|  |  |
| --- | --- |
| **Bloc de compétences 2 SLAM – Conception et développement d’applications Semestre 2** | |
| **Finalité métier :** Vous exercez les activités suivantes sous la responsabilité d’un chef de projet pour répondre aux besoins de digitalisation de vos clients :   * Recueil et analyse des besoins issus des utilisateurs et/ou des métiers * Conception de la solution applicative * Gestion des données * Implémentation de la solution applicative avec les technologies retenues * Conception et réalisation de plans de tests * Rédaction de la documentation technique * Recette de l’application et accompagnement des utilisateurs * Gestion de la maintenance corrective et/ou évolutive de l’application | **Contexte professionnel** : Vous travaillez pour le compte d’une entreprise de services du numérique qui intervient auprès de différentes entreprises clientes pour assurer des missions de conception, de développement d’applications informatiques. Vous assurez également la maintenance corrective ou évolutive de leurs applications existantes ainsi que la gestion des données de leurs applications. Le déploiement des applications ainsi que le stockage des données peuvent être réalisés à travers une solution d’informatique en nuage (*cloud computing*). Les projets sont réalisés dans le respect d’un cahier des charges et en tenant compte des contraintes techniques, environnementales et réglementaires. Vous assurez une veille technologique régulière et appliquez les bonnes pratiques de qualité et de développement pour apporter des solutions performantes et des technologies innovantes à vos clients. Les projets de développement de ces applications digitales sont menés à l’aide de méthodes classiques de gestion de projets (cycle en V) ou bien en mode agile. |

|  |  |
| --- | --- |
| **2.1 Concevoir et développer une solution applicative Semestre 2** | |
| **Votre mission :** Vous participez, sous la responsabilité d’un chef de projet, aux différentes activités d’analyse, de conception et de développement de nouvelles applications de bureau pour vos clients dans le respect du cahier des charges en veillant à la qualité logicielle, à l’application des bonnes pratiques, à l’évolutivité et la réutilisation de composants logiciels. | **Contexte professionnel** : En qualité de développeuse ou de développeur junior dans le service « étude et développement », vous participez à différents projets de développement d’applications informatiques pour le compte de différents clients travaillant dans le secteur tertiaire. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Séquence 2.1. A2D** | **Modéliser et maquetter une solution applicative** | | | | | |
| Durée totale en heures du scénario pour la séquence  6 h | Au sein de la direction des systèmes d’information d’une entreprise cliente, votre travail consiste à participer à la conception d’une nouvelle application. À partir du cahier des charges, vous êtes responsable de l’élaboration de la cartographie fonctionnelle des besoins exprimés et de la formalisation des spécifications fonctionnelles et techniques en vous appuyant sur les méthodes, normes et standards d’analyse et de conception de solutions logicielles. | | | | | |
| **Compétences travaillées** | **Savoirs associés** | | **Indicateurs de performance** | | **Prérequis / Transversalités** |
| * Analyser un besoin exprimé et son contexte juridique. * Modéliser une solution applicative. * Participer à la conception de l’architecture d’une solution applicative. * Rédiger des documentations techniques et d’utilisation d’une solution applicative. * Exploiter les fonctionnalités d’un environnement de développement et de tests. | Savoirs technologiques   * Méthodes, normes et standards associés au processus de conception et de développement d’une solution applicative. * Architectures applicatives ; concepts de base et typologies. * Techniques et outils d’analyse et de rétro-conception. * Typologies et techniques des cycles de production d’un service et acteurs associés. * Interfaces homme-machine : principes ergonomiques, techniques de conception, d’évaluation et de validation.   Savoirs économiques, juridiques et managériaux   * Cahier des charges et ses enjeux juridiques. * Contraintes éthiques et environnementales dans la conception d’une solution applicative. * Contrat de développement (formation, exécution, inexécution) et ses clauses spécifiques. * Responsabilité civile et pénale du concepteur de solutions applicatives. | | * La proposition de la solution applicative répond au besoin exprimé dans le cahier des charges y compris dans sa dimension contractuelle :   + la modélisation de l’application est conforme aux besoins ;   + la maquette des éléments applicatifs de la solution respecte les fonctionnalités exprimées ;   + les spécifications de l’interface utilisateur répondent aux contraintes ergonomiques. | | Prérequis :  B1.2. Répondre aux incidents et aux demandes d’assistance et d’évolution - compétence : traiter des demandes concernant les applications.  B2.2. Assurer la maintenance corrective ou évolutive d’une solution applicative – séquence 2.2 A2D. |
