Conception détaillée

Projet de Programmation

L2D1

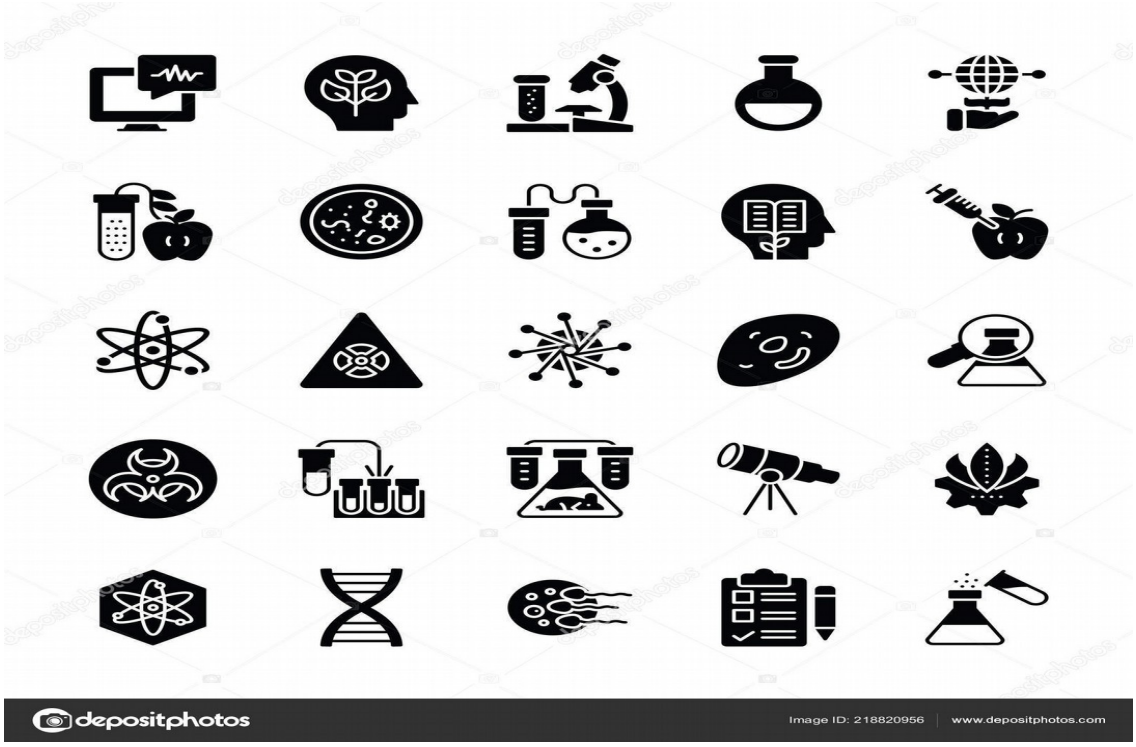


Illustration 1: ***Image de présentation***

Boutajar Manal
Bodnar Adelin
Hamiche Mehdi

Groupe L2D1

Les informations d'identification du document :

Référence du document :
Version du document : 2
Date du document : 01/05/2021
Auteurs : **Mehdi Hamiche** **Manal Boutajar** **Adelin Bodnar**

Les éléments de vérification du document :

Validé par :
Validé le :
Soumis le :
Type de diffusion :
Confidentialité :

Les éléments d'authentification :

Maître d'ouvrage :	**Chef de projet :**
Date / Signature :	**Date / Signature :**

Sommaire

1. Introduction

La conception détaillée affine la conception générale en présentant toutes les fonctions, méthodes, classes, paquetages, et librairies qui seront nécessaires au bon développement de l'application.

Ce document vise à faciliter l'implémentation de notre application par les développeurs et vise à garantir que le fonctionnement de l'application correspondra bien aux besoins de l'utilisateur final, ce qui permettrait au développeur de pouvoir réaliser facilement une traduction de ce document en langage de programmation.

