

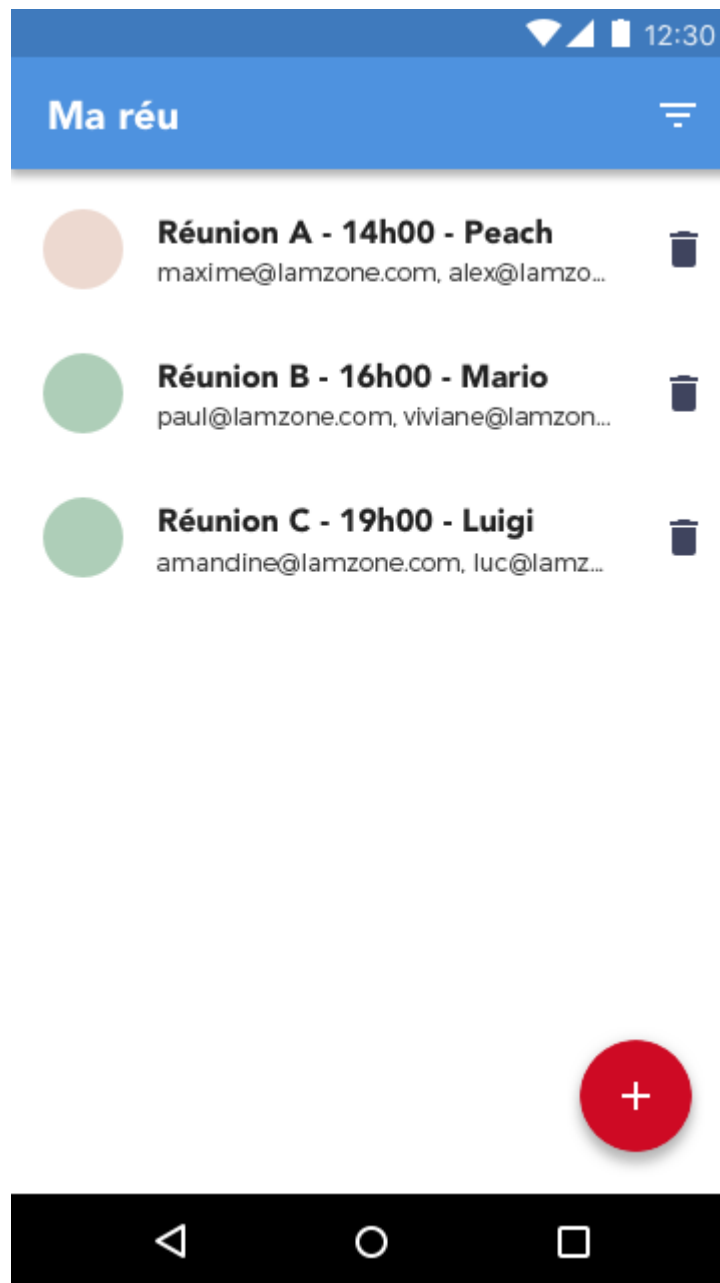
# Présentation Projet 4 :

🕒 Created	@22 août 2022 23:43
🏷 Tags	

## Projet 4 : Créez une application de gestion de réunions :

Valentin Famery

## Cahiers des charges :



1

## Sommaire de la réalisation de l'application MaRéu :

- Fonctionnalité 1 : Afficher 3 réunions initial dans le fragment de l'activité principale sous forme de RecyclerView (p. 3-8)
- Fonctionnalité 2 : Supprimer une réunion (p. 9-13)

- Fonctionnalité 3 : Ajouter une nouvelle réunion dans une nouvelle activité d'ajout ,tout en gerant la disponibilité des salles avec les horaires (p. 14-18)
- Fonctionnalité 4 : Filtrer les réunions par date (p. 19-23)
- Fonctionnalité 5 : Filtrer les réunions par salle (p. 24-28)
- Fonctionnalité 6 : Filtrer pour afficher toutes les réunions (p. 29-33)
- Fonctionnalité 7 : Afficher les details d'une réunion dans une activité de detail (p. 34-38)
- Fonctionnalité 8 : Gestion de plusieurs tailles d'écran (p. 39)

2

## **Fonctionnalité 1 : Création des classes Models**

### **Classe RoomMeeting :**

```

package com.example.projet_4_oc_maru.models;

import androidx.annotation.NonNull;

public class RoomMeeting {
    private final String nameRoomMeeting;
    private final int roomMeetingColor;
    private final Integer id;

    public RoomMeeting(Integer id, String nameRoomMeeting, int roomMeetingColor) {
        this.id = id;
        this.nameRoomMeeting = nameRoomMeeting;
        this.roomMeetingColor = roomMeetingColor;
    }

    public Integer getId() {
        return id;
    }

    public String getNameRoomMeeting() {
        return nameRoomMeeting;
    }

    public int getRoomMeetingColor() {
        return roomMeetingColor;
    }

    @NonNull
    public static RoomMeeting getRoomMeetingById(Integer id) {
        for (RoomMeeting roomMeeting : getAllRoomMeetings()) {
            if (roomMeeting.id.equals(id))
                return roomMeeting;
        }
        return null;
    }

    @NonNull
    public static RoomMeeting[] getAllRoomMeetings() {
        return new RoomMeeting[]{
            new RoomMeeting(1, "Birdo", 0xFF6D726F),
            new RoomMeeting(2, "Daisy", 0xFF5D210C),
            new RoomMeeting(3, "Donkey Kong", 0xFFDF3CB8),
            new RoomMeeting(4, "Harmonie", 0xFF813CDF),
            new RoomMeeting(5, "Luigi", 0xFF1527DD),
            new RoomMeeting(6, "Mario", 0xFF3CBCDF),
            new RoomMeeting(7, "Peach", 0xFF23CAC3),
            new RoomMeeting(8, "Toad", 0xFF15DD42),
            new RoomMeeting(9, "Wario", 0xFFDD2415),
            new RoomMeeting(10, "Yoshi", 0xFFFF1F114),
        };
    }
}