| **Séance 1** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | L’expression des besoins des utilisateurs de la nouvelle application a été formulée dans un cahier des charges par la maîtrise d’ouvrage du client. Vous êtes en charge des tâches suivantes :   1. Analyser le cahier des charges. 2. Formaliser les spécifications de l’application dans un diagramme de cas d’utilisation en respectant la sémantique de la méthode adoptée. 3. Identifier les acteurs de l’application. 4. Dresser la liste des logiciels nécessaires à l’installation et à l’exécution de l’application. | | * Fiche des savoirs technologiques des concepts relatifs à la **modélisation fonctionnelle** des applications et des méthodes standards associées au processus de leur analyse et de leur conception. * Exemple du processus d’analyse à travers une étude de cas et un questionnaire. * Cahier des charges de la nouvelle application (*les auteurs peuvent s’appuyer ou s’inspirer d’une étude de cas).* * Outil d’analyse et de modélisation d’applications informatiques. | | * Les spécifications fonctionnelles de l’application et les acteurs concernés sont modélisés dans un diagramme de cas d’utilisation à l’aide de l’outil de modélisation et respectent la sémantique de la méthode utilisée. * Une documentation des logiciels nécessaires à l’installation et à l’exploitation de l’application est rédigée. | |
| **Séance 2** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Afin d’intégrer le client aux différentes phases du projet, le chef de projet vous demande de réaliser une maquette de l’application pour la valider auprès des utilisateurs finaux.   1. Installer un outil de maquettage. 2. Maquetter l’interface de l’application. | | * Cahier des charges de l’application (*les auteurs peuvent s’appuyer ou s’inspirer d’une étude de cas).* * Outil de maquettage fonctionnel. * Tutoriel d’installation et d’exploitation de l’outil de maquettage. | | * L’outil de maquettage est installé. * La maquette fonctionnelle de l’application est réalisée à l’aide d’un outil de maquettage et répond au cahier des charges. | |
| **Séance 3** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Le chef de projet a réparti les différentes fonctionnalités de l’application entre les membres développeurs de votre équipe.  Il vous a confié l’étude et la conception de l’une des fonctionnalités de l’application. En respectant le formalisme de la méthode adoptée :   1. Rédiger la description du cas d’utilisation en décrivant le scénario nominal ainsi que le ou les scénario(s) alternatifs et/ou d’exception éventuels. 2. Modéliser le cas d’utilisation à l’aide d’un diagramme graphique selon le formalisme de la méthode de conception adoptée (*diagramme d’activité*). | | * Fiche des savoirs technologiques des concepts relatifs à la **vue comportementale** d’une application informatique et des méthodes standards de conception et de spécification. * Exemple de modélisation graphique de la vue comportementale d’une fonctionnalité d’une application (*diagramme d’activité ou tout autre diagramme équivalent)*. * Description du cas d’utilisation. * Outil d’analyse et de modélisation d’applications informatiques. | | * La fiche descriptive du cas d’utilisation de la fonctionnalité est rédigée en respectant la sémantique de la méthode utilisée. * Le diagramme du cas d’utilisation est modélisé à l’aide de l’outil de modélisation et respecte la sémantique de la méthode d’analyse utilisée. | |
