



Rapport de :

MINI-PROJET EN LANGAGE C

Sujet :

**DEVELOPPER UNE APPLICATION POUR LA GESTION D'UNE AGENCE DE
LOCATION DE VOITURES**

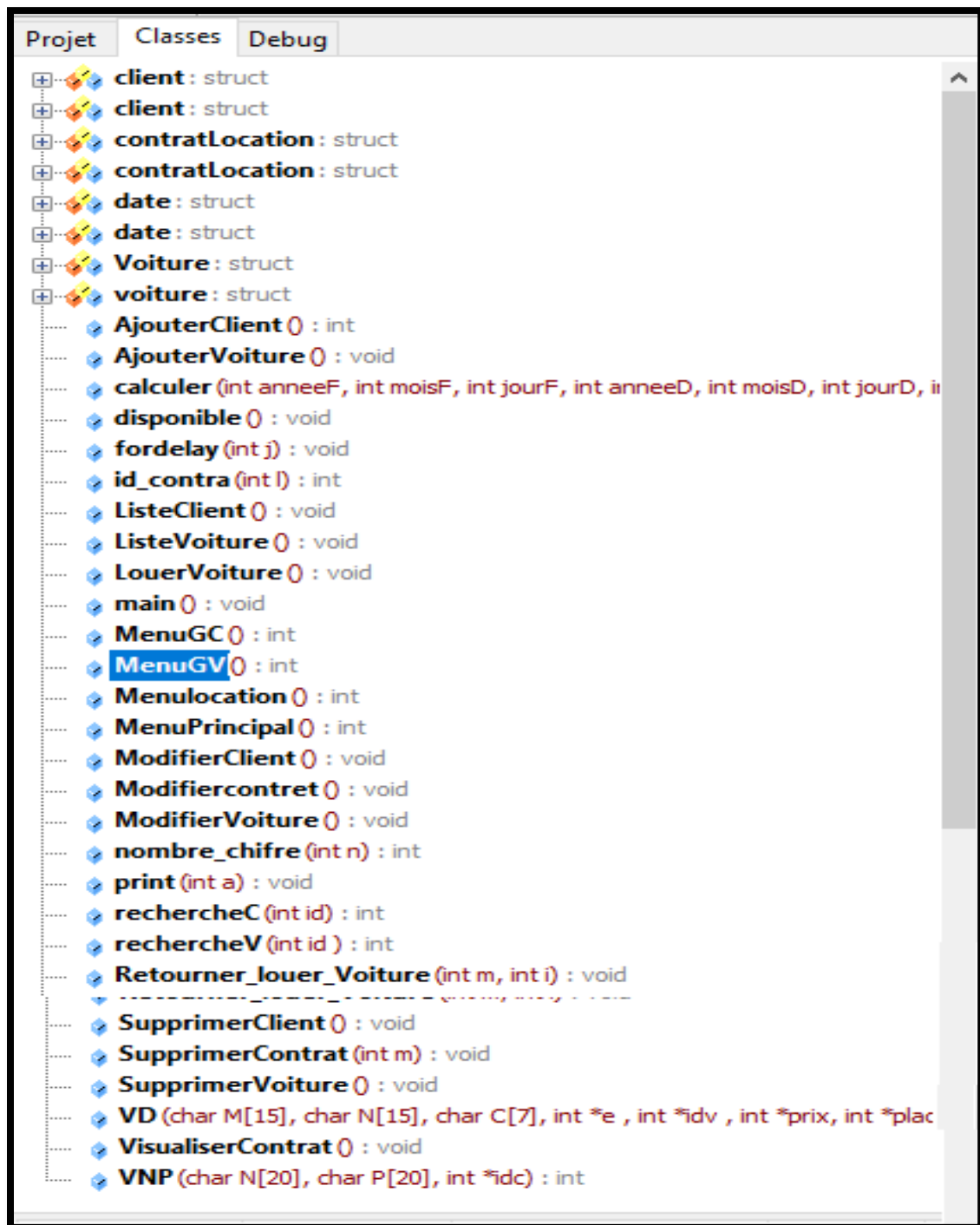
Réalisé par :

EL AZZAOUY MOHAMED

Filière : SMI4

❖ La structure du Programme :

L'application contient un ensemble des fonctions :



- **Menu Principal** : ou l'accueil pour objectif de présenter le contenu de l'application.

