



**CHALLENGE
BITUM JAVA**

**CHALLENGE
BITUME 27**

REVU DE PROJET

The logo features the text "CHALLENGE BITUM JAVA" stacked above "CHALLENGE BITUME 27". Below this, the words "REVU DE PROJET" are centered. The background is a dark, circular graphic with a yellow gradient on the right side.

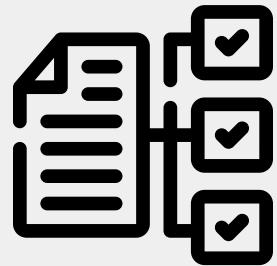
SOMMAIRE

| | | | |
|-----|------------------------|----|------------------------|
| 03 | CONTEXTE | 15 | GESTION DU CHRONOMÈTRE |
| 04 | RÉPARTITION DES TÂCHES | 16 | GESTION DU CLASSEMENT |
| 05 | CAHIER DES CHARGES | 17 | CLIENT LOURD |
| 06 | TECHNOLOGIES | 18 | CAHIER DES CHARGES |
| 7-9 | LES DIAGRAMMES | 19 | CRÉATION D'UN PDF |
| 10 | ARCHITECTURE | 20 | VISUALISATION |
| 12 | PARTIE MVC | 21 | BUG |
| 13 | GESTION WEBCAM | 22 | CONCLUSION |
| 14 | CROQUIS | | |

CONTEXTE

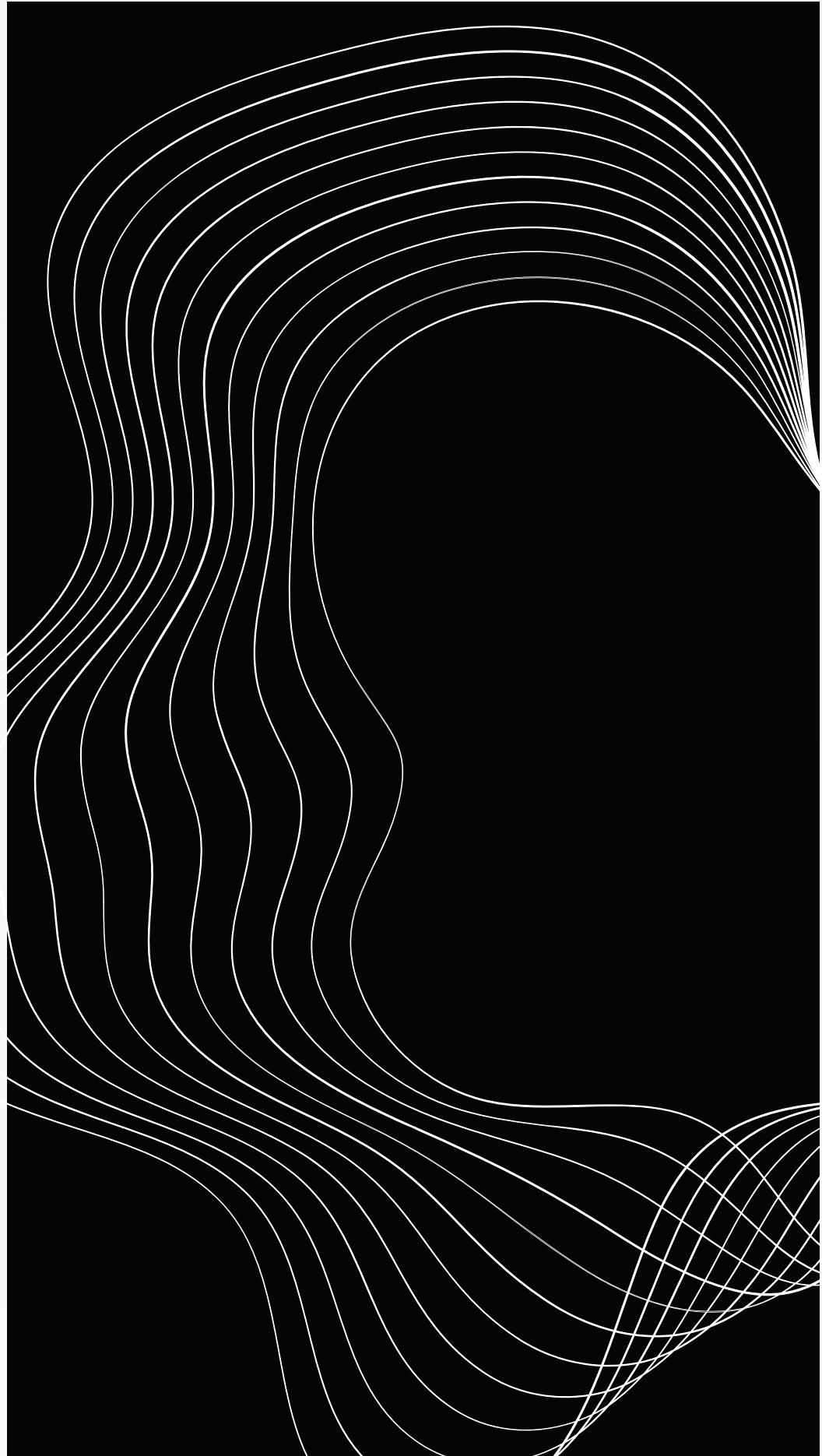


Le challenge **Bitume27** est une **association** de coureurs qui organise un mini championnat de course à pied dans le département de l'Eure chaque année.



Le but de ce projet est de :

- gérer le classement.
- faciliter la gestion des courses.
- La reconnaissance automatique des coureurs.

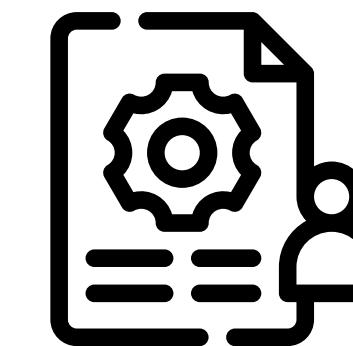


RÉPARTITION DES TÂCHES

| | Taches à réaliser |
|------------|---|
| Étudiant 1 | <ul style="list-style-type: none">• Mise en place de la couche métier (entité et service généraux) ;• Gestion de la base de données (Technologie JPA avec utilisation des Lib de la section) ; |
| Étudiant 2 | <ul style="list-style-type: none">• Réalisation de l'application WEB (Technologie JSF – PrimeFace) :• Frontend client (s'inscrire, consulter le calendrier, les résultats...) ;• Backend (gérer les courses, les coureurs avec dossards, etc..) . |
| Étudiant 3 | <ul style="list-style-type: none">• Mise en place d'un serveur RESTFull pour donner l'accès aux services à distance de manière sécurisé (HTTPS / TOKEN / AUTHENTICATION) ;• Mise en place de l'API coté client. |
| Étudiant 4 | <ul style="list-style-type: none">• Réalisation de l'application client lourd de chronométrage ;• Gestion WebCAM ;• Reconnaissance QRCode. |