| **Séance 4** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | Le chef de projet vous fournit une archive du code-source de la structure de l’application qu’il a développé dans un langage orienté objet et vous demande d’effectuer une rétro-conception de l’application (*reverse engineering*) à l’aide de l’outil exploité par l’équipe.   1. Récupérer le code source de la structure de l’application depuis le dépôt du projet sur le serveur distant. 2. Générer le diagramme de classes correspondant au code fourni (*ou tout autre diagramme équivalent*). 3. Analyser le diagramme de classes et déduire les règles de correspondance utilisées lors de l’opération de rétro-conception (du *langage POO vers UML par exemple)*. 4. Vérifier le diagramme de classes par rapport aux documents d’analyse générés précédemment. | | * Fiche des savoirs technologiques des concepts relatifs à la **vue structurelle** d’une application informatique et des méthodes standards associées au processus de leur conception et de leur modélisation. * Exemples de diagrammes de classes avec leur code correspondant dans un langage orienté objet. * Environnement de développement intégré avec l’outil de rétro-conception installé (greffon UML). * Archive du code-source de la structure de l’application (*le code-source fourni sera* *simple et* *volontairement incomplet)*. | | * Le diagramme de classes de l’application généré de manière automatisée à l’aide de l’outil de rétro-conception est sauvegardé dans un document. * Le document de synthèse des règles de correspondance du langage de programmation vers le langage de modélisation de la méthode adoptée (*UML par exemple*) est rédigé. * Le diagramme de classes est corrigé. * Le code-source est mis à jour à partir du diagramme de classes corrigé. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Séquence 2.1. B2D** | **Participer à la conception de l’architecture logicielle d’une solution applicative** | | | | | |
| Durée totale en heures du scénario pour la séquence  5 h | Vous travaillez dans les locaux d’une entreprise cliente et vous faites partie de l’équipe « développement et innovation » dont l’objectif est de développer des applications pour répondre à de nouveaux besoins métiers.  Votre prestation, organisée en plusieurs missions et supervisée par un chef de projet interne, consiste à créer une de ces applications. Les spécifications fonctionnelles, transmises par la direction métier et formalisées dans un cahier des charges, seront implémentées dans un langage de programmation orienté objet dans le respect des choix techniques et structurels transmis par le chef de projet. | | | | | |
| **Compétences travaillées** | **Savoirs associés** | | **Indicateurs de performance** | | **Prérequis / Transversalités** |
| * Analyser un besoin exprimé et son contexte juridique. * Participer à la conception de l’architecture d’une solution applicative. * Identifier, développer, utiliser ou adapter des composants logiciels. * Intégrer en continu les versions d’une solution applicative. * Réaliser les tests nécessaires à la validation ou à la mise en production d’éléments adaptés ou développés. * Exploiter les fonctionnalités d’un environnement de développement et de tests. | Savoirs technologiques   * Méthodes, normes et standards associés au processus de conception et de développement d’une solution applicative * Architectures applicatives : concepts de base et typologies. * Concepts de la programmation objet : classe, objet, abstraction, interface, héritage, polymorphisme, annotations, patrons de conception, interface de programmation d’applications. * Concepts de la programmation événementielle : techniques de gestion des événements et exploitation de bibliothèques de composants graphiques. * Techniques de gestion de versions. * Techniques et outils de test et d’intégration de composants logiciels. | | * La solution est développée dans les règles de l’art : * le développement répond à l’expression des besoins fonctionnels et respecte les contraintes techniques figurant dans le cahier des charges ; * un outil collaboratif de gestion des itérations de développement et de versions est utilisé. * L’application développée est opérationnelle conformément au cahier des charges et stable dans l’environnement de production. | | Prérequis :  B2.1. Séquence 2.1 A2D  B2.2. Assurer la maintenance corrective ou évolutive d’une solution applicative – séquence 2.2 A2D. |