[illegible]

```
Menu Principal
*****
```

```
Location.....1
Gestion voitures.....2
Gestion clients.....3
Quitter.....4
```

Votre choix :

- Menu de gestion des voitures, clientes et des locations : permet de présenter les options de chaque menu et retourné un choix.

- Main :

Il permet de faire le choix entre les autres menus, et ilsécurisé l'application avec Un mot de passe.

Le mot de passe est : ELOUAFDI

```

830 void main() { system("color E0");
831     char pass[10], password[10] = "ELOUAFDI", ppassword[10] = "med";
832     int i = 0;
833     repet:
834     printf("\n\n\t\tEntrez le mot de passe (code-->ELOUAFDI) :"); scanf("%s", pass);
835
836     if (strcmp(pass, password) == 0 || strcmp(pass, ppassword) == 0) {
837         printf("\n\nPassword Match!\nChargement");
838         for(i = 0; i <= 6; i++) {
839             fordelay(100000000);
840             printf(".");
841             system("cls");
842         }
843
844     else
845         { printf("\n\nLe mot de passe est un correct !!\a\a\a"); goto repet;}
846
847     int choix = MenuPrincipal(), id;
848 do{
849     switch(choix){
850
851     case(1):      system("cls");      system("color B0");      choix = Menulocation();
852     do{
853
854
855
856
857
858
859
860
861
862
863     case(2):      system("cls");      system("color C0");      choix = MenuGV();
864     do{
865
866
867
868
869
870
871
872

```

❖ Gestion des voitures :

- liste des voitures :

Permet d'afficher la liste des voitures dans un tableau.

La fonction **nombre_chifre()** calculer le nombre des chiffres d'un nombre, et la fonction **print()** ajoute le nombre des espaces nécessaire pour la bonne présentation.

```

764
765 int nombre_chifre(int n){
766     int i=0;
767     do{ i++;      n=n/10;          }while(n!=0);
768     return i;
769 }
770
771
772
773 void print(int a){
774
775     for(a;a<22;a++){
776         if(a==20) printf("\xb3");
777         else      printf(" ");
778     }
779 }
780

```

Liste des voitures

ID du Voiture	la Marque	le Nom	le colore	Num place	Prix jour	Enlocation
1	ford	ford	roug	10	300 DH	0
3	fait	fait_fait	bleu	35	270 DH	0
5	lamborghini	lamborghini	noir	41	400 DH	0

•Ajouter une voiture :

