# Traumato: Gestion Cabinet Traumatologie

## Table de contenu

* [Traumato: Gestion Cabinet Traumatologie](#traumato-gestion-cabinet-traumatologie)
  + [Table de contenu](#table-de-contenu)
  + [Contexte du projet](#contexte-du-projet)
  + [Objectif du projet](#objectif-du-projet)
  + [Périmètre des clients](#périmètre-des-clients)
  + [Maquettage](#maquettage)
    - [Charte Graphique](#charte-graphique)
    - [Maquette](#maquette)
    - [Prototype](#prototype)
  + [Choix Technologique](#choix-technologique)
    - [Planning](#planning)
    - [Modelisation](#modelisation)
    - [Frontend](#frontend)
    - [Backend](#backend)
    - [Base de données](#base-de-données)
  + [Description fonctionnelle des besoins](#description-fonctionnelle-des-besoins)
    - [Fonctions Principal](#fonctions-principal)
    - [Fonctions Secondaires](#fonctions-secondaires)
  + [Phases de projet (Delais)](#phases-de-projet-delais)
  + [Livrables](#livrables)

## Contexte du projet

Le docteur Rahbani Rafik est un traumatologue qui souhaite renforcer sa présence sur le web et informatiser la gestion de ses patients.

Le besoin minimum est de réaliser une page publicitaire du cabinet, et une page de gestion des patients (Ajouter,Modifier,Supprimer, et afficher).

## Objectif du projet

Nous voulons réaliser une site web pour renforcer la présence sur du docteur Rahbani Rafik le web et informatiser la gestion de ses patients.

## Périmètre des clients

Nous nous concentrons sur les clients Maroccain qui ont 20+ ans et besoin d’un **traumatologue**.

## Maquettage

### Charte Graphique

### Maquette

### Prototype

## Choix Technologique

### Planning

* **Tableau Gantt**: TeamGantt
* **Tableau Kanban + Scrum**: Trello

### Modelisation

* **UML**: Est destiné à faciliter la conception des documents nécessaires au développement d’un logiciel orienté objet, comme standard de modélisation de l’architecture logicielle.
* **Draw.io**: L’interface est simple et facile d’utilisation, les objets sont classés à gauche de l’écran par thèmes, une moteur de recherche vous permet d’interroger la base de données de dessins. Une fois votre travail terminé, la sauvegarde est réalisée au format XML (Diagramly XML Document), l’exportation propose différents formats (png, jpg, gif, svg, html, intégration en pages web…).. Il est possible d’importer un diagramme au format XML.

### Frontend

* **Pug.js**: Est de vous permettre de coder la structure d’un site dans une syntaxe bien plus simple et lisible que celle du HTML, et de rendre votre code maintenable en vous offrant la possibilité de créer entre autres, des variables, mixins, includes…
* **SCSS**: *(Sassy Cascading StyleSheet)*Est mieux organisée, plus lisible, moins répétitive. Ce qui booste les performances de votre site et rend le travail sur le fichier plus agréable.
* **BEM**: *(Block Element Modifier)*st une Methodologie qui donne l’indépendance des blocs. Cela signifie que le fonctionnement du code est prévisible et qu’il n’y a pas de croisements de noms.
* **JavaScript**: Est un langage de programmation qui permet de créer du contenu mis à jour de façon dynamique, de contrôler le contenu multimédia, d’animer des images, et tout ce à quoi on peut penser.
* **JQuery**: Permet, entre autres, de gagner en rapidité dans l’interaction avec le code HTML d’une page Web.

### Backend

* **PHP**: *(Hypertext Preprocessor)* est principalement conçu pour servir de langage de script coté serveur, ce qui fait qu’il est capable de réaliser tout ce qu’un script CGI quelconque peut faire, comme collecter des données de formulaire, générer du contenu dynamique, ou gérer des cookies.
* **MVC**: *(Model View Controller)* le pattern MVC permet de bien organiser son code source. Il va vous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l’on retrouve dans des fichiers distincts.

### Base de données

* **SQL**: *(Structured Query Language)* est un langage informatique utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données.
* **MySQL**: est un serveur de bases de données relationnelles développé dans un souci de performances élevées en lecture, ce qui signifie qu’il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées.