| **Séance 1** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | Le chef de projet vous fournit le dossier documentaire de l’application.  Votre tâche consiste à concevoir l’architecture modèle-vue-contrôleur (MVC) de l’application en respectant les choix structurels définis par le chef de projet.   1. Étudier le dossier documentaire relatif au projet. 2. Récupérer les classes utilitaires développées par l’équipe en interne. 3. Concevoir le diagramme de paquetages UML de l’application. 4. Concevoir le diagramme de classes de l’application. | | * Fiche des savoirs technologiques sur les concepts de base de la programmation orientée objet (POO – *classe, objet, abstraction*). * Dossier documentaire numérique de l’application comportant le cahier des charges, des captures écran de certaines fonctionnalités, des maquettes d’interfaces, la charte graphique et un tableau qui synthétise l’ensemble des classes de l’application par couche logicielle. * Classes utilitaires et leur documentation technique. | | * Le diagramme de paquetages UML regroupant les classes par couche logicielle est réalisé (*tout autre diagramme équivalent peut être utilisé*). * Le diagramme de classes UML de l’application est réalisé. | |
| **Séance 2** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Les diagrammes construits lors de la séance précédente sont revus et corrigés par le chef de projet.  Votre tâche consiste maintenant à construire la structure MVC de l’application en créant les différentes classes modélisées et en intégrant les classes utilitaires. | | * Fiche des savoirs technologiques sur les concepts de base de la programmation orientée objet (POO). * Fiche de savoirs technologiques sur les IHM et la programmation événementielle. * Documents de modélisation de l’architecture logicielle (livrables de la séance précédente) revus et corrigés. * Environnement de développement intégré (IDE). | | * L’architecture MVC de l’application est implémentée. * Les classes sont créées (définition des attributs, des accesseurs/mutateurs et des constructeurs). * Les classes utilitaires sont intégrées dans l’application. | |
| **Séance 3** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | 1. Construire les différentes interfaces graphiques à l’aide d’un éditeur visuel. 2. Tester la création de ces interfaces en exécutant les opérations fournies dans la classe principale (version 1) importée depuis le dépôt du projet créé par le chef de projet. 3. Créer un nouveau dépôt distant pour la gestion des versions du projet. 4. Pousser et documenter la version actuelle de l’application sur le dépôt distant. | | * Fiche des savoirs technologiques portant sur les solutions de gestion de versions des solutions applicatives. * Synthèse des commandes les plus utilisées de la solution de gestion de versions. * Captures écran de certaines fonctionnalités et maquettes d’interfaces fournis dans le dossier documentaire de l’application. * Procédure d’installation et d’exploitation d’un éditeur visuel dans l’IDE. * Version 1 de la classe principale de l’application importée depuis le dépôt du projet. | | * L’éditeur visuel est intégré à l’IDE. * Les interfaces graphiques sont construites et leur code est généré. * La création des différentes interfaces graphiques est testée. * L’application est opérationnelle dans l’environnement de test. * La version actuelle de l’application est stockée sur le serveur distant. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Séquence 2.1. C2D** | **Implémenter les fonctionnalités d’une IHM développée dans un langage orienté objet** | | | | | |
| Durée totale en heures du scénario pour la séquence  12 h | *Rappel du contexte :* Vous participez au sein de l’équipe « développement et innovation » à l’évolution du patrimoine applicatif d’une entreprise cliente. Vous êtes en charge de la conception et de la réalisation d’une solution applicative métier.  Votre mission consiste à réaliser les différentes fonctionnalités liées à l’IHM de l’application. La documentation technique des composants développés et/ou adaptés est produite de manière incrémentale en adoptant le principe KISS (*Keep It Simple, Stupid*). | | | | | |
| **Compétences travaillées** | **Savoirs associés** | | **Indicateurs de performance** | | **Prérequis / Transversalités** |