```

# Classe Meeting :

```
package com.example.projet_4_oc_maru.models;

import org.joda.time.DateTime;
import java.util.List;

public class Meeting {

    private int id;
    private final String subject;
    private final DateTime dateTimeBegin;
    private final DateTime dateTimeEnd;
    private final List<String> participants;
    private final RoomMeeting meetingRoom;

    public Meeting(int id, String subject, DateTime dateTimeBegin, DateTime dateTimeEnd, List<String>
participants, RoomMeeting meetingRoom) {
        this.id = id;
        this.subject = subject;
        this.dateTimeBegin = dateTimeBegin;
        this.dateTimeEnd = dateTimeEnd;
        this.participants = participants;
        this.meetingRoom = meetingRoom;
    }

    /**
     * Getters
     */
    public int getId() {
        return id;
    }

    public String getSubject() {
        return subject;
    }

    public DateTime getDateTimeBegin() {
        return dateTimeBegin;
    }

    public DateTime getDateTimeEnd() {
        return dateTimeEnd;
    }

    public List<String> getParticipants() {
        return participants;
    }

    public RoomMeeting getMeetingRoom() {
        return meetingRoom;
    }

    /**
     * Setters
     */
    public void setId(int id) {
        this.id = id;
    }
}
```

## Création de MeetingApiService :

### Classe DummyMeetingGenerator :

```
package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.models.Meeting;
import com.example.projet_4_oc_maru.models.RoomMeeting;

import org.joda.time.DateTime;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public abstract class DummyMeetingGenerator {
    public static List<Meeting> DUMMY_ROOM_MEETS = Arrays.asList(
        new Meeting(1, "Logistique", new DateTime(2021, 02, 14, 14, 00), new DateTime(2021, 02,
14, 15, 00), Arrays.asList("maxime@lamzone.com", "alex@lamzone.com"),
RoomMeeting.getRoomMeetingById(7)),
        new Meeting(2, "Qualité", new DateTime(2021, 02, 15, 15, 00), new DateTime(2021, 02, 15,
16, 00), Arrays.asList("paul@lamzone.com", "viviane@lamzone.com"),
RoomMeeting.getRoomMeetingById(6)),
        new Meeting(3, "Reception", new DateTime(2021, 02, 16, 19, 00), new DateTime(2021, 02,
16, 20, 00), Arrays.asList("amandine@lamzone.com", "luc@lamzone.com"),
RoomMeeting.getRoomMeetingById(5))
    );

    static List<Meeting> generateMeetings() {
        return new ArrayList<>(DUMMY_ROOM_MEETS);
    }
}
```

### Classe Interface MeetingApiService :

```
package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

/**
 * Meeting API client
 */
public interface MeetingApiService {
    /**
     * Get all my RoomMeets
     *
     * @return {@link List}
     */
    List<Meeting> getMeetings();
}
```

**Classe DummyMeetingApiService :**

```
package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.di.DI;
import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

public class DummyMeetingApiService implements MeetingApiService {

    private final List<Meeting> meetings = DummyMeetingGenerator.generateMeetings();

    @Override
    public List<Meeting> getMeetings() {
        return meetings;
    }
}
```

**Classe DI :**



```
package com.example.projet_4_oc_maru.di;

import com.example.projet_4_oc_maru.service.DummyMeetingApiService;
import com.example.projet_4_oc_maru.service.MeetingApiService;

public class DI {
    private static MeetingApiService apiService;

    static {
        apiService = new DummyMeetingApiService();
    }

    public static MeetingApiService getMeetingApiService() {
        return apiService;
    }

    public static MeetingApiService getNewInstanceApiService() {
        return new DummyMeetingApiService();
    }
}
```

4

## Implémentation de la fonctionnalité 1 sur Android Studio :

5

## Presentation des tests de la Fonctionnalité 1 :

6

## Resultats des Tests :

### Test Unitaires :

<code>UnitTest.getMeetingsWithSuccess</code>	passed	1 ms
--	--------	------

### Test Instrumentalisés :

<code>listMeetingContainsAllMeetingInitial</code>	passed	8.92 s
---	--------	--------