Ce document est aussi organisé en fonction de l'architecture de l'application, en répétant autant de fois que nécessaire les rubriques suivantes. Pour chaque composant logiciel issu de la conception générale.

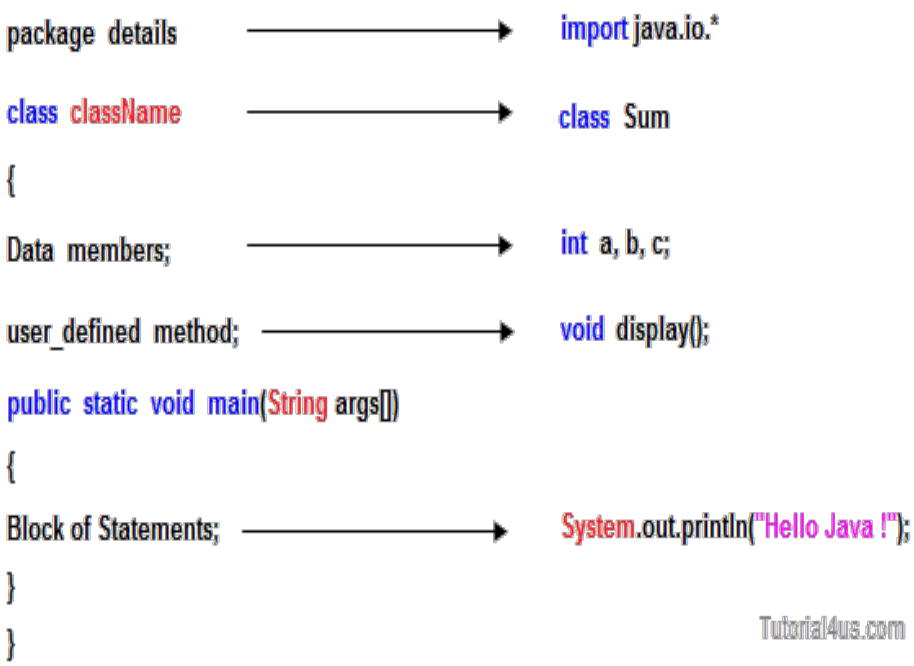


Illustration 2: ***Structure d'un programme Java avec ses fonctions, méthodes, paquetages, etc...***

2. Guide de lecture

2.1. Maîtrise d'œuvre

La maîtrise d'œuvre présente l'équipe du développement chargé du bon suivi de la conception détaillée et des besoins dont le maître d'ouvrage fait commande.

Elle représente l'équipe du développement :

- Adelin Bodnar
- Manal Boutajar
- Mehdi Hamiche

Cette équipe veillera au bon suivi de la conception détaillée coordonnées avec le conception générale représentant les besoins des enseignants encadrants.

2.1.1. Responsable

Il est conseillé pour le responsable de la maîtrise d'œuvre de lire le document dans sa totalité afin de prendre conscience de l'ensemble des éléments.

2.1.2. Personnel administratif

Il est conseillé pour le personnel administratif de lire la présentation du produit ainsi que la décomposition des tâches.

2.1.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte la partie de la description du module, avec la liste des composants utilisés par ce composant et ceux utilisant ce composant, ainsi que la liste des classes / structures de données / méthodes / fonctions du composant.

2.2. Maîtrise d'ouvrage

La maîtrise d'ouvrage représente dans notre cas le client du projet, c'est-à-dire les personnes dont les besoins permettent la conception du projet.

La maîtrise d'ouvrage est assistée par l'équipe de la maîtrise d'œuvre et donc ce rôle sera assuré par les enseignants encadrants Dragutin Jastrebic et Koviljka Lukic Jastrebic.

2.2.1. Responsable

Il est conseillé pour le responsable de la maîtrise d'ouvrage de lire le document dans toute sa totalité afin de prendre conscience de l'ensemble des documents.

2.2.2. Personnel administratif

Il est conseillé pour le personnel administratif de lire la présentation du produit ainsi que la décomposition des tâches.