| * Identifier, développer, utiliser ou adapter des composants logiciels. * Réaliser les tests nécessaires à la validation ou à la mise en production d’éléments adaptés ou développés. * Rédiger des documentations technique et d’utilisation d’une solution applicative. * Exploiter les fonctionnalités d’un environnement de développement et de tests. | Savoirs technologiques   * Méthodes, normes et standards associés au processus de conception et de développement d’une solution applicative. * Architectures applicatives ; concepts de base et typologies. * Techniques et outils d’analyse et de rétro-conception. * Typologies et techniques des cycles de production d’un service et acteurs associés. * Concepts de la programmation objet : classe, objet, abstraction, interface, héritage, polymorphisme, annotations, patrons de conception, interface de programmation d’applications. * Interfaces homme-machine : principes ergonomiques, techniques de conception, d’évaluation et de validation. * Concepts de la programmation événementielle : techniques de gestion des événements et exploitation de bibliothèques de composants graphiques. | | * Le choix des composants logiciels à utiliser et/ou à développer est pertinent. * La solution est développée dans les règles de l’art : * le développement répond à l’expression des besoins fonctionnels et respecte les contraintes techniques figurant dans le cahier des charges ; * un outil collaboratif de gestion des itérations de développement et de versions est utilisé ; * une documentation des versions vient appuyer l’intégration continue ; * les composants logiciels sont documentés de manière à être réutilisés. * L’application développée est opérationnelle conformément au cahier des charges et stable dans l’environnement de production. | | Prérequis :  B2.1. Séquence 2.1 B2D |
| **Séance 1** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 3 h | Votre tâche consiste à implémenter les parcours utilisateur définis par la direction métier.   1. Écrire les instructions réalisant la navigation en simulant l’exécution d’une tâche selon le parcours fourni. 2. Implémenter dans les classes concernées le contrôle de la validité de la saisie. 3. Exécuter le plan de test du parcours utilisateur. 4. Pousser et documenter la version actuelle de l’application sur le dépôt distant. | | * Fiche des savoirs technologiques sur les concepts de la POO. * Fiche de savoirs technologiques sur les IHM et la programmation événementielle. * Dossier documentaire de l’application. * Prototype du parcours utilisateur. * Plan de test comportant des parcours types. * Rapport de test à compléter. | | * La navigation entre les écrans de l’IHM est opérationnelle conformément au scénario établi. * Le développement de la navigation respecte les bonnes pratiques de programmation et le principe de couplage faible entre les couches MVC de l’application. * La gestion des événements liés à la navigation de l’IHM est réalisée. * Le rapport de test est complété. | |
| **Séance 2** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 3 h | Votre tâche consiste à implémenter les fonctionnalités de base de l’IHM.   1. Compléter les classes des couches modèle-vue-contrôleur pour implémenter les fonctionnalités de base de l’IHM. 2. Rédiger un plan de tests d’une fonctionnalité. 3. Exécuter le plan de tests. 4. Générer la documentation technique des classes métiers. | | * Fiche des savoirs technologiques sur les concepts de la POO. * Fiche des savoirs technologiques sur les IHM et la programmation événementielle. * Fiche des savoirs technologiques liés à la documentation technique. * Outil de génération de documentation technique. * Dossier documentaire de l’application : descriptions des cas d’utilisation des fonctionnalités de base de l’IHM. | | * Les cas d’utilisation sont implémentés. * Le plan de tests est rédigé. * Les classes métiers sont documentées. * La documentation technique des classes métiers est générée à l’aide d’un outil de génération de documentation. * L’application est opérationnelle dans l’environnement de test. | |
| **Séance 3** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 4 h | Le chef de projet vous demande de développer et de tester les fonctionnalités (*refactorisation, délégation, usage de patrons de conception..*.) du parcours type qu’il vous fournit.   1. Compléter le codage des classes des coches modèle-vue-contrôleur concernées. 2. Exécuter le plan de tests fourni 3. Rédiger la documentation d’utilisateurs du cas d’utilisation spécifié. 4. Pousser la nouvelle version sur le dépôt distant. | | * Fiche des savoirs technologiques sur les concepts avancés de la POO. * Fiche des savoirs technologiques sur les IHM et la programmation événementielle. * Dossier documentaire : cas d’utilisation des fonctionnalités à implémenter. * Plan de tests. | | * Les fonctionnalités du parcours type sont implémentées conformément à la description des cas d’utilisation. * Les tests fonctionnels planifiés sont réalisés. * Le document de synthèse des tests réalisés est rédigé. * La documentation d’utilisateur de la fonctionnalité spécifiée est rédigée. * Les composants développés et/ou adaptés sont documentés dans l’application. * La version actuelle du projet est enregistrée sur le dépôt distant. | |