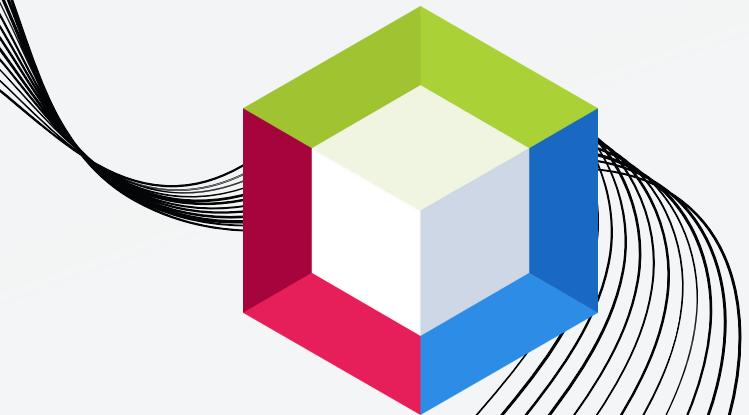
CAHIER DES CHARGES

4 sous-systèmes :



- Visualiser la course et scanner les QR codes (coureurs) à l'arriver
- Créer/supprimer un classement et le visualiser
- Chronométrier la course à son lancement

TECHNOLOGIES



Maven™



JPA
Java Persistence API



DIAGRAMME UML

Les cas d'utilisation

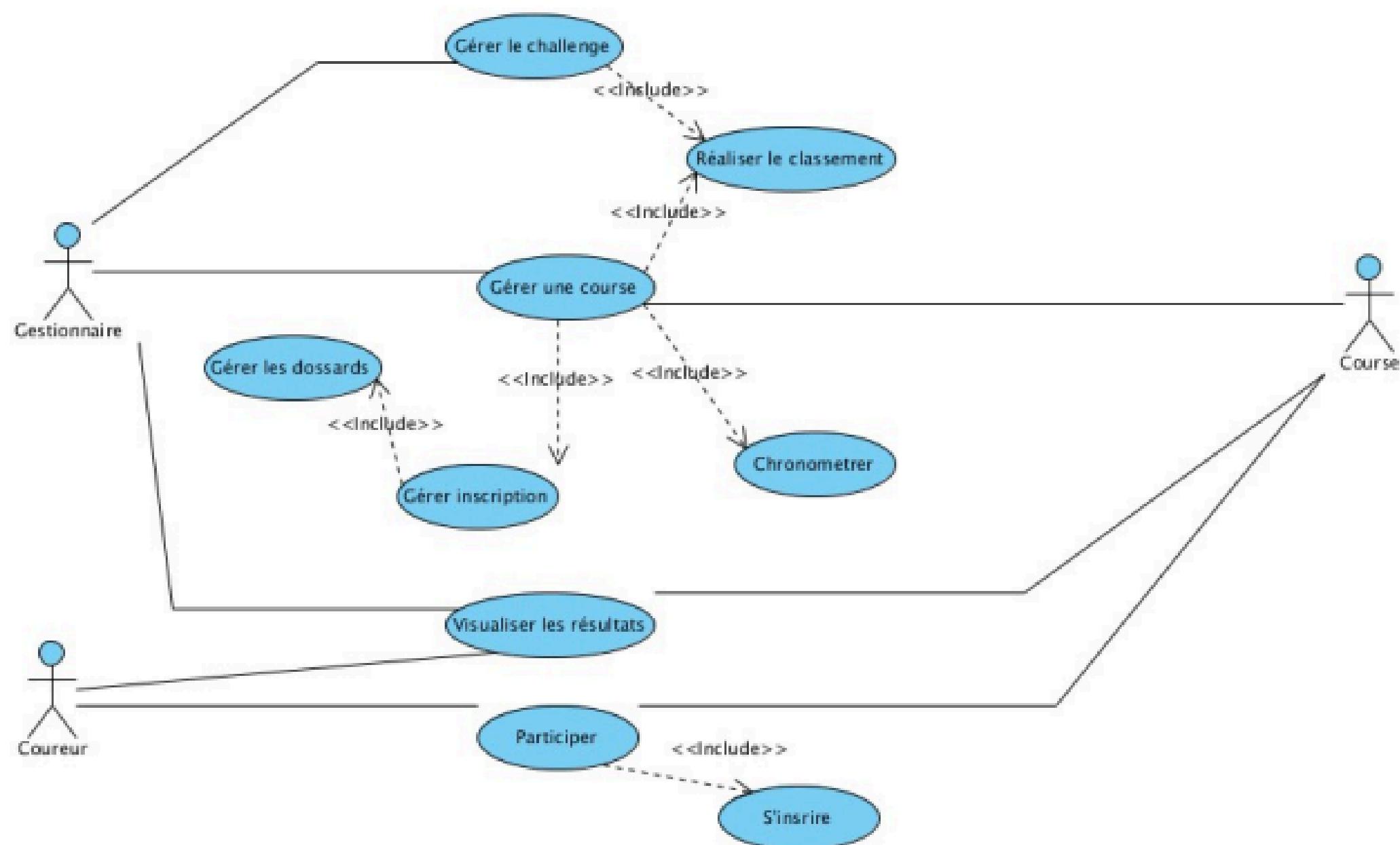


DIAGRAMME UML

classe

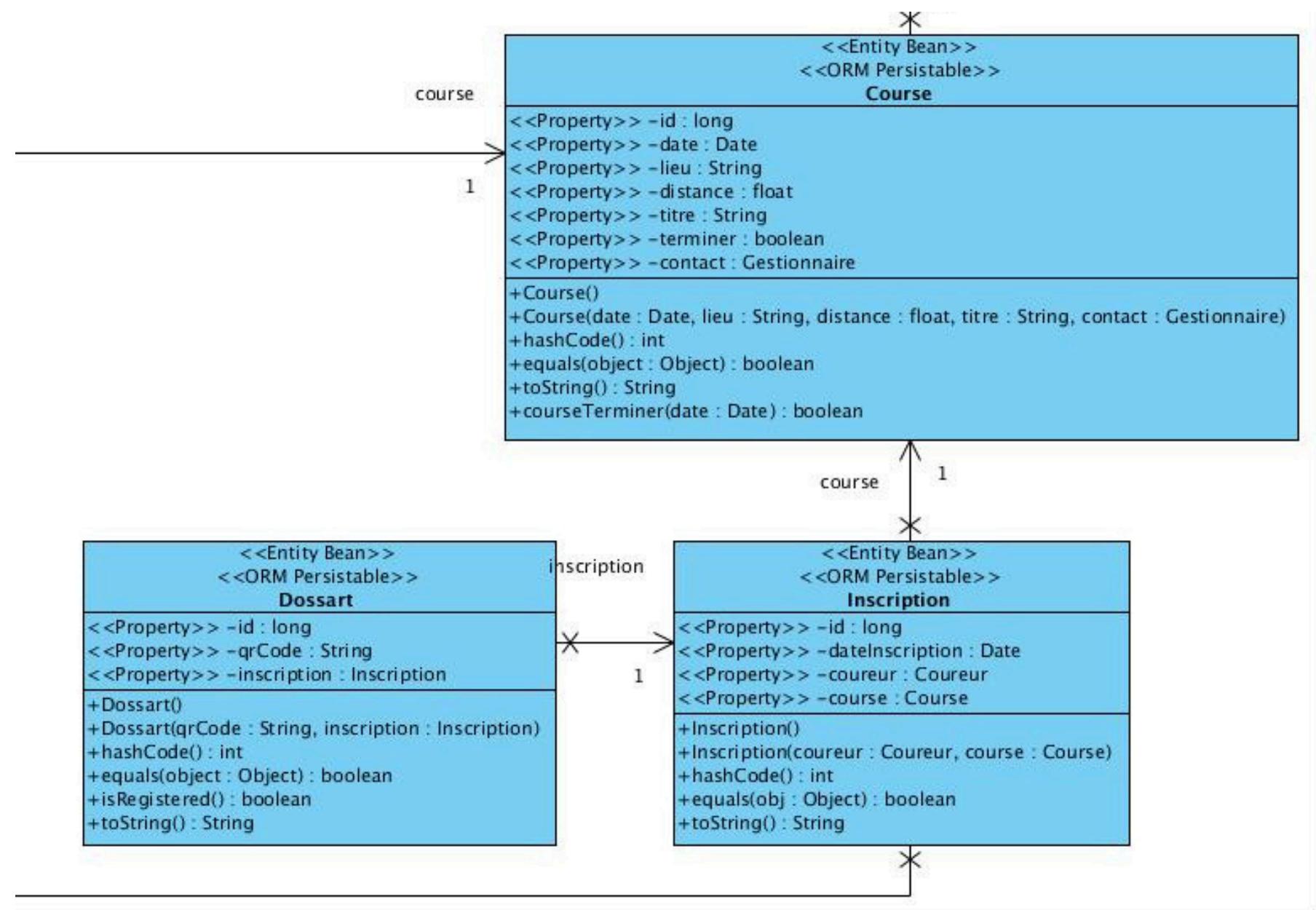
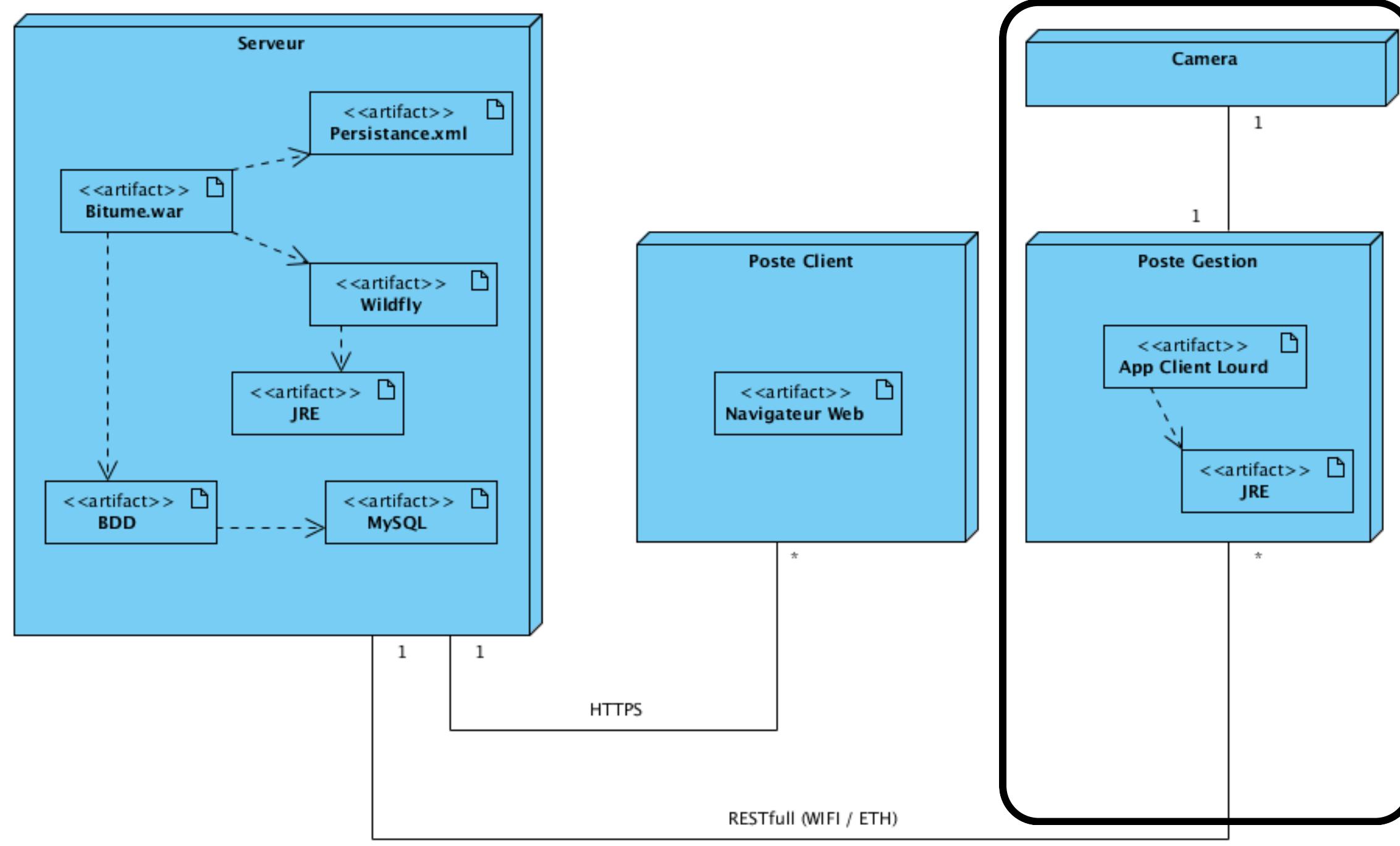
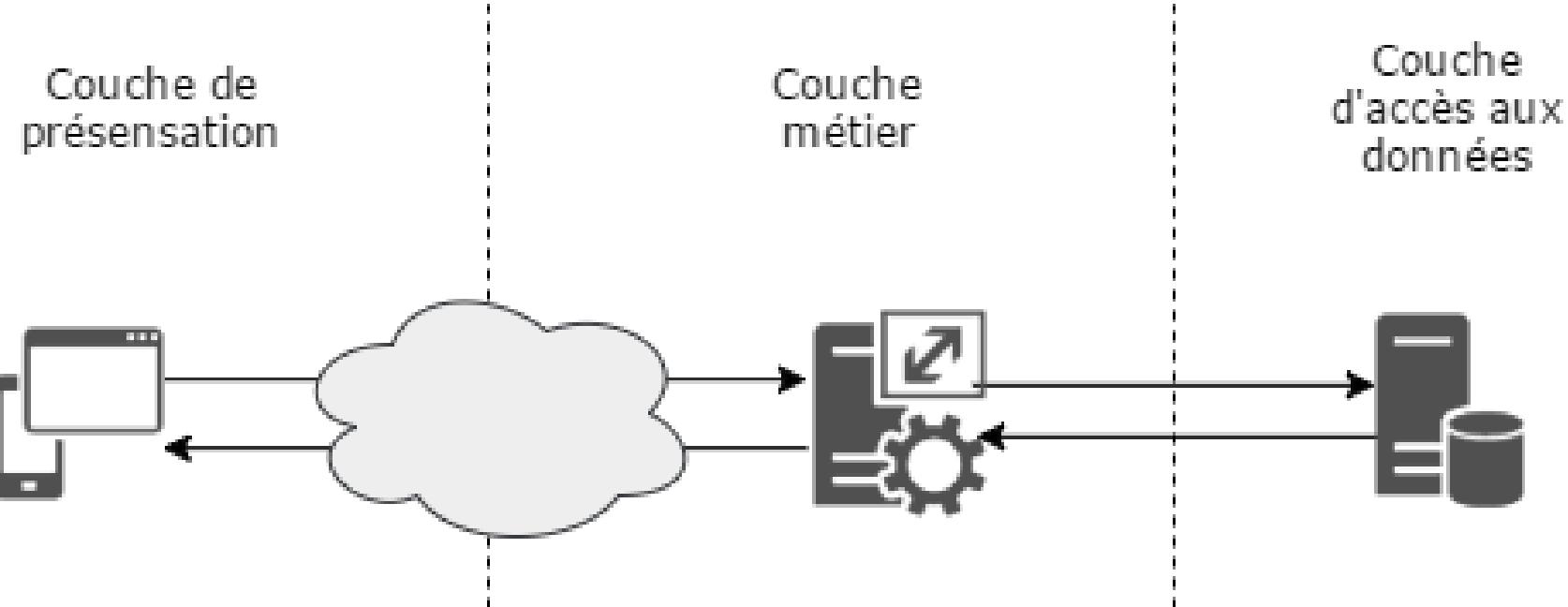