2.2.3. Personnel technique

Il est conseillé pour le personnel technique de prendre en compte la partie de la description du module, avec la liste des composants utilisés par ce composant et ceux utilisant ce composant, ainsi que la liste des classes / structures de données / méthodes / fonctions du composant.

3. Description du module

1. Rappel des objectifs du composant

Le produit que nous réalisons sous forme d'application web, sera livré en état de marche avec son code source. La documentation du produit contiendra les manuels d'utilisation et d'installation, le cahier des charges, le cahier de recettes, et d'autres documents nécessaires au projet.

1.2. Lister les composants utilisés par ce composant et ceux utilisant ce composant

Pour pouvoir utiliser l'application, l'élément principal qui lui est lié est d'avoir un pc / ordinateur dont la version des mises à jour soit assez récente, cela permettrait de pouvoir utiliser à 100% toutes les fonctionnalités et d'en avoir les meilleures performances.

Le site lui-même possède des composantes qui sont représentées par les fonctionnalités comme, aligner les séquences protéiques ou nucléotidiques et les représenter sous forme graphique.

1.2.1. Définir les différentes interfaces et leurs visibilité

Il existe trois interfaces pour l'application, la première c'est la page d'accueil qui redirige directement l'utilisateur vers l'interface possédant les fonctionnalités de l'alignement global ou bien multiple des séquences.

1.2.2. Décomposition en tâches

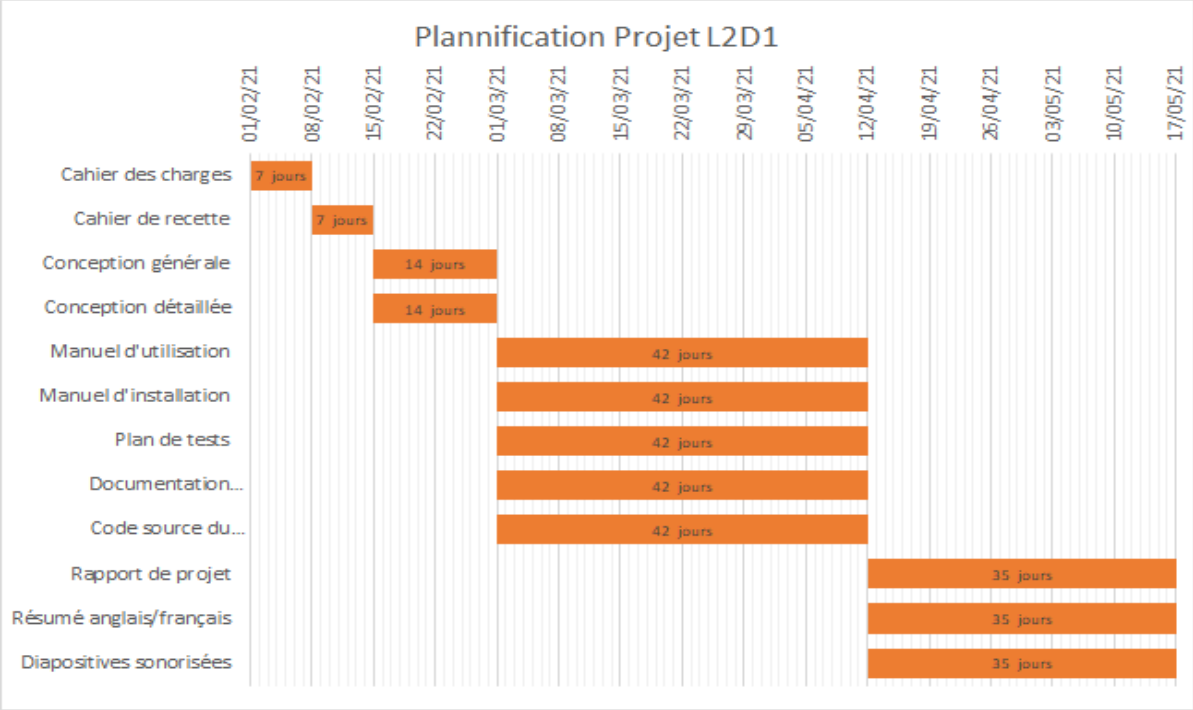


Illustration 3: ***Diagramme de Gantt***

1.2.3. Méthodes et outils

Les outils de développement du point de vue technique, permettant de réaliser le projet, seront :

- Les langages de programmation JAVA, HTML, CSS, PYTHON, R;
- Le SVN pour pouvoir gérer la synchronisation des différentes versions des fichiers sources.