7

## Démo de la Fonctionnalité 1 :

8

## Fonctionnalité 2 : Supprimer une réunion

### Classe MeetingApiService :

```
package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

/**
 * Meeting API client
 */
public interface MeetingApiService {

    /**
     * Deletes a meeting
     *
     * @param meeting
     */
    void deleteMeeting(Meeting meeting);

}
```

**ClasseDummyMeetingApiService :**

```

package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.di.DI;
import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

public class DummyMeetingApiService implements MeetingApiService {

    @Override
    public void deleteMeeting(Meeting meeting) {
        meetings.remove(meeting);
    }
}

```

9

## Implémentation de la Fonctionnalité 2 sur Android Studio :

10

## Presentation des tests de la Fonctionnalité 2 :

11

## Resultats des Tests :

## Test Unitaires :

`UnitTest.deleteMeetingWithSuccess`

passed

0 ms

## Test instrumentalisés :

`deleteMeeting`

passed

3.01 s

12

## Démo de la Fonctionnalité 2 :

13

**Fonctionnalité 3 : Ajouter une nouvelle réunion dans une nouvelle activité d'ajout ,tout en gerant la disponibilité des salles avec les horaires**

**Classe MeetingApiService :**

```
package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

/**
 * Meeting API client
 */
public interface MeetingApiService {

    /**
     * Create a meeting
     *
     * @param meeting
     */
    void createMeeting(Meeting meeting);

    Boolean theRoomIsAvailableOrNotAvailable(DateTime finalDateTimeBegin, DateTime finalDateTimeEnd,
Integer positionRoom);
}
```

## ClasseDummyMeetingApiService :

```

package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.di.DI;
import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

public class DummyMeetingApiService implements MeetingApiService {

    @Override
    public void createMeeting(Meeting meeting) {
        meetings.add(meeting);
    }

    public Boolean theRoomIsAvailableOrNotAvailable(DateTime finalDateTimeBegin, DateTime
finalDateTimeEnd, Integer positionRoom) {
        Boolean reserved = false;
        for (Meeting m : DI.getMeetingApiService().getMeetings()) {
            if ((m.getMeetingRoom().getId().equals(positionRoom) &&
finalDateTimeBegin.isBefore(m.getDateTimeEnd()) && finalDateTimeBegin.isAfter(m.getDateTimeBegin()))
                || (m.getMeetingRoom().getId().equals(positionRoom) &&
finalDateTimeEnd.isBefore(m.getDateTimeEnd()) && finalDateTimeEnd.isAfter(m.getDateTimeBegin()))
                || (m.getMeetingRoom().getId().equals(positionRoom) &&
finalDateTimeBegin.isEqual(m.getDateTimeBegin())
                || (m.getMeetingRoom().getId().equals(positionRoom) &&
finalDateTimeEnd.isEqual(m.getDateTimeEnd()))
                || (m.getMeetingRoom().getId().equals(positionRoom) &&
m.getDateTimeBegin().isAfter(finalDateTimeBegin) && m.getDateTimeBegin().isBefore(finalDateTimeEnd))
                || (m.getMeetingRoom().getId().equals(positionRoom) &&
m.getDateTimeEnd().isBefore(finalDateTimeEnd) && m.getDateTimeEnd().isAfter(finalDateTimeBegin))
            ) {
                reserved = true;
                break;
            }
        }

        return reserved;
    }
}

```

14

## Implémentation de la Fonctionnalité 3 sur Android Studio :

15

## Presentation des tests de la Fonctionnalité 3 :

16

### Resultats des Tests :

#### Test Unitaires :

UnitTest.addMeetingWithSuccess	passed	0 ms
UnitTest.theRoomIsNotAvailable	passed	2 ms

#### Test Instrumentalisés :

addMeeting	passed	34.37 s
theRoomIsNotAvailable	passed	36.71 s

17

## Démo de la Fonctionnalité 3 :

18

## Fonctionnalité 4 : Filtrer les réunions par date

### Classe Meeting :



```

package com.example.projet_4_oc_maru.models;

import org.joda.time.DateTime;
import java.util.List;

public class Meeting {

    private boolean isMeetingInFilterList;

    public Meeting(int id, String subject, DateTime dateTimeBegin, DateTime dateTimeEnd, List<String>
participants, RoomMeeting meetingRoom) {
        this.id = id;
        this.subject = subject;
        this.dateTimeBegin = dateTimeBegin;
        this.dateTimeEnd = dateTimeEnd;
        this.participants = participants;
        this.meetingRoom = meetingRoom;
        this.isMeetingInFilterList = false;
    }

    public boolean isMeetingInFilterList() {
        return isMeetingInFilterList;
    }

    public void setMeetingInFilterList(boolean isMeetingInFilterList) {
        this.isMeetingInFilterList = isMeetingInFilterList;
    }

}

```

## Classe MeetingApiService :

```
package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

/**
 * Meeting API client
 */
public interface MeetingApiService {

    List<Meeting> getMeetingsFilterDate(LocalDate date);

    void resetFilter();
}
```