DIAGRAMME UML

Déploiement



Architecture en N-Tiers



MVC

```
lml.snir.ChallengeBitume27.App  
  ClientCamera.java [R/M]  
  Coureur.java [R/M]  
  ModeleDynamique.java [R/M]
```

Couche métier

```
lml.snir.ChallengeBitume27.metier.transactionnel  
  CategorieService.java  
  CategorieServiceImpl.java  
  ChampionnatService.java  
  ChampionnatServiceImpl.java  
  CoureurService.java  
  CoureurServiceImpl.java  
  CourseService.java  
  CourseServiceImpl.java  
  DossartService.java  
  DossartServiceImpl.java  
  InscriptionService.java  
  InscriptionServiceImpl.java  
  ResultatService.java  
  ResultatServiceImpl.java
```

Couche physique

```
lml.snir.ChallengeBitume27.physique.data  
  CategorieDataService.java  
  CategorieDataServiceJPAImpl.java  
  ChampionnatDataService.java  
  ChampionnatDataServiceJPAImpl.java  
  CoureurDataService.java  
  CoureurDataServiceJPAImpl.java  
  CourseDataService.java  
  CourseDataServiceJPAImpl.java  
  DossartDataService.java  
  DossartDataServiceJPAImpl.java  
  InscriptionDataService.java  
  InscriptionDataServiceJPAImpl.java  
  PhysiqueDataFactory.java  
  ResultatDataService.java  
  ResultatDataServiceJPAImpl.java
```