1.2.4. Standard et outils

Les outils et standard qui supporteront la documentation est le plan de test.

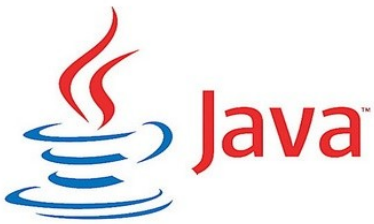


Illustration 4: ***Logo JAVA***

Illustration 5: ***Logo PHP***

Illustration 6: ***Logo Javascript***

Illustration 7: ***Logo Python***

Illustration 8: ***Logo CSS et HTML***

Illustration 9: ***Logo SVN***

1.2.5. Références

- Le cahier des charges ;
- L'architecture et les modules.

1.3 Lister les classes / les structures de données / les méthodes / les fonctions du composant. Pour chacune :

1.3.1. Explication de son rôle

Classes	Structure de données	Fonctions	Autres
• **public class AlignementG** • // Liste des alignements possibles • // Séquences à aligner • // Calcul des alignements • // Calcul des scores	• ***public* AlignementG (Sequence seq_haut,Sequence seq_gauche)**	• public int init_chemins (int i,int j) • // Calcul des alignements public void backtrack(int i, int j, StringBuffe r seq_sup, Strin gBuffer seq_inf) • // affiche les alignements et	• public : méthode qui peut être appelée par n'importe quel objet

• // affiche les alignements et leur scores		leur scores public void affiche() {	
• **class Chemin** • // Lecture • // Écriture	• **public Chemin()**	• // Lecture public String getSeq_sup() { return seq_sup; } • public Sequence getSeq_inf() { return seq_inf; } • public int getScore() {return score; } • // Ecriture public void setScore(int s) { score=s; } • public void setSeq_sup(St	• void : aucune valeur ne sera retournée par la méthode

		ring s) {
		seq_sup=s;
		}
		• public void setSeq_inf(String s) { seq_inf=s;
		}	

• **public class Alignement_ Multiple**
//alignement des séquences
//affichage des résultats
	• **Alignement_ Multiple(Fasta1 f)**	• public void calcul()
		• public void afficheSeqMult()	

• **class Fleches** • // Élément pour mise en mémoire du mouvement qui donne le meilleur score • // Accès	• **public Fleches(int haut,int gauche,int diagonale,int max)**		• main : partie principale du programme, permettant l'exécution d'une application Java • Méthode main : devient

mouvements			la méthode à partir de laquelle démarre automatiquement le programme lorsqu'elle est trouvé
• **public class Matrice** • // Tableau valeurs et séquences • // Tableau de mise en mémoire mouvements • // Remplissage séquences et valeurs gap • // Remplissage score et mouvements • // Test	• **public Matrice(StringBuffer seq1,StringBuffer seq2)**	• // Remplissage séquences et valeurs gap public void init_matrice() • //Remplissage score et mouvements public void calcul() • // Test - public void affiche() • // Test - public void afficheFleches()	• static :méthode de classe - accessible même s'il n'existe aucune instance de la classe.
• **public class**		• public static	• L'argument de

Main		void main(String[] args) { • Une seule classe qui renferme une méthode main regroupant les instructions destinées à être exécutées. • Utiliser une méthode main() avec un type de retour, mais non appelée lorsque la classe sera invoquée.	la méthode main: tableau d'éléments textuels qui fournit le moyen de passer des informations à l'application. Chaque chaîne de caractères du tableau est appelée un argument de ligne de commande
• **public class Sequence**	• **public Sequence(String id, StringBuffer seq)**	• public String getId() { return id; } • public static String getNature() { return	

		nature;}	
• **public class Fasta1**	• **public Fasta1 (String nom,File fichier)** • **public Fasta1(String sequences)**	• public String getNom() { return nomFichier; } • public ArrayList getSequences (){ return sequences; }	•