**Classe DummyMeetingApiService :**

```

package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.di.DI;
import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

public class DummyMeetingApiService implements MeetingApiService {

    @Override
    public List<Meeting> getMeetingsFilterDate(LocalDate date) {

        resetFilter();

        for (Meeting m : meetings) {
            if (m.getDateTimeBegin().toLocalDate().equals(date)) {
                m.setMeetingInFilterList(true);
            }
        }
        return meetings;
    }

    @Override
    public void resetFilter() {
        for (Meeting m : meetings) {
            m.setMeetingInFilterList(false);
        }
    }
}

```

## Implémentation de la Fonctionnalité 4 sur Android Studio :

20

## Presentation des tests de la Fonctionnalité 4 :

21

## Resultats des Tests :

### Test Unitaires :

<code>UnitTest.getMeetingFilterDate</code>	passed	11 ms
--	--------	-------

### Test Instrumentalisé :

<code>filterDatelsWorking</code>	passed	11.54 s
----------------------------------	--------	---------

22

## Démo de la Fonctionnalité 4 :

23

## Fonctionnalité 5 : Filtrer les réunions par salle

### Classe MeetingApiService :

```
package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

/**
 * Meeting API client
 */
public interface MeetingApiService {

    /**
     * Get all Meetings in order of Rooms (liste filtrée)
     */
    List<Meeting> getMeetingsFilterRoom(String salle);

}
```

## Classe DummyMeetingApiService :

```

package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.di.DI;
import com.example.projet_4_oc_maru.models.Meeting;

import org.joda.time.DateTime;
import org.joda.time.LocalDate;

import java.util.List;

public class DummyMeetingApiService implements MeetingApiService {

    @Override
    public List<Meeting> getMeetingsFilterRoom(String salle) {

        resetFilter();

        for (Meeting m : meetings) {
            if (m.getMeetingRoom().getNameRoomMeeting().equals(salle)) {
                m.setMeetingInFilterList(true);
            }
        }

        return meetings;
    }
}

```

24

## Implémentation de la Fonctionnalité 5 sur Android Studio :

25

## Presentation des tests de la Fonctionnalité 5 :

26

## Resultats des Tests :

### Test Unitaires :

<code>UnitTest.getMeetingsFilterRoom</code>	passed	193 ms
---	--------	--------

### Test Instrumentalisé :

<code>filterRoomIsWorking</code>	passed	7.04 s
----------------------------------	--------	--------

27

## Démo de la Fonctionnalité 5 :

28

## Fonctionnalité 6 : Filtrer pour afficher toutes les réunions

### Classe DummyMeetingApiService :

```

package com.example.projet_4_oc_maru.service;

import com.example.projet_4_oc_maru.di.DI;
import com.example.projet_4_oc_maru.models.Meeting;
import org.joda.time.DateTime;
import org.joda.time.LocalDate;
import java.util.List;

public class DummyMeetingApiService implements MeetingApiService {

    private final List<Meeting> meetings = DummyMeetingGenerator.generateMeetings();

    @Override
    public List<Meeting> getMeetings() {
        resetFilter();
        return meetings;
    }

}

```

29

## Implémentation de la Fonctionnalité 6 sur Android Studio :

30

## Presentation des tests de la Fonctionnalité 6 :

31

## Resultats des Tests :

### Test Instrumentalisé :



32

## Démo de la Fonctionnalité 6 :

33

## Fonctionnalité 7 : Afficher les details d'une réunion dans une activité de detail

Classe Meeting :

```

package com.example.projet_4_oc_maru.models;

import android.os.Parcel;
import android.os.Parcelable;
import org.joda.time.DateTime;
import java.io.Serializable;
import java.util.List;

public class Meeting implements Parcelable {

    protected Meeting(Parcel in) {
        id = in.readInt();
        subject = in.readString();
        dateTimeBegin = (DateTime) in.readSerializable();
        dateTimeEnd = (DateTime) in.readSerializable();
        participants = (List<String>) in.readSerializable();
        meetingRoom = in.readParcelable(RoomMeeting.class.getClassLoader());
    }

    public static final Creator<Meeting> CREATOR = new Creator<Meeting>() {
        @Override
        public Meeting createFromParcel(Parcel in) {
            return new Meeting(in);
        }

        @Override
        public Meeting[] newArray(int size) {
            return new Meeting[size];
        }
    };

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeInt(id);
        dest.writeString(subject);
        dest.writeSerializable(dateTimeBegin);
        dest.writeSerializable(dateTimeEnd);
        dest.writeSerializable((Serializable) participants);
        dest.writeParcelable(meetingRoom, flags);
    }
}

```

## Classe RoomMeeting :

```

package com.example.projet_4_oc_maru.models;

import android.os.Parcel;
import android.os.Parcelable;

import androidx.annotation.NonNull;

public class RoomMeeting implements Parcelable {

    protected RoomMeeting(Parcel in) {
        nameRoomMeeting = in.readString();
        roomMeetingColor = in.readInt();
        if (in.readByte() == 0) {
            id = null;
        } else {
            id = in.readInt();
        }
    }

    public static final Creator<RoomMeeting> CREATOR = new Creator<RoomMeeting>() {
        @Override
        public RoomMeeting createFromParcel(Parcel in) {
            return new RoomMeeting(in);
        }

        @Override
        public RoomMeeting[] newArray(int size) {
            return new RoomMeeting[size];
        }
    };

    @Override
    public int describeContents() {
        return 0;
    }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(nameRoomMeeting);
        dest.writeInt(roomMeetingColor);
        if (id == null) {
            dest.writeByte((byte) 0);
        } else {
            dest.writeByte((byte) 1);
            dest.writeInt(id);
        }
    }
}