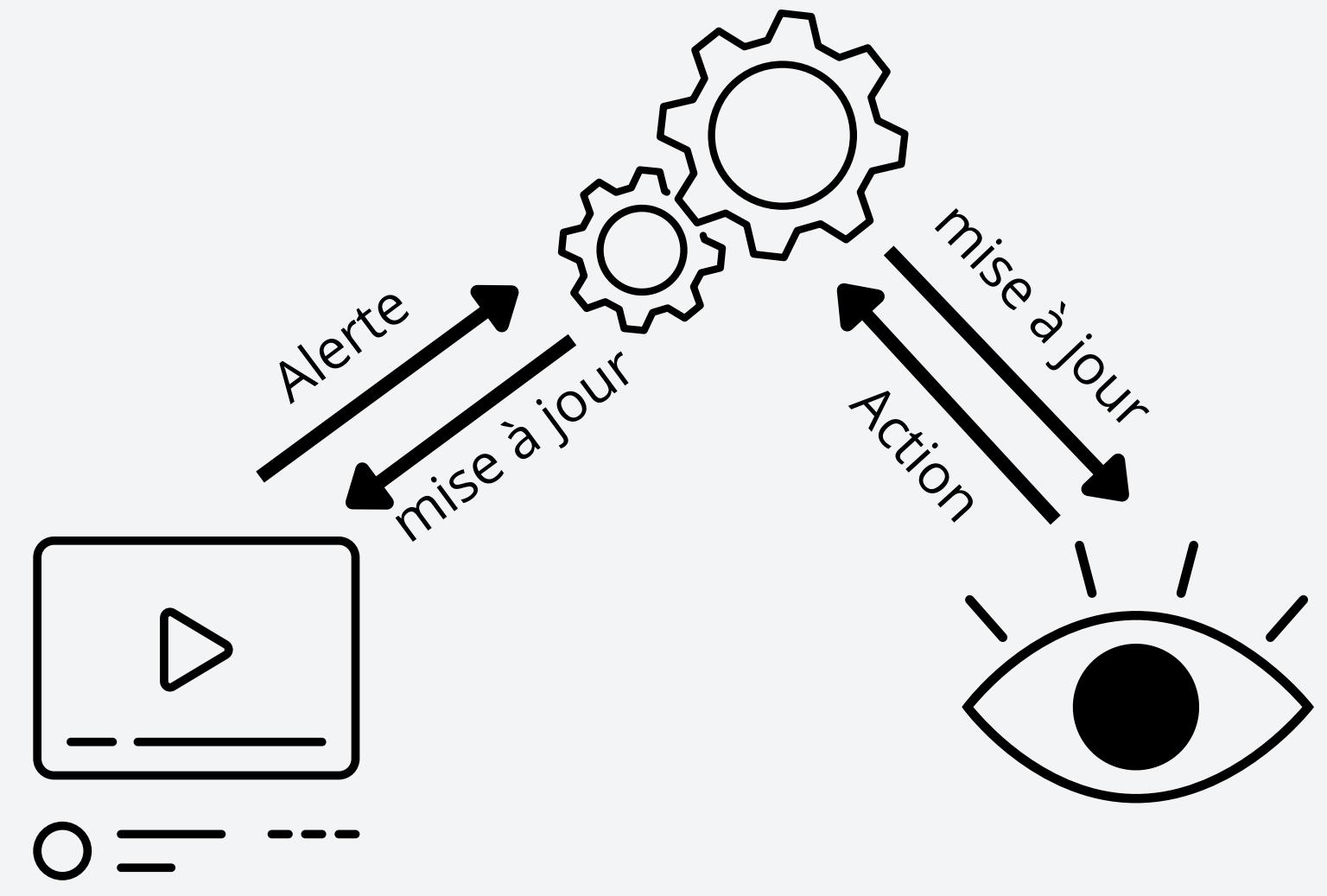
CHALLENGE
BITUME 27

MA PARTIE DU
CHALLENGE

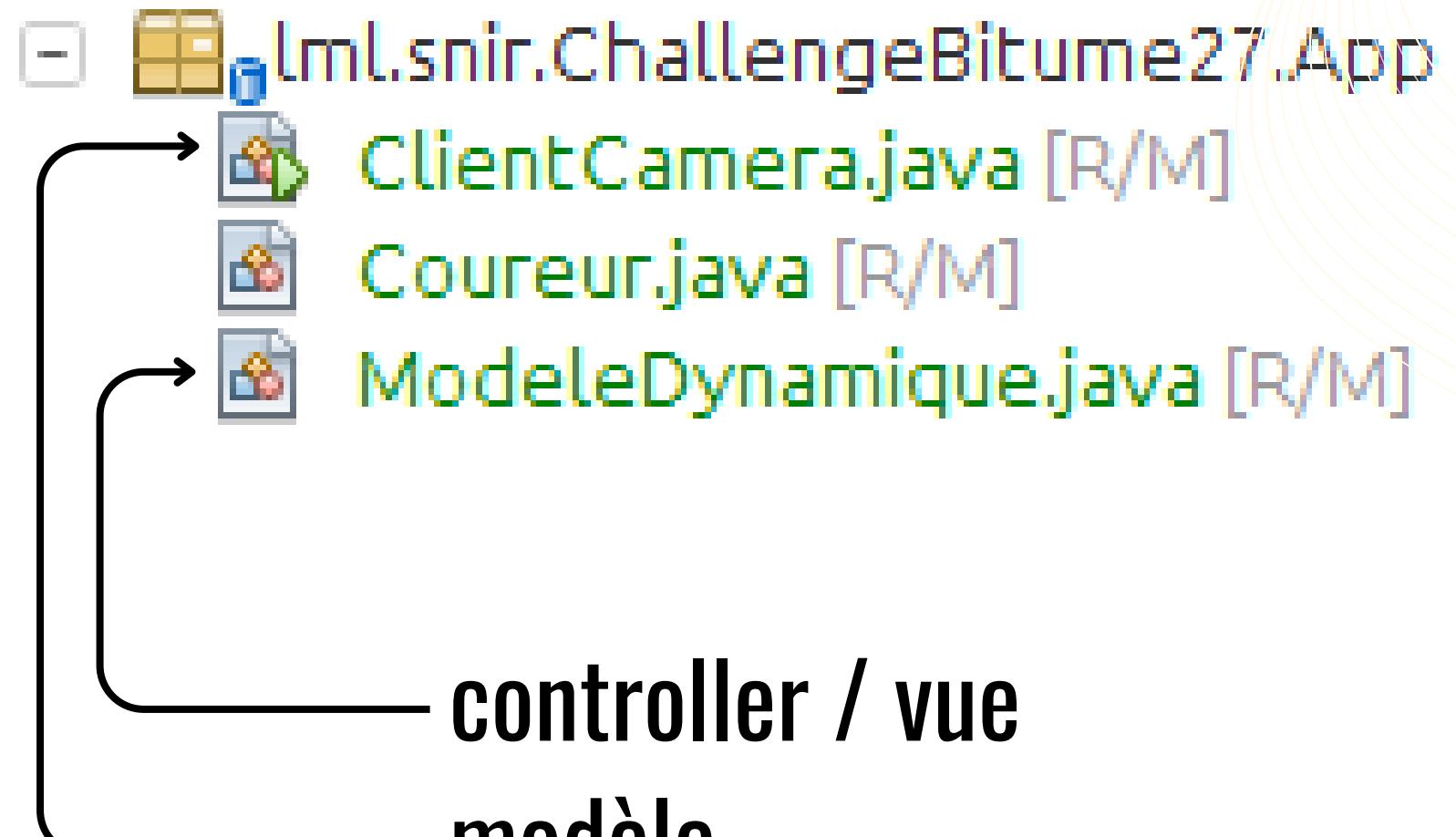
BITUME 27

Partie MVC

modèle/vue/controller

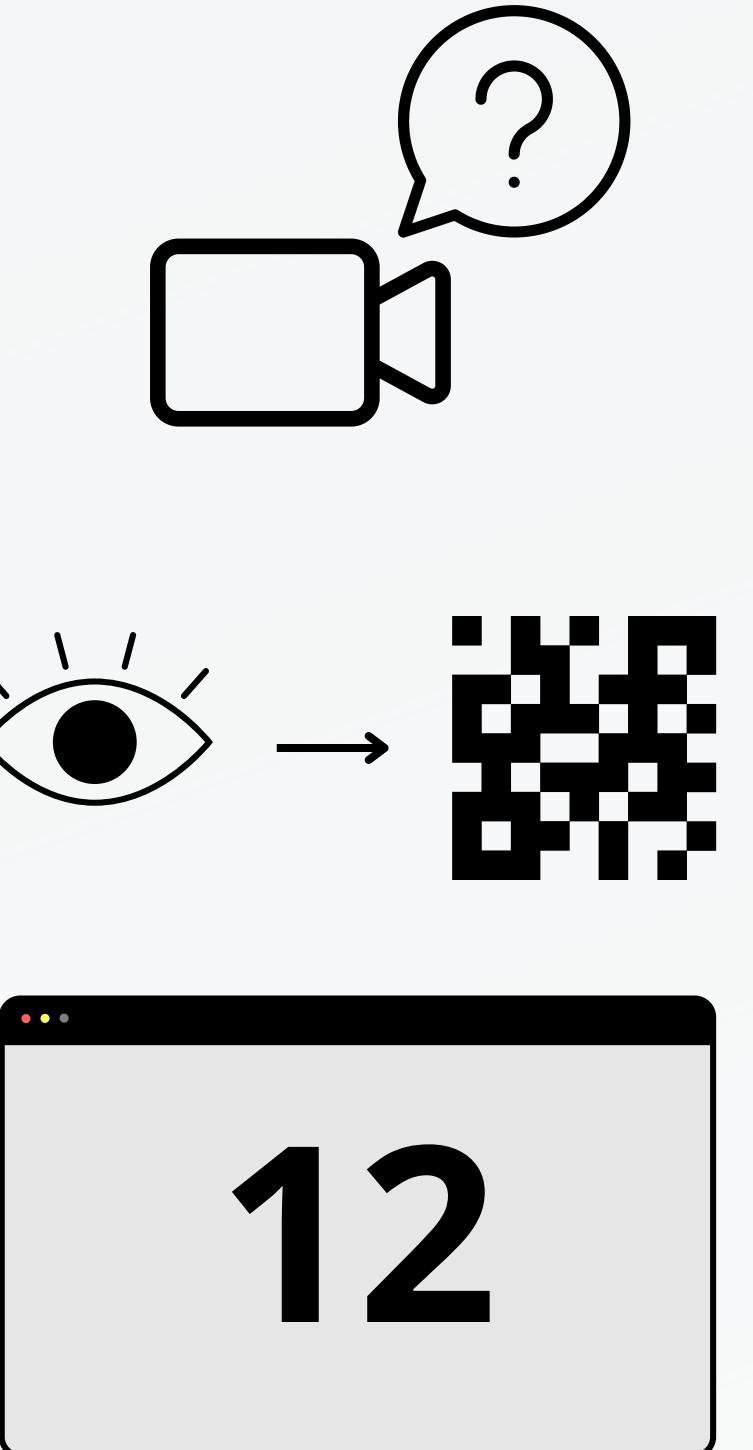


Application



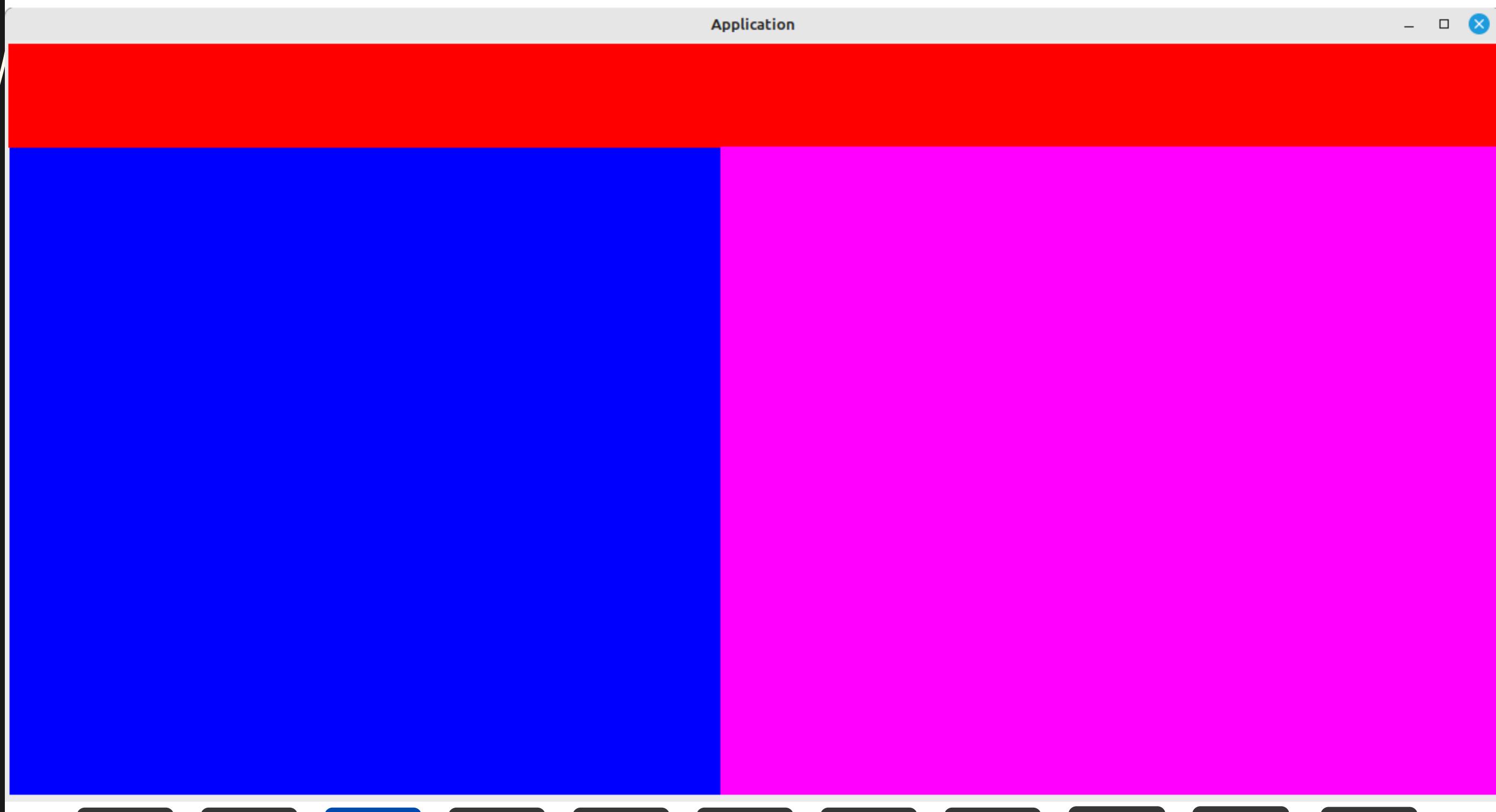
GESTION DE WEBCAM

```
@Override  
public void run() {  
  
    do {  
        try {  
            Thread.sleep( millis: 100 );  
        } catch ( InterruptedException e ) {  
            e.printStackTrace();  
        }  
  
        Result result = null;  
        BufferedImage image = null;  
  
        if ( webcam.isOpen() ) {  
            if ((image = webcam.getImage()) == null) {  
                continue;  
            }  
  
            LuminanceSource source = new BufferedImageLuminanceSource( image );  
            BinaryBitmap bitmap = new BinaryBitmap( new HybridBinarizer( source ) );  
  
            try {  
                result = new MultiFormatReader().decode( image: bitmap );  
            } catch ( NotFoundException e ) {  
                // fall thru, it means there is no QR code in image  
            }  
        }  
  
        if ( result != null ) {  
            textarea.setText( t: result.getText() );  
        }  
    } while ( true );  
}
```