Cette fonction permet à écrire les informations de la voiture à ajouter dans le fichier voiture.

```

87
88 void AjouterVoiture(){
89     FILE* fichier = NULL;
90     fichier=fopen("voiture.txt", "a");
91     if(fichier==NULL){printf("\n\n\t\t\t Le fichier est Introuvable !");}
92
93     else{
94         voiture n;
95
96         printf("\n\n\n
97             LES Fmformation de la voiture a ajouter :\n\n");
98             *
99             *
100             *
101             *
102             *
103             *
104             *
105             *
106             *
107             *
108             *
109             *
110             *
111             *
112             *
113             *
114             *
115             *
116             *
117             *
118             *
119             *
120             *
121             *
122             *
123             *
124             *
125             *
126             *
127             *
128             *
129             *
130             *
131             *
132             *
133             *
134             *
135             *
136             *
137             *
138             *
139             *
140             *
141             *
142             *
143             *
144             *
145             *
146             *
147             *
148             *
149             *
150             *
151             *
152             *
153             *
154             *
155             *
156             *
157             *
158             *
159             *
160             *
161             *
162             *
163             *
164             *
165             *
166             *
167             *
168             *
169             *
170             *
171             *
172             *
173             *
174             *
175             *
176             *
177             *
178             *
179             *
180             *
181             *
182             *
183             *
184             *
185             *
186             *
187             *
188             *
189             *
190             *
191             *
192             *
193             *
194             *
195             *
196             *
197             *
198             *
199             *
200             *
201             *
202             *
203             *
204             *
205             *
206             *
207             *
208             *
209             *
210             *
211             *
212             *
213             *
214             *
215             *
216             *
217             *
218             *
219             *
220             *
221             *
222             *
223             *
224             *
225             *
226             *
227             *
228             *
229             *
230             *
231             *
232             *
233             *
234             *
235             *
236             *
237             *
238             *
239             *
240             *
241             *
242             *
243             *
244             *
245             *
246             *
247             *
248             *
249             *
250             *
251             *
252             *
253             *
254             *
255             *
256             *
257             *
258             *
259             *
260             *
261             *
262             *
263             *
264             *
265             *
266             *
267             *
268             *
269             *
270             *
271             *
272             *
273             *
274             *
275             *
276             *
277             *
278             *
279             *
280             *
281             *
282             *
283             *
284             *
285             *
286             *
287             *
288             *
289             *
290             *
291             *
292             *
293             *
294             *
295             *
296             *
297             *
298             *
299             *
300             *
301             *
302             *
303             *
304             *
305             *
306             *
307             *
308             *
309             *
310             *
311             *
312             *
313             *
314             *
315             *
316             *
317             *
318             *
319             *
320             *
321             *
322             *
323             *
324             *
325             *
326             *
327             *
328             *
329             *
330             *
331             *
332             *
333             *
334             *
335             *
336             *
337             *
338             *
339             *
340             *
341             *
342             *
343             *
344             *
345             *
346             *
347             *
348             *
349             *
350             *
351             *
352             *
353             *
354             *
355             *
356             *
357             *
358             *
359             *
360             *
361             *
362             *
363             *
364             *
365             *
366             *
367             *
368             *
369             *
370             *
371             *
372             *
373             *
374             *
375             *
376             *
377             *
378             *
379             *
380             *
381             *
382             *
383             *
384             *
385             *
386             *
387             *
388             *
389             *
390             *
391             *
392             *
393             *
394             *
395             *
396             *
397             *
398             *
399             *
400             *
401             *
402             *
403             *
404             *
405             *
406             *
407             *
408             *
409             *
410             *
411             *
412             *
413             *
414             *
415             *
416             *
417             *
418             *
419             *
420             *
421             *
422             *
423             *
424             *
425             *
426             *
427             *
428             *
429             *
430             *
431             *
432             *
433             *
434             *
435             *
436             *
437             *
438             *
439             *
440             *
441             *
442             *
443             *
444             *
445             *
446             *
447             *
448             *
449             *
450             *
451             *
452             *
453             *
454             *
455             *
456             *
457             *
458             *
459             *
460             *
461             *
462             *
463             *
464             *
465             *
466             *
467             *
468             *
469             *
470             *
471             *
472             *
473             *
474             *
475             *
476             *
477             *
478             *
479             *
480             *
481             *
482             *
483             *
484             *
485             *
486             *
487             *
488             *
489             *
490             *
491             *
492             *
493             *
494             *
495             *
496             *
497             *
498             *
499             *
500             *
501             *
502             *
503             *
504             *
505             *
506             *
507             *
508             *
509             *
510             *
511             *
512             *
513             *
514             *
515             *
516             *
517             *
518             *
519             *
520             *
521             *
522             *
523             *
524             *
525             *
526             *
527             *
528             *
529             *
530             *
531             *
532             *
533             *
534             *
535             *
536             *
537             *