```

## Implémentation de la Fonctionnalité 7 sur Android Studio :

35

## Presentation des tests de la Fonctionnalité 7 :

36

## Resultats des Tests :

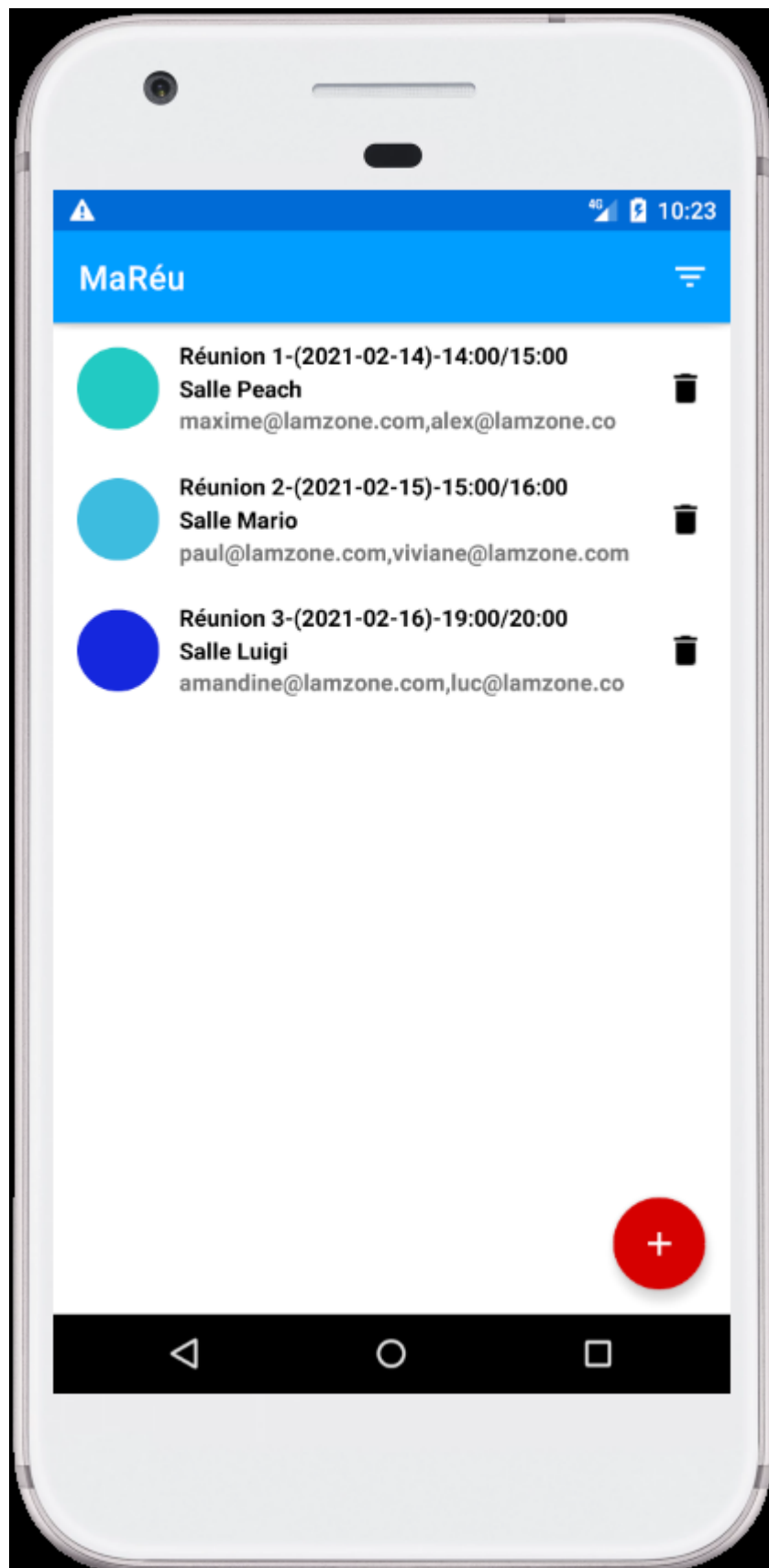
 launchDetailActivityAndContainMeetingClicked passed 8.81 s