CLIENT LOURD

CROQUIS



GESTION DU CHRONOMÈTRE

```
public void reset() {
    if (stop) {
        millisec = 0;
        sec = 0;
        min = 0;
        hours = 0;
        Timer t = new Timer();
        temps.setText("00 h : 00 m : 00 s : 000 ms");
    }
}

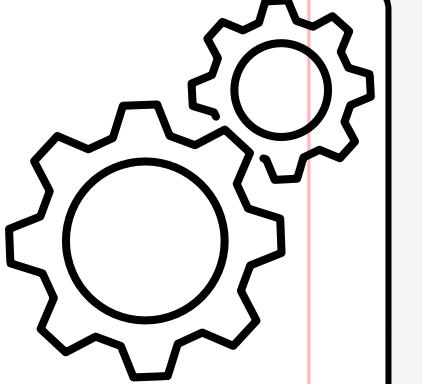
private String convertTimeToString() {

    if (millisec > 999) {
        this.millisec = 0;
        sec++;
    }

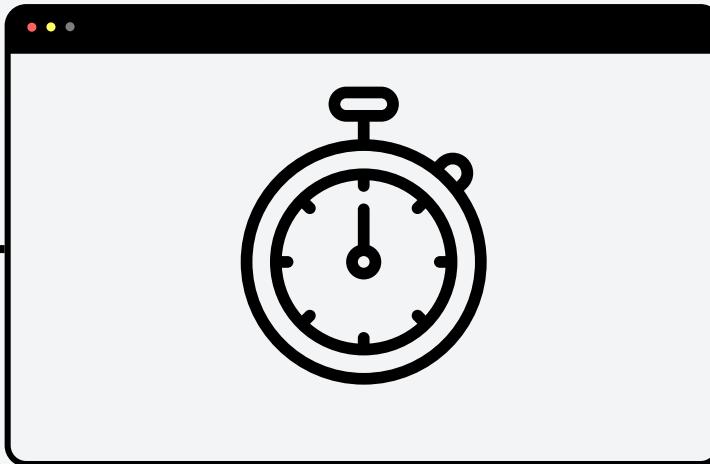
    if (sec == 60) {
        this.sec = 0;
        min++;
    }

    if (min == 60) {
        this.min = 0;
        hours++;
    }

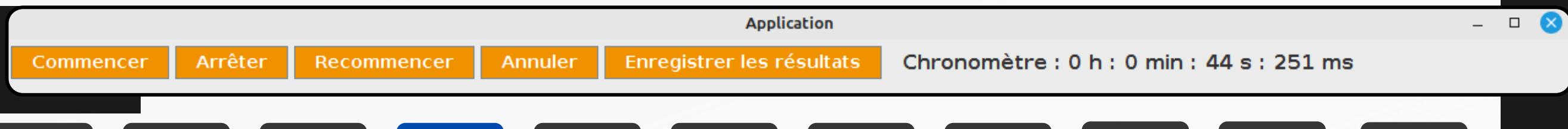
    return hours + " h : " + min + " min : " + sec + " s : " + millisec + " ms";
}
```



```
chrono = new JLabel( string: " Chronomètre :");
temps = new JLabel( string: "00 h : 00 m : 00 s : 000 ms");
```

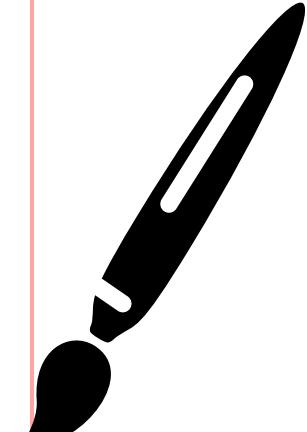
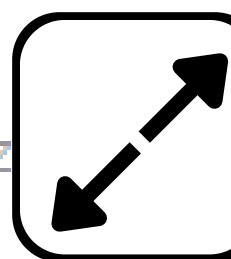


```
@Override
public void setVisible(boolean b) {
    super.setVisible(b);
    Timer t = new Timer();
    t.schedule(new TimerTask() {
        @Override
        public void run() {
            if (!stop) {
                millisec++;
                SwingUtilities.invokeLater(() -> {
                    temps.setText( text: convertTimeToString());
                });
            }
        }
    }, delay: 0, period: 1);
}
```



GESTION DU CLASSEMENT

```
//design table
table.setEnabled( enabled: true);
table.setBackground( bg: Color.white);
table.setOpaque( isOpaque: true);
table.setShowVerticalLines( showVerticalLines: false);


|                                                                                     |                                                                                       |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|  |  |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|

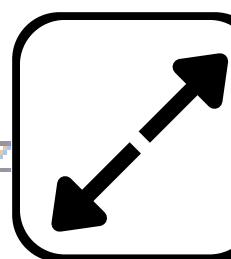
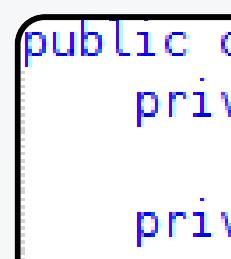


```
table.getTableHeader().setBackground(new Color(i: 240, il: 145, i2: 0));
table.getTableHeader().setForeground(fg: Color.white);
table.getTableHeader().setFont(new java.awt.Font(string: "SansSerif", i: 1, il: 14));
table.setRowHeight(rowHeight: 20);

JScrollPane scroll = new JScrollPane(cmpnt: table,
 i: JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED,
 il: JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);

JScrollBar bar = scroll.getVerticalScrollBar();
bar.setPreferredSize(new Dimension(i: 15, il: 0));
scroll.setPreferredSize(new Dimension(i: 600, il: 550)); //table size
```



|                                                                                       |                                                                                      |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|  |  |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|



