

# Vélo connecté Clean-Bike

Yohann - Robin - Yannick



# Sommaire

1. Les Équipements intégrés
2. Le type de connectivité
3. Pourquoi nous avons choisi de faire une application mobile
4. Les fonctionnalités de Clean-Bike
5. La présentation technique
  - a. Principes de développement
  - b. Diagramme de classes
  - c. Exemples de tests

# Équipements intégrés



Capteur de cadence Garmin

Caractéristique	Garmin Cadence 2	Wahoo Fitness RPM
Marque	Garmin	Wahoo Fitness
Prix	39.99	40.00
Connectivité	ANT+ , Bluetooth	Bluetooth, ANT+
Compatibilité	iOS, Android	iOS, Android
Poids	6 grammes	7 grammes
Étanchéité	IPX7	IPX7
Type de fixation	Aimant	Sans aimant
Applications compatibles	Garmin Connect, Zwift, strava	Wahoo Fitness, Strava

# Équipements intégrés



Tablette Android

- Application Clean-Bike
- Récupère les données du capteur
- Affiche la vidéo en fonction de la vitesse de l'utilisateur

# Connectivité



- Bluetooth multipoint
- Connexion entre tablette et équipements (capteur, montre connectée)

# Pourquoi une application Mobile

- Facilité d'installation pour les utilisateurs
- Utilisable partout et sans connexion
- Peut être utilisé avec un simple smartphone ou une tablette Android
- Tous les smartphones récents peuvent se connecter à des capteurs et récupérer les données envoyées par ceux-ci



# Fonctionnalités de l'application

- Système de connexion et déconnexion
- Choix et réalisation d'un parcours
- Accéder à l'historique des parcours réalisés
- Fonctionnement avec une montre connectée
- Avoir un coach permettant de diriger l'entraînement

# Différents parcours

Différents parcours sont disponibles pour utiliser votre vélo aux 4 coins du monde

Le parcours est une vidéo dont la vitesse s'adapte en fonction de votre vitesse de pédalage

0 km/h -> Pause

20 km/h -> X1

40 km/h -> X2



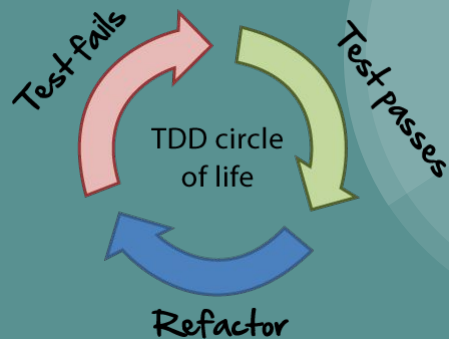
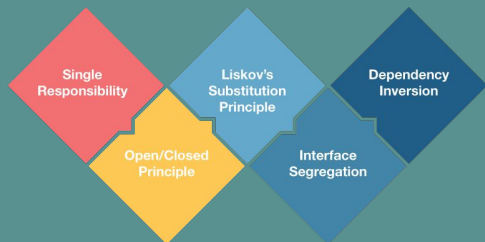