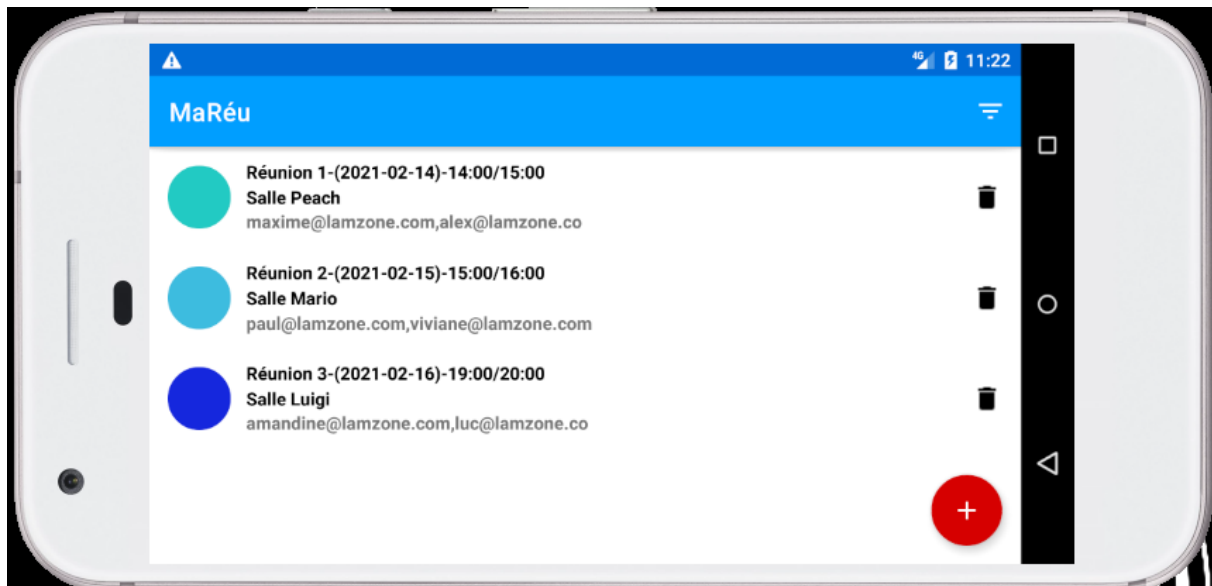
37

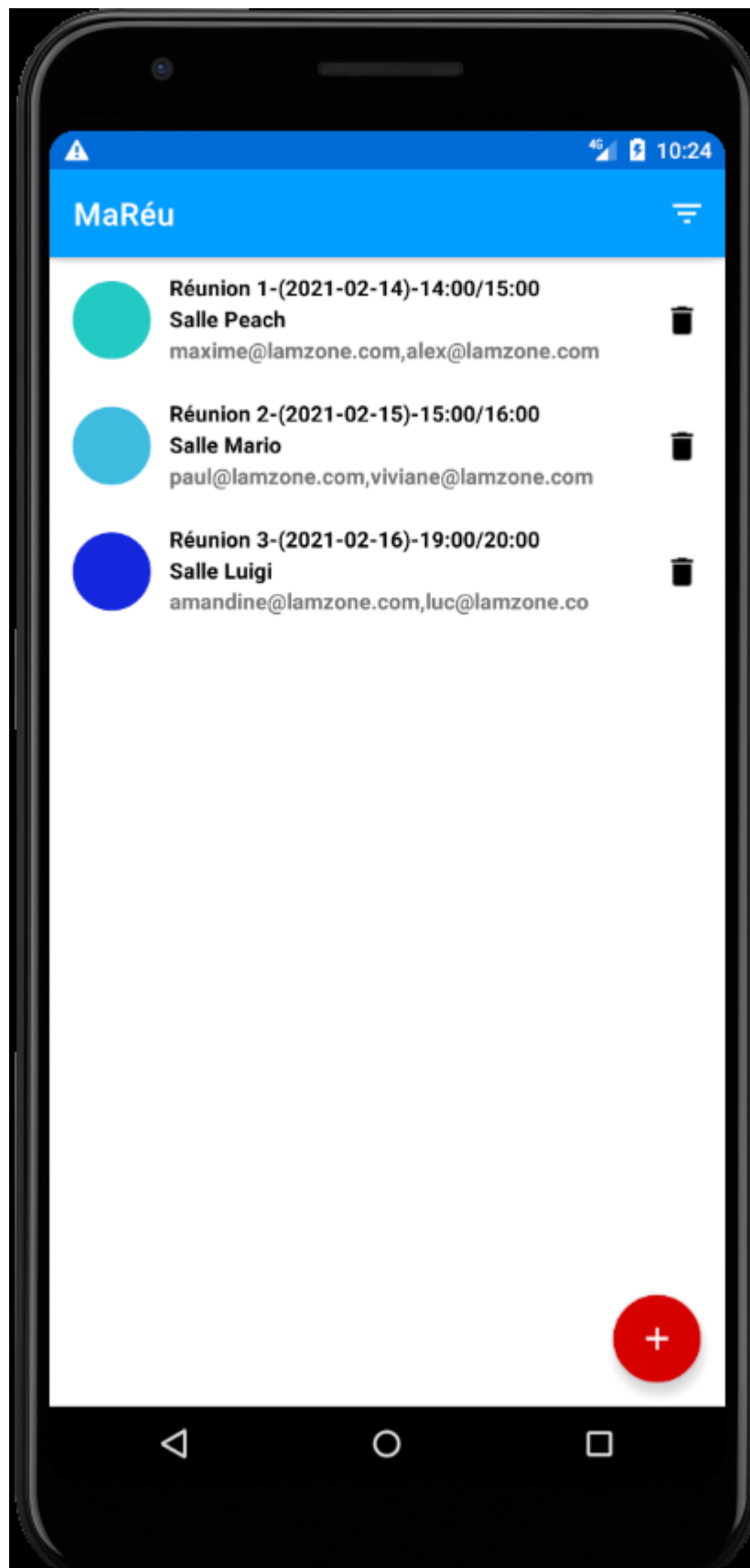
## Démo de la Fonctionnalité 7 :