1.3.2. Définir le type de tous les attributs / variables la composant

- **public class AlignementG {**

  private ArrayList<Chemin> chemins ;

  private Sequence seq_haut ;

  private Sequence seq_gauche ;

  private Matrice matrice ;

- **class Chemin {**

  private Sequence seq_sup ;

  private Sequence seq_inf ;

private int score; …

- **public class Alignement_Multiple {**

  private Fasta1 fasta ;

  private ArrayList<Sequence> sequences ;

  private ArrayList<AlignementG> alignements

- **public class Fasta1 {**

  private String nomFichier ;

  private ArrayList<Sequence> sequences ;

- **public class Matrice {**

  private static int match ;

  private static int mismatch ;

  private static int gap ;

  private StringBuffer seq_haut ;

  private StringBuffer seq_gauche ;

  private ArrayList<ArrayList<Integer>> tableau ;

  private ArrayList<ArrayList<Fleches>> tableauFleches ;

- **class Fleches {**

  private ArrayList<String> fleches ;

- **public class Sequence {**

  private String id ;

  private StringBuffer seq ;

  private static String nature ;

- **class ADN extends Sequence {**

  private String alphabet ;

- **class Proteine extends Sequence {**

  private String alphabet

1.3.3. Préciser l'algorithme utilisé pour chaque méthode / fonction

Chaque élément présenté correspond à un besoin exprimé dans le cahier des charges et à un point qui sera vérifié et validé dans le cahier de recettes.

Algorithme de Needleman et Wunsch

Alignement global

- Développé pour aligner deux séquences protéiques. Soit A et B deux séquences de longueur m et n.

- L'algorithme construit un tableau à deux dimensions (m,n) que l'on appelle matrice de comparaison. L'équation suivante résume le principe de calcul d'une case de cette matrice :

  S (i, j) = se (i, j) + MAX ((S (i+1,j+1)),(S (x, j+1) - P) ;(S (i+1, y) - P))

Alignement multiple

L'alignement multiple consiste à aligner collectivement un ensemble de séquences homologues, comme des séquences de protéines assurant des fonctions similaires dans différentes espèces vivantes.

1. Aligner les trois séquences ;
2. Exemple : "ACTG / CTTG / CCTG" ;
3. Exemple : "MPRCLCQRINCYA / PYRCKCRNICIA / PPRCLCQRINCIA" ;
4. cytochromes c - trois séquences (minimum) ;
5. cytochromes c / autres séquences pour les tests faire plusieurs alignements multiples changeant le nombre de séquences à aligner, respectant aussi les critères demandés au début (nature, longueur, nombre de séquences).

4. Annexes

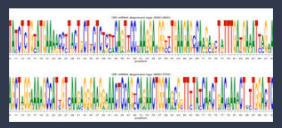
Cahier des charges, Cahier de recettes et Maquette Figma

Introduction à la bioinformatique

Maquette du produit

Groupe L2D1

Description du projet



- Programme qui analyse et aligne des séquences protéiques,
- Accéder aux bases biologiques,
- Manipuler les listes, fonctions, objets, matrices,
- 3 approches : manuel, programmable et production,
- Les protéines sont stockées dans des bases de données,
- Les types d'alignements : global, local, multiple,
- Affichage des résultats.