| **Séance 4** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 2 h | La revue de code de l’application réalisée par le chef de projet a révélé quelques exceptions qui ne sont pas gérées dans certaines parties du code.   1. Traiter les exceptions répertoriées dans le document de revue de code. 2. Pousser la nouvelle version sur le dépôt distant. 3. Déployer l’application dans l’environnement de production. 4. Tester l’application dans l’environnement de production. | | * Fiche des savoirs technologiques sur les concepts **avancés** de la POO. * Fiche des savoirs technologiques sur les IHM et la programmation événementielle. * Document de synthèse de la revue de code. * Procédure de déploiement de l’application dans l’environnement de production. * L’environnement de production (*machine virtuelle sur un serveur physique ou virtualisé*). | | * Les bugs répertoriés dans le document de synthèse de la revue de code sont corrigés. * Les exceptions sont traitées. * Le document de synthèse de la revue de code est mis à jour avec les nouvelles corrections. * L’application est opérationnelle dans l’environnement de test. * L’application est déployée dans l’environnement de production. * L’application est testée dans l’environnement de production. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Séquence 2.1. D2D** | **Développer une solution logicielle avec persistance des données dans une base de données relationnelle** | | | | | |
| Durée totale en heures du scénario pour la séquence  6 h | Vous assurez une prestation de développement auprès d’une jeune pousse cliente spécialisée dans le développement d’interfaces graphiques (*front*-*end*). Vous intervenez sur la persistance des données d’une solution applicative existante (partie *back-end*). L’application est destinée aux développeurs de la startup qui doivent être authentifiés pour l’utiliser. Toutes les interfaces nécessaires à la gestion des utilisateurs (création, modification, suppression) dans l’application sont développées. Votre interlocuteur privilégié est un développeur interne à l’entreprise spécialisé dans les interfaces graphiques et qui est également un futur utilisateur de l’application.  Chaque version de l’application fait l’objet d’un envoi sur le dépôt distant du projet et d’une validation de la part du chef du projet agile. | | | | | |
| **Compétences travaillées** | **Savoirs associés** | | **Indicateurs de performance** | | **Prérequis / Transversalités** |
| * Participer à la conception de l’architecture d’une solution applicative * Identifier, développer, utiliser ou adapter des composants logiciels * Utiliser des composants d’accès aux données * Réaliser les tests nécessaires à la validation ou à la mise en production d’éléments adaptés ou développés * Rédiger des documentations technique et d’utilisation d’une solution applicative * Exploiter les fonctionnalités d’un environnement de développement et de tests | Savoirs technologiques   * Architectures applicatives ; concepts de base et typologies. * Concepts de la programmation objet : classe, objet, abstraction, interface, héritage, polymorphisme, annotations, patrons de conception, interface de programmation d’applications. * Interfaces homme-machine : principes ergonomiques, techniques de conception, d’évaluation et de validation. * Techniques de gestion des erreurs et des exceptions. * Fonctionnalités avancées d’un environnement de développement.   Savoirs économiques, juridiques et managériaux   * Réglementation en matière de collecte, de traitement et de conservation des données à caractère personnel | | * Le choix des composants logiciels à utiliser et/ou à développer est pertinent. * La solution est développée dans les règles de l’art : * le développement répond à l’expression des besoins fonctionnels et respecte les contraintes techniques figurant dans le cahier des charges ; * les composants logiciels sont documentés de manière à être réutilisés ; * un document est rédigé pour chaque contexte d’utilisation de l’application et est adapté à chaque destinataire tant par son contenu que par sa présentation. * Les données persistantes liées à la solution applicative sont exploitées à travers un langage de requête lié à la base de données qui peut être le langage de requête proposé par les échanges applicatifs des technologies Web, un langage de requête présent dans l’outil de correspondance objet-relationnel ou toute autre solution de persistance * L’application développée est opérationnelle conformément au cahier des charges et stable dans l’environnement de production. | | Prérequis :  B1.3 Développer la présence en ligne de l’organisation.    Transversalité :  B3.1 Protéger les données à caractère personnel.  B2.3. Gérer les données. |
