Login Form**.
* Use **TextView** for **Login**.

This component contains “**Already has an account? Login** **here**” which will **direct the user** to the **Login Form** when clicked.

1. **View All Dolls Form**

This form will **list all available dolls** data from the **application’s database**. Dolls listed in this form are inserted by other users. The requirements for this form are:

* Use **ListView** to **list all available dolls**.
  + Use **ImageView** to **show doll** **image**.
  + Use **TextView** to **show doll name, doll creator** and **description**.
  + Use **Button** for:
    - **View**

If this button is selected, **redirect the user to the View Doll Detail form** with the selected data in the **ListView**.

* + - **Edit**

If this button is selected, **redirect the user to the Modify Doll Detail form** with the selected data in the **ListView**.

* + - **Delete**

If this button is selected, **remove the doll from the application database**. Several notes for this button are:

* This button will only work for **doll owners** and **administrator**.
* If the user **is not eligible to delete the doll**, then show a message using **Toast** that contains “**You are not eligible to remove this doll**”.

1. **View Doll Detail Form**

This form will **show** **detailed doll information** for a specific doll from the **application’s database**. This form also allow user to **share** the joy of dolls **via SMS**. The requirements for this form are:

* Use **TextView** to show **doll’s name** and **description**.
* Use **ImageView** to show **doll’s image**.
* Use **EditText** for **phone number** input field.
* Use **Button** for **Share**.

This button will be used to **validate** and **send SMS** to the specified phone number. The **validation** processes are:

* Validate that **phone number must be filled**.
* Validate that **application has permission** to **send SMS**.
* **Send SMS** to the specified phone number, informing them about the doll, for example:

Hey, check this doll from BlueDoll! It’s the **[doll’s name]** and it’s so awesome!

- **[logged in user’s name]**

* Show proper **error message** for each validation process.

1. **Modify Doll Form**

This can be accessed through **2 different methods** which are the “**Insert new doll**” option from the application **navigation bar** and the “**edit dolls**” button from the **View Dolls edit button**. The form **allows** the user to **insert** or **update** a specific doll from the **application’s database**. The requirements for this form are:

* If the form is accessed through the “**edit dolls**” button, then fill **all the inputs with the doll data** **selected** from the previous page.
* Use **EditText** for **name and description**.
* Use **Spinner** for **image**.
* Use **Button** for **save** and **delete**.
  + **Save Button**

The application will **validate all input** when the **save button is pressed** according to the following criteria:

* Validate that **doll name must be filled**.
* Validate that **doll name must be unique** and there are **no other dolls** **with the same name** in the **application’s database**.
* Validate that **doll description must be filled**.
* Validate that **doll image must be chosen**.
* Show proper **error message** for each validation process.

After all validation has been completed, the application will **read all input value** and **insert/update** them into the application database according to the following detail:

* If this form is accessed through the “**Insert doll**” option from the **navigation bar**, then **insert the data** into the application database.
* Else if this form is accessed through the “**edit dolls**” button from the **View Dolls edit button**, then **modify the doll data** in the database according to the user input.

1. **About Us Form**

This form will **show the user of BlueDoll location with a marker** according to the **JSON data given** from **BlueDoll**. The requirements for this form are:

* Provide a **TextView** that serves as a header that writes “**About Us**”.
* Provide another **TextView** that Shows the company description.
* Use **Google Play Services** for **Android Maps SDK**.
* Use **Volley** library to **fetch and extract information** fromthe given **JSON data**. The **JSON data** is available at <https://bit.ly/3kwdAd5>.
* Put **marker** on the specified **location (latitude & longitude)** and name the **marker**.

**Note:**

1. There **must be** at least 1 **administrator** **user** which could not be delete and you may **hard-code it into the application.**
2. **JSON data** used in **Google Map Location Form** must befetched from the given link above. **Please note that hard-coded JSON data will not be scored.**

**If there is something you don’t understand, feel free to ask your laboratory assistant!**