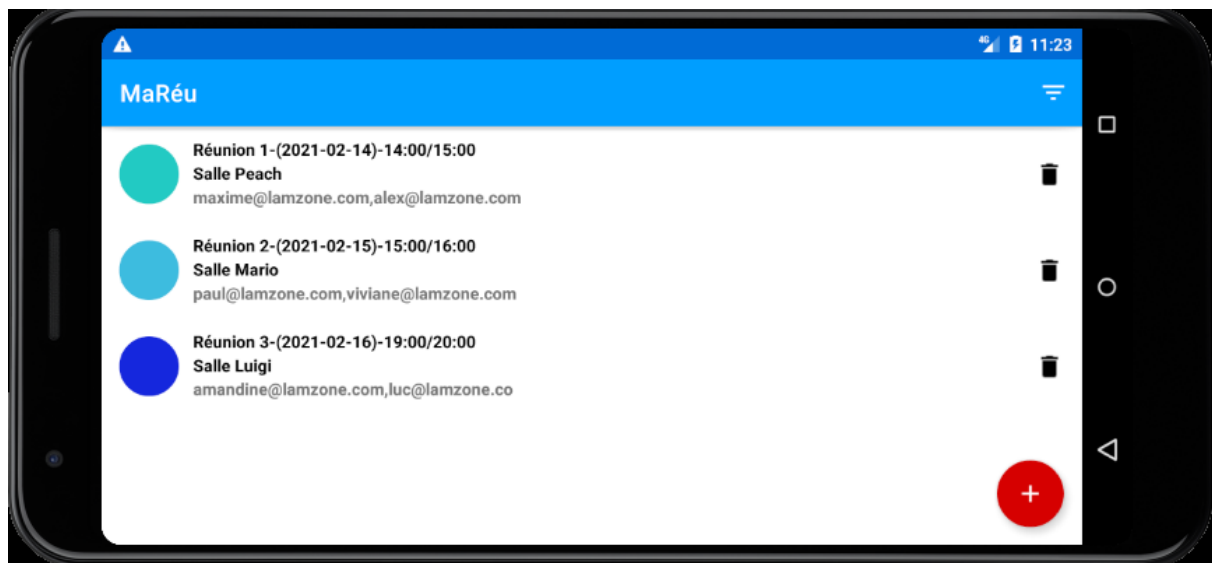
38

## Fonctionnalité 8 : Gestion de plusieurs tailles d'écran