# Présentation technique



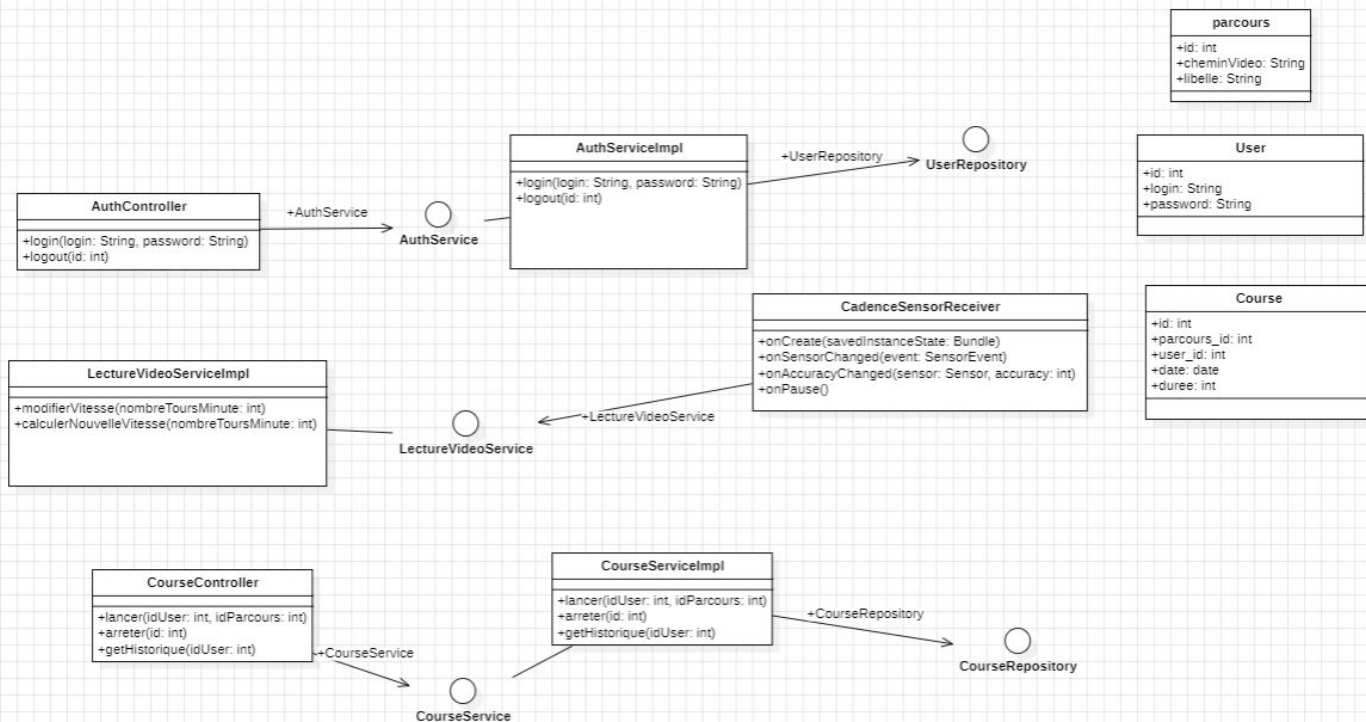
# Principes de développement

## S.O.L.I.D.



**KEEP  
IT  
SIMPLE,  
STUPID**

# Diagramme de classes



# Test d'ajout d'un parcours

```
@SpringBootTest
class ParcoursRepositoryTest {
    3 usages
    @Autowired
    private ParcoursRepository parcoursRepository;

    @BeforeEach
    void deleteAllParcours() { parcoursRepository.deleteAll(); }

    @Test
    @Order(1)
    void shouldAjouterParcours(){
        parcoursRepository.save(new Parcours( id: 1L, cheminVideo: "/test", libelle: "Course 1"));
        assertEquals( expected: 1L, parcoursRepository.count());
    }
}
```

# Test d'ajout d'une course à un utilisateur

```
@SpringBootTest
class CourseRepositoryTest {
    5 usages
    @Autowired
    private CourseRepository courseRepository;

    @BeforeEach
    void deleteAllCourse() { courseRepository.deleteAll(); }

    @Test
    @Order(1)
    void shouldReturnCoursesByUserId(){
        Long userId = 123L;
        courseRepository.save(new Course( id: 1L, parcoursId: 1L, userId, new Date()));
        courseRepository.save(new Course( id: 2L, parcoursId: 3L, userId, new Date()));
        courseRepository.save(new Course( id: 3L, parcoursId: 2L, userId, new Date()));
        List<Course> courses = courseRepository.findAllByUserId(userId);
        assertEquals( expected: 3, courses.size());
    }
}
```