```
table.setFillViewport(fillsViewportHeight: true);
```


```

```
public class ModeleDynamique extends AbstractTableModel {
    private final List<Courreur> amis = new ArrayList<Courreur>();

    private final String[] entetes = {"Top", "Temps", "Nom", "Prénom", "QR"};
```

CLIENT LOURD

Application

Commencer Arrêter Recommencer Annuler Enregistrer les résultats Chronomètre : 0 h : 0 min : 44 s : 251 ms

| Top | Temps | Nom | Prénom | QR |
|-----|-----------------------------|--------|----------|----|
| 1 | 0 h : 0 min : 11 s : 719 ms | Aire | Mélanie | 3 |
| 2 | 0 h : 0 min : 14 s : 6 ms | Faivre | Clément | 5 |
| 3 | 0 h : 0 min : 16 s : 250 ms | Lucas | Morgan | 20 |
| 4 | 0 h : 0 min : 17 s : 383 ms | Lucas | Morgan | 13 |
| 5 | 0 h : 0 min : 17 s : 731 ms | Lucas | Morgan | 6 |
| 6 | 0 h : 0 min : 19 s : 831 ms | Duret | Romain | 21 |
| 7 | 0 h : 0 min : 21 s : 769 ms | Alonso | Audrey | 9 |
| 8 | 0 h : 0 min : 32 s : 352 ms | Alonso | Stéphane | 8 |

5km, le Tue Jun 04

5km, le Tue Jun 04

of 10 courses << < 1 2 3 4 > >>

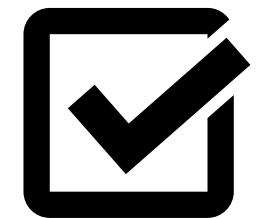
Les 3 meilleurs résultats :

FPS: 28,6

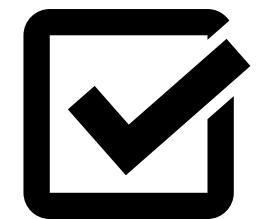
CAHIER DES CHARGES



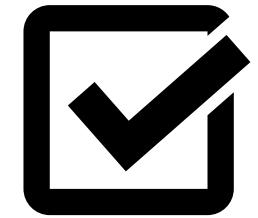
-Visualiser la course et scanner les QR codes (coureurs) à l'arriver



-Créer/supprimer un classement et le visualiser



-Chronométrer la course à son lancement



Enregister les résultats en PDF

CRÉATION DU PDF

```
// Création d'un pdf avec les résultats
public void createPdf() {
    System.out.println("Méthode à faire");

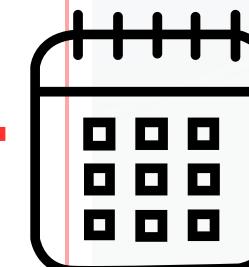
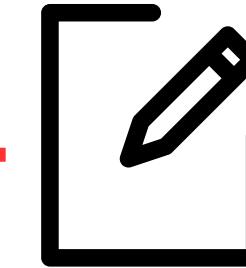
    String dest = "/home/jupiter/" + nomPdf + ".pdf";
    PdfWriter writer = null;

    try {
        writer = new PdfWriter(filename: dest);
    } catch (Exception e) {
        e.printStackTrace();
    }

    PdfDocument pdfDocument = new PdfDocument(writer);
    pdfDocument.addNewPage();
    Document document = new Document(pdfDocument);

    SimpleDateFormat sdf = new SimpleDateFormat(string: "dd/MM/yyyy");
    document.add(new Paragraph(text: sdf.format(new Date())));

    float[] pointColumnWidths = {30F, 250F, 200F, 200F, 30F};
    Table table = new Table(pointColumnWidths);
```



VISUALISATION

Vignettes ▾ X

04/06/2024

| CLT | Temps | NOM | Prénom | Numéro |
|-----|-----------------------------|--------|----------|--------|
| 1 | 0 h : 0 min : 5 s : 409 ms | Alonso | Stéphane | 1 |
| 2 | 0 h : 0 min : 5 s : 650 ms | Alonso | Stéphane | 8 |
| 3 | 0 h : 0 min : 10 s : 977 ms | Alonso | Stéphane | 15 |
| 4 | 0 h : 0 min : 18 s : 164 ms | Alonso | Audrey | 2 |
| 5 | 0 h : 0 min : 23 s : 495 ms | Aire | Mélanie | 3 |
| 6 | 0 h : 0 min : 28 s : 501 ms | Lucas | Morgan | 6 |

BUG

Effacer deux fois

Application

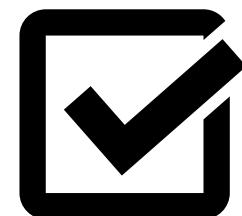
Commencer Arrêter Recommencer Annuler Enregistrer les résultats Chronométrer

| Top | Temps | Nom | Prénom | QR |
|-----|-----------------------------|--------|----------|----|
| 1 | 0 h : 0 min : 11 s : 719 ms | Aire | Mélanie | 3 |
| 2 | 0 h : 0 min : 14 s : 6 ms | Faivre | Clément | 5 |
| 3 | 0 h : 0 min : 16 s : 250 ms | Lucas | Morgan | 20 |
| 4 | 0 h : 0 min : 17 s : 383 ms | Lucas | Morgan | 13 |
| 5 | 0 h : 0 min : 17 s : 731 ms | Lucas | Morgan | 6 |
| 6 | 0 h : 0 min : 19 s : 831 ms | Duret | Romain | 21 |
| 7 | 0 h : 0 min : 21 s : 769 ms | Alonso | Audrey | 9 |
| 8 | 0 h : 0 min : 32 s : 352 ms | Alonso | Stéphane | 8 |

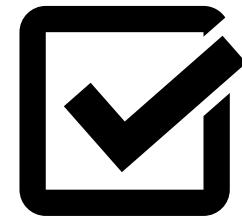


```
public void deleteEveryRowInTable(){
    int last = courreurs.size();
    courreurs.removeAll(courreurs);
    fireTableRowsDeleted(firstRow: 0, lastRow: last);
}
```

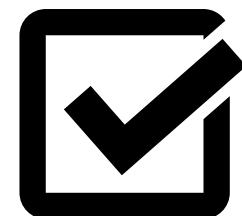
CE QUE J'AI AJOUTÉ



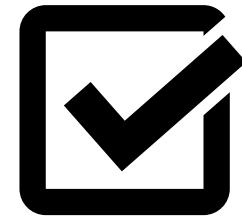
-Visualiser la course et scanner les QR codes (coureurs) à l'arriver



-Créer/supprimer un classement et le visualiser



-Chronométrer la course à son lancement



-Enregistrer les résultats en PDF

**MERCI DE
M'AVOIR
ÉCOUTÉ**

Avez vous des question ?