538             *
539             *
540             *
541             *
542             *
543             *
544             *
545             *
546             *
547             *
548             *
549             *
550             *
551             *
552             *
553             *
554             *
555             *
556             *
557             *
558             *
559             *
560             *
561             *
562             *
563             *
564             *
565             *
566             *
567             *
568             *
569             *
570             *
571             *
572             *
573             *
574             *
575             *
576             *
577             *
578             *
579             *
580             *
581             *
582             *
583             *
584             *
585             *
586             *
587             *
588             *
589             *
590             *
591             *
592             *
593             *
594             *
595             *
596             *
597             *
598             *
599             *
600             *
601             *
602             *
603             *
604             *
605             *
606             *
607             *
608             *
609             *
610             *
611             *
612             *
613             *
614             *
615             *
616             *
617             *
618             *
619             *
620             *
621             *
622             *
623             *
624             *
625             *
626             *
627             *
628             *
629             *
630             *
631             *
632             *
633             *
634             *
635             *
636             *
637             *
638             *
639             *
640             *
641             *
642             *
643             *
644             *
645             *
646             *
647             *
648             *
649             *
650             *
651             *
652             *
653             *
654             *
655             *
656             *
657             *
658             *
659             *
660             *
661             *
662             *
663             *
664             *
665             *
666             *
667             *
668             *
669             *
670             *
671             *
672             *
673             *
674             *
675             *
676             *
677             *
678             *
679             *
680             *
681             *
682             *
683             *
684             *
685             *
686             *
687             *
688             *
689             *
690             *
691             *
692             *
693             *
694             *
695             *
696             *
697             *
698             *
699             *
700             *
701             *
702             *
703             *
704             *
705             *
706             *
707             *
708             *
709             *
710             *
711             *
712             *
713             *
714             *
715             *
716             *
717             *
718             *
719             *
720             *
721             *
722             *
723             *
724             *
725             *
726             *
727             *
728             *
729             *
730             *
731             *
732             *
733             *
734             *
735             *
736             *
737             *
738             *
739             *
740             *
741             *
742             *
743             *
744             *
745             *
746             *
747             *
748             *
749             *
750             *
751             *
752             *
753             *
754             *
755             *
756             *
757             *
758             *
759             *
760             *
761             *
762             *
763             *
764             *
765             *
766             *
767             *
768             *
769             *
770             *
771             *
772             *
773             *
774             *
775             *
776             *
777             *
778             *
779             *
780             *
781             *
782             *
783             *
784             *
785             *
786             *
787             *
788             *
789             *
790             *
791             *
792             *
793             *
794             *
795             *
796             *
797             *
798             *
799             *
800             *
801             *
802             *
803             *
804             *
805             *
806             *
807             *
808             *
809             *
810             *
811             *
812             *
813             *
814             *
815             *
816             *
817             *
818             *
819             *
820             *
821             *
822             *
823             *
824             *
825             *
826             *
827             *
828             *
829             *
830             *
831             *
832             *
833             *
834             *
835             *
836             *
837             *
838             *
839             *
840             *
841             *
842             *
843             *
844             *
845             *
846             *
847             *
848             *
849             *
850             *
851             *
852             *
853             *
854             *
855             *
856             *
857             *
858             *
859             *
860             *
861             *
862             *
863             *
864             *
865             *
866             *
867             *
868             *
869             *
870             *
871             *
872             *
873             *
874             *
875             *
876             *
877             *
878             *
879             *
880             *
881             *
882             *
883             *
884             *
885             *
886             *
887             *
888             *
889             *
890             *
891             *
892             *
893             *
894             *
895             *
896             *
897             *
898             *
899             *
900             *
901             *
902             *
903             *
904             *
905             *
906             *
907             *
908             *
909             *
910             *
911             *
912             *
913             *
914             *
915             *
916             *
917             *
918             *
919             *
920             *
921             *
922             *
923             *
924             *
925             *
926             *
927             *
928             *
929             *
930             *
931             *
932             *
933             *
934             *
935             *
936             *
937             *
938             *
939             *
940             *
941             *
942             *
943             *
944             *
945             *
946             *
947             *
948             *
949             *
950             *
951             *
952             *
953             *
954             *
955             *
956             *
957             *
958             *
959             *
960             *
961             *
962             *
963             *
964             *
965             *
966             *
967             *
968             *
969             *
970             *
971             *
972             *
973             *
974             *
975             *
976             *
977             *
978             *
979             *
980             *
981             *
982             *
983             *
984             *
985             *
986             *
987             *
988             *
989             *
990             *
991             *
992             *
993             *
994             *
995             *
996             *
997             *
998             *
999             *
1000            *