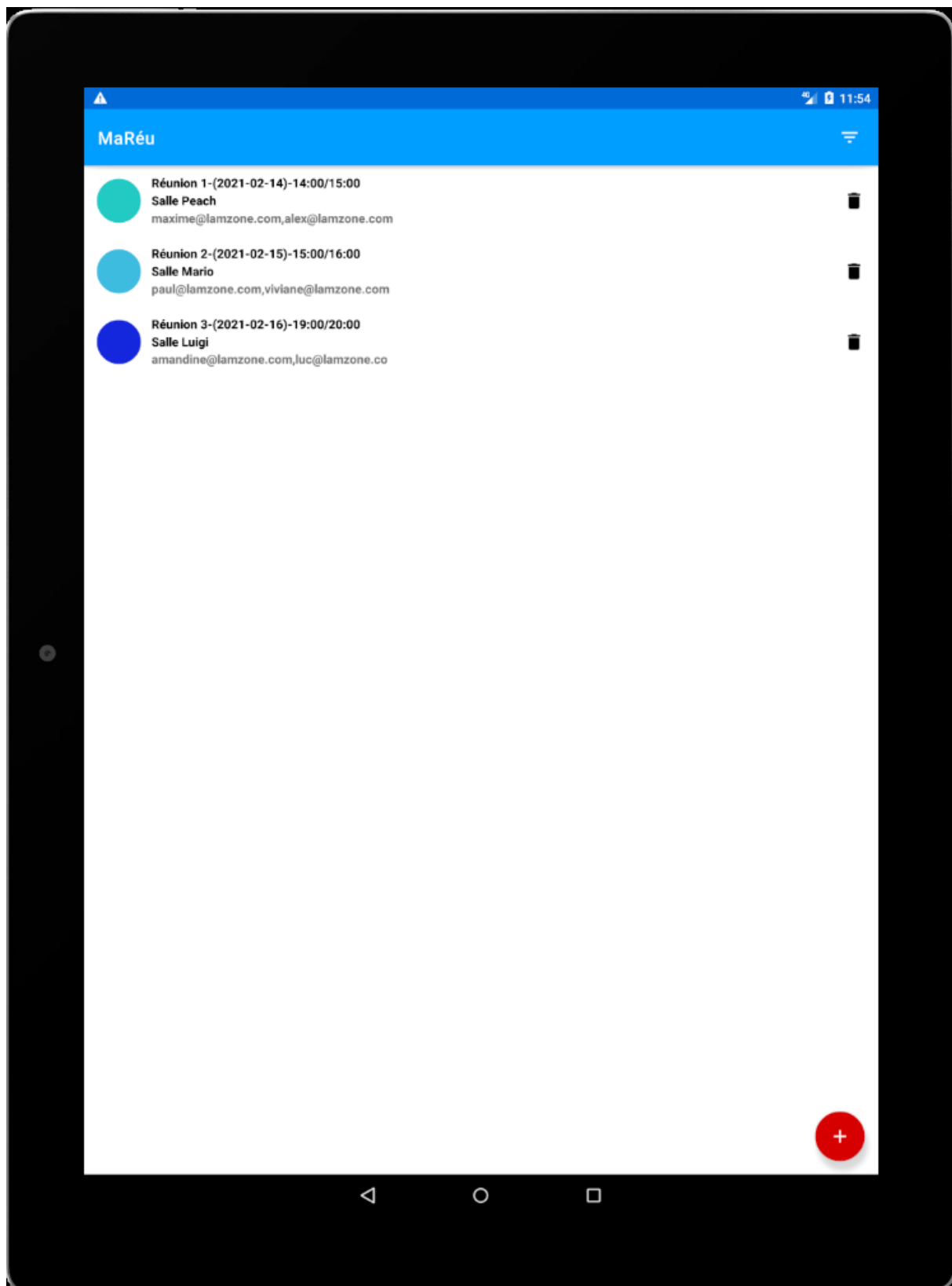


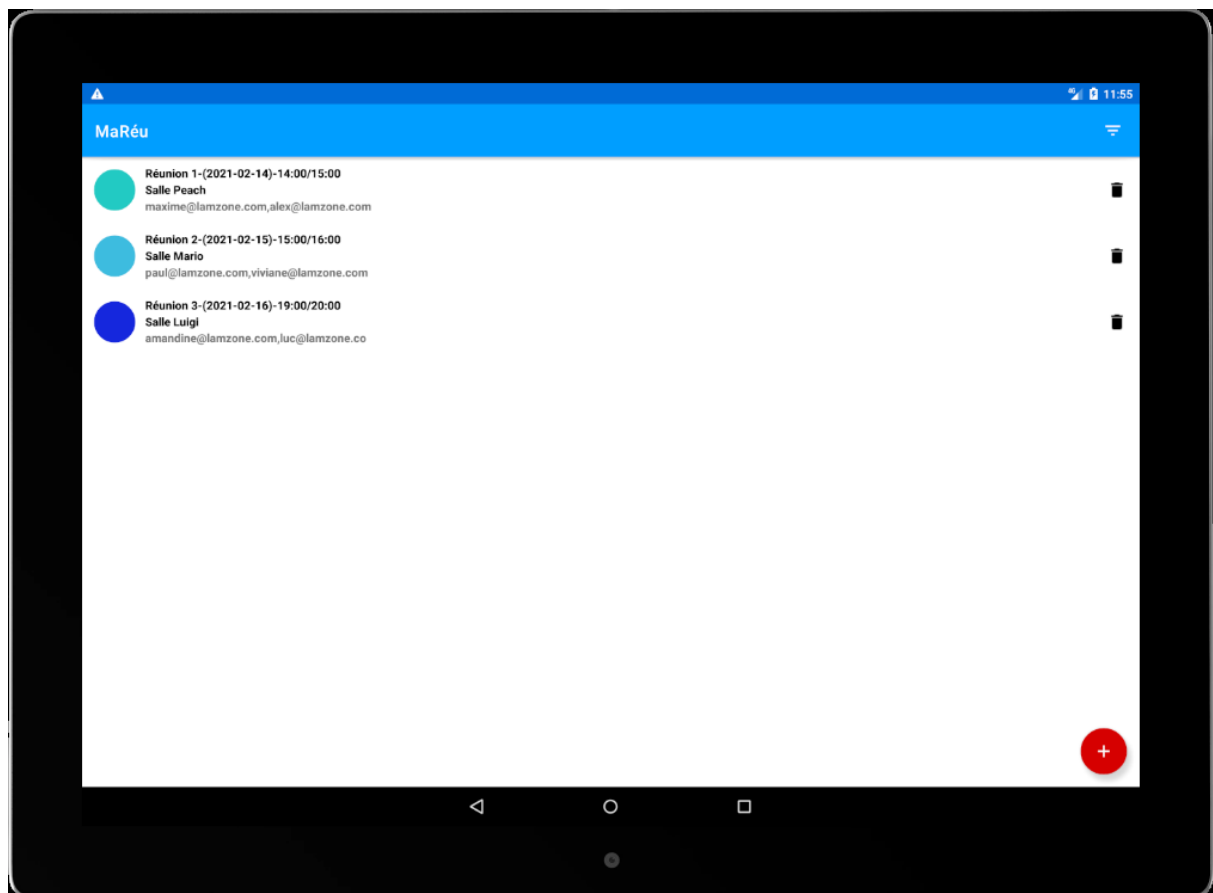












39