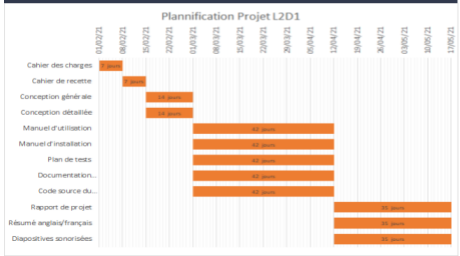
Technologie utilisées :

- **HTML** : Ce langage sera utilisé pour définir la structure des pages,
- **CSS** : Ce langage servira à styliser l'ensemble des pages,
- **PHP** : Ce langage permettra la communication entre le client et le serveur,
- **Javascript** : Afin d'animer nos pages web.

Objectif

- Travailler sur les parties traitement et visualisation des données - domaine éducatif,
- Analyse de la structure primaire (amino acides en lettres) des séquences protéiques (cytochrome c et séquences pour les tests) et des séquences nucléiques (gènes) à travers des alignements réalisés,
- Présenter les résultats des alignements de séquences protéiques et nucléique avec interface graphique.

Différents délais



- **Le cahier des charges** (à rendre semaine 3);
- **Le cahier de recette** (à rendre semaine 4);
- **La conception générale**;
- **La conception détaillée** (à rendre semaine 5);
- **Le manuel d'utilisation** (à rendre semaine 11);
- **Le manuel d'installation** (à rendre semaine 11);
- **Le plan de tests** (à rendre semaine 11);
- **La documentation interne du code** (à rendre semaine 11);
- **Le code sources du programme** (à rendre semaine 11);
- **Le rapport du projet** (avant la soutenance);
- **Le résumé en français et en anglais** (avant la soutenance);
- **Les diapositives sonorisées** (avant la soutenance).

5. Glossaire

★ ***BIO-INFORMATIQUE :*** discipline permettant d'analyser de l'information biologique qu'il y a dans les séquences nucléotidiques et protéiques

★ ***PROGRAMMATION ORIENTÉE OBJET :*** met en œuvre différents objets (instances de classes). Chaque objet associe des données et des méthodes (fonctions) agissant exclusivement sur les données

★ ***ALGORITHME NEEDLEMAN - WUNSCH :*** effectue un alignement global maximal de deux chaînes de caractères

★ ***ALIGNEMENT GLOBAL :*** recouvre les séquences alignées (par exemple 2 séquences) sur l'ensemble de leur longueur

★ ***ALIGNEMENT MULTIPLE :*** aligner un ensemble de séquences homologues, comme des séquences protéiques ou nucléiques qui assure des fonctions similaires dans différentes espèces vivantes

★ ***WEBLOGO :*** application web conçue pour rendre la génération de logos de séquence aussi simple que possible

★ ***INTERFACE GRAPHIQUE :*** relie les 3 modules et les présente de manière simple et efficace

★ ***GAP, MATCH, MISMATCH :*** valeurs pour alignement global

★ ***FASTA :*** format de fichier texte dans lesquels les séquences sont affichées par une suite de lettres

★ ***NCBI :*** National Centre for Biotechnology Information

★ ***PROTEIN :*** Base de données pour la récupération des séquences protéiques en format .fasta

★ ***GENE :*** Base de données pour récupération des séquences nucléiques en format .fasta

★ ***NUCLEOTIDE :*** Base de données pour la récupération des séquences nucléiques en format .fasta

★ ***UNIPROT :*** Base de données pour la récupération des séquences protéiques en format .fasta

★ ***DDBJ :*** Base de données pour la récupération des séquences nucléiques en format .fasta

★ ***DNA :*** Data Bank Japan

★ ***EMBL :*** European Molecular Biology Laboratory

★ ***EMBOSS*** : outil bioinformatique pour l'alignement global – NEEDLE

★ ***BLAST*** : outil bioinformatique pour l'alignement local

★ ***MULTALIN*** : outil bioinformatique pour l'alignement multiple

6. Références

Alignement de séquences

BNLEARN

DDBJ

National Centre for Biotechnology Information

Réseaux Bayésiens

UniProt

Alignement global

Alignement multiple

JAVA - Main (site 1)

JAVA - Main (site 2)

Chaînes de caractères

Tableau

Needleman et Wunsch – Java

Needleman et Wunsch - Python

7. Index

Index des figures

Index lexical