| **Séance 1** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | La modélisation de la base de données (BDD) ainsi que les informations d’accès sécurisés au serveur et à la base vous sont fournies.   1. Tester l’accès sécurisé au serveur et à la base de données. 2. Installer un serveur de bases de données dans votre environnement de travail. 3. Installer la base de données sur le serveur. 4. Intégrer dans l’IDE les composants de l’API d’accès aux données. | | * Fiche de savoirs technologiques sur la persistance et la couche d’accès à la base de données. * Serveur de bases de données (machine virtuelle ou conteneur Docker). * Procédure d’installation du serveur de base de données. * Modélisation de la base de données. * Script de sauvegarde de la base de données. * Procédure d’accès sécurisé au serveur de la base de données. | | * L’accès sécurisé au serveur de la base de données est opérationnel. * Le serveur de base de données est installé dans l’environnement de test. * La base de données est installée sur le serveur de test. * Les composants de l’API d’accès aux données sont installés dans l’environnement de développement. | |
| **Séance 2** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | 1. Modifier la classe Modèle d’accès aux données. 2. Implémenter la méthode de connexion au serveur et à la base de données à travers une interface de programmation. 3. Utiliser le patron de conception Singleton pour la connexion au serveur. 4. Tester la connexion depuis l’application. | | * Fiche des savoirs technologiques sur les concepts de la POO. * Fiche de savoirs technologiques sur les IHM et la programmation événementielle. * Fiche de savoirs technologiques sur la persistance et la couche d’accès à la base de données. * L’archive du code source de l’application. | | * L’accès à la base de données depuis l’application est opérationnel et respecte l’architecture MVC. * La connexion au serveur met en œuvre le patron de conception Singleton. | |
| **Séance 3** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Votre tâche consiste à Implémenter le traitement du récit utilisateur « Ajouter un nouvel utilisateur ».   1. Écrire le code relatif au traitement du formulaire pour persister l’objet renseigné en base de données via l’API installée. 2. Tester la persistance de l’objet dans la base de données. | | * Fiche de savoirs technologiques sur la persistance, la couche d’accès à la base de données et le langage SQL. | | * La création d’un nouvel utilisateur est effective en base de données. * La table des utilisateurs est mise à jour. * Les requêtes d’accès à la base de données sont paramétrées. | |
| **Séance 4** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Votre tâche consiste à implémenter et à tester le traitement des récits d’utilisateur « Authentifier un utilisateur ». | | * Fiche de savoirs technologiques sur la persistance, la couche d’accès à la base de données et le langage SQL. | | * L’authentification d’un utilisateur est opérationnelle. * Les requêtes d’accès à la base de données sont paramétrées. | |
| **Séance 5** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Votre tâche consiste à implémenter et à tester le traitement du récit utilisateur « Supprimer un utilisateur ». | | * Fiche de savoirs technologiques sur la persistance, la couche d’accès à la base de données et le langage SQL. | | * La suppression d’un utilisateur est effective en base de données. * La table des utilisateurs est mise à jour. | |
| **Séance 6** | **Tâches à réaliser** | | **Ressources fournies** | | **Résultats attendus** | |
| 1 h | Votre tâche consiste à implémenter et à tester le traitement du récit utilisateur « Modifier un utilisateur ». | | * Fiche de savoirs technologiques sur la persistance, la couche d’accès à la base de données et le langage SQL. | | * La modification d’un utilisateur est effective en base de données. * La table des utilisateurs est mise à jour. | |