## Description fonctionnelle des besoins

### Fonctions Principal

| Fonction | Prendre Rendez-vous |
| --- | --- |
| Objectif | Creation de Patient et Rendez-vous |
| Description | Une formulaire pour créer des patients avec des Rendez-vous |
| Contraintes / règles de gestion | N/A |
| Niveau de priorité | Priorité Haute |

| Fonction | Connextion |
| --- | --- |
| Objectif | S’authentifier au platform en tant que Admin ou Medecin |
| Description | Une formulaire d’inscription avec nom d’utilisateur et mot de passe, une Admin s’authentifier avec une clé twoFactorAuth de plus |
| Contraintes / règles de gestion | Seulement les Admins et les Medecins peut authentifier au platform |
| Niveau de priorité | Priorité Haute |

| Fonction | Gestion des Patients |
| --- | --- |
| Objectif | Ajouter, Modifier, Supprimer, et Afficher les patients |
| Description | Une dashboard contient des actions a faire pour manager des patients. Un patient est défini par son nom, prénom, date de naissance et sa maladie. |
| Contraintes / règles de gestion | Seulement les Médecins et les Admins peut gérer des patients |
| Niveau de priorité | Priorité Haute |

### Fonctions Secondaires

| Fonction | Contact |
| --- | --- |
| Objectif | Envoi de mail de contact vers l’address de Cabinet |
| Description | Une formulaire pour envoyer des emails au address mail de Cabinet avec PHPMailer et google smtp server |
| Contraintes / règles de gestion | N/A |
| Niveau de priorité | Priorité Normal |

| Fonction | Telecharger report des patients |
| --- | --- |
| Objectif | Sauvegarder tout les Patients |
| Description | les Médecins et les Admins peut telecharger une CSV avec des information de toutes les patients |
| Contraintes / règles de gestion | Seulement les Médecins et les Admins peut telecharger le report |
| Niveau de priorité | Priorité Normal |

| Fonction | Gestion des Médecins |
| --- | --- |
| Objectif | Ajouter, Modifier, Supprimer, et Afficher les Médecins |
| Description | Une dashboard contient des actions a faire pour manager des médecins. Un Médecin est définie par son nom, prénom,date de naissance et sa spécialité. |
| Contraintes / règles de gestion | Seulement les Admins peut gérer des Médecins |
| Niveau de priorité | Priorité Normal |

| Fonction | Reinitialisé le mot de passe |
| --- | --- |
| Objectif | Reinitialisé le mot de passe de Medecin |
| Description | Envoyer un email au address de Medecin avec son mot de passe |
| Contraintes / règles de gestion | Seulement les Admins et les Medecins peut reinitalisé leur mot de passe |
| Niveau de priorité | Priorité Faible |

## Phases de projet (Delais)

Pour plus d’information sur les phases et délais de projet aller au Tableau Gantt [ici](https://prod.teamgantt.com/gantt/schedule/?ids=2927527&public_keys=pHfgvPgVU4G9&zoom=d120&font_size=12&col_width=455&documents=0&comments=0&estimated_hours=1&assigned_resources=1&percent_complete=1&hide_header_tabs=1&menu_view=0&resource_filter=1&name_in_bar=0&name_next_to_bar=0&resource_names=1#)

| Numéro | Nom de phase | Date du rendu | durée |
| --- | --- | --- | --- |
| 1 | **Initialisation** | 10/01 | 2 jours/homme |
| 2 | **Conception** | 12/01 | 2 jours/homme |
| 3 | **Maquettage** | 14/01 | 2 jours/homme |
| 4 | **Frontend** | 17/01 | 3 jours/homme |
| 5 | **Base de Données** | 18/01 | 1 jour/homme |
| 6 | **Backend** | 22/01 | 4 jours/homme |
| 7 | **Déploiement** | 24/01 | 2 jours/homme |

## Livrables

Pour plus d’information sur les levrables et les taches aller au Tableau Trello [ici](https://trello.com/b/2wAe2cHY/traumato-gestion-cabinet-traomatologie)

* Phase 1: **Initialisation** [Markdown]
  + Github Repo
  + Un Cahier des Charges
  + Tableau Gantt
  + Tableau Trello
* Phase 2: **Conception** [UML]
  + Diagramme de Cas d’utilisations
  + Diagramme de Classes
  + Diagrammes de Séquences
* Phase 3: **Maquettage** [Figma]
  + Charte graphique adéquat du site web
  + Wireframe
  + Maquette
  + Prototype
* Phase 4: **Frontend** [Pug (Html), SCSS (CSS), JavaScript (Regex), jQuery (Ajax)]
  + Composants et page Template
  + Page Home
  + page Contact
  + Page App
* Phase 5: **Base de Données** [SQL (MySQL)]
  + Base de Données
  + Table Patient
  + Table Médecin
* Phase 6: **Backend** [PHP (PHPMailer)]
  + Composant d’accès aux données
  + Fonctionnalité page Contact
  + Fonctionnalité page App
* Phase 7: **Déploiement** [Heroku, RemoteMySQL]
  + Déploiement de base de données
  + Déploiement de site web