```

```

* * * * *
*         Donne ID du Voiture : 16
*             la Marque : X
*             le Nom : Y
*             la couleur : NOIR
*             Num plac : 12
*             le Prix : 12000
*             Enlocation : 0
* * * * *

```

la voiture est ajoute avec succes.

L'ID sera vérifié dans la fonction recherchev() pour éviter la répétition des IDs.

```

673
674 int rechercheV(int id ){
675     FILE* fichier = NULL;
676     fichier=fopen("voiture.txt", "r");
677     int i=0;  voiture n;
678
679     while(fscanf(fichier,"%d %s %s %s %d %d %d",&n.idVoiture,n.marque,n.nomVoiter,n.couleur,&n.nbplaces,&n.prixJour,&n.Enlocation)!=EOF){
680         .....
681         if(id==n.idVoiture)i++;
682     }
683
684     fclose(fichier);
685     return i;
686 }
687

```

- Modifier une voiture :

La modification des informations des voitures à partir de son ID

[illegible]

- Supprimer une voiture :

La suppression se fait partir d'un fichier intermédiaire. La boucle if à vérifier que l'ID tape existe, et que la voiture ne pas louer.

❖ Gestion des clients :

- Liste des clients :

Affiche les infos des clients.

[illegible]

- Ajouter un client :

Ajoute un client âpre passe par argument le nouveau ID du client à la fonction **recherchC()** pour garantir l'unicité d'ID

```

636 int AjouterClient(){
637     FILE* fichier = NULL;
638     fichier=fopen("clients.txt", "a");
639 
640     if(fichier==NULL){printf("\n\n\t\tle fichier est introuvable !");}
641 
642     else{                client C;
643         jc:
        printf("\n\n\n                                LES information de le client a Ajouter :\n\n\n");
        printf(" * * * * * \n");
        printf("                               Donne ID client : ");
646         scanf("%d",&C.idClient);
647         if(rechercheC(C.idClient)!=0){system("cls");printf("\n\n\t\t\t ID est deja stocke   tape un autre:");goto jc;}//stoke
648     }
649     else{
650         printf(" *                            le Nom : ");
651         scanf("%s",C.nom);
652         printf(" *                            le Prenom : ");
653         scanf("%s",C.prenom );
654         printf(" *                            CIN : ");
655         scanf("%d",&C.cin);
656         printf(" *                            La Adresse : ");
657         scanf("%s",C.adresse);
658         printf(" *                            Telephone : +212 ");
659         scanf("%d",&C.telephone);
660         printf(" * * * * * \n");
661         printf("                             le client este Ajoute avec succes.\n");
662         fprintf(fichier,"%d %s %s %d %s %d\n",C.idClient,C.nom,C.prenom,C.cin,C.adresse,C.telephone);
663     }
664 
665     return C.idClient;    fclose(fichier);    }

```

La fonction retourne un entier en aura besoin à la location.

- Modifier un client :

[illegible]

- Supprimer un client :

[illegible]

❖ Location :

- Visualiser les contrats :

Pour objectif d'afficher un contrat.

[illegible]

Le rôle de cette fonction est de créer un contrat avec le client.

[illegible]

- La fonction **id contrat()** vérifié que la ID est ne pas déjà stocke.
- La fonction **VNP()** vérifie le nom et le prénom du client, et il récupéré son ID.
- **VD()** Il vérifier que la voiture souhaite par le client existe.ET il renvoi son ID, sa place ainsi que le prix de la

location par jour.

```

37/
738 int VD(char M[15],char N[15],char C[7],int *e , int *idv , int *prix,int *place){
739     FILE* fichier = NULL;
740     fichier=fopen("voiture.txt", "r");
741     int i=0;   voiture n;
742
743     while(fscanf(fichier,"%d %s %s %s %d %d %d",&n.idVoiture,n.marque,n.nomVoiter,n.couleur,&n.nbplaces,&n.prixJour,&n.Enlocation)!=EOF){
744
745         if(strcmp(M,n.marque)==0 && strcmp(N,n.nomVoiter)==0 && strcmp(C,n.couleur)==0 ){
746             i++;      *e=n.Enlocation;      *idv=n.idVoiture;      *prix=n.prixJour;*place=n.nbplaces;
747         }
748     }
749
750     fclose(fichier);
751     return i;
752 }
753
754

```

- La disponibilité de la voiture, se fait par la comparaison de la variable « e » avec "1".
- La possibilité d'afficher les voitures disponibles via la fonction **disponible()**.

[illegible]

- **Retourner louer Voiture()** cette fonction à deux rôles.
- 1) Il change l'avaleur de l'attribut En location on "1", pour rendre la voiture louer.
 - 2) Il change l'avaleur de l'attribut En location on "0", si la voiture rendu.

Pour clore j'ai rencontré des problèmes sur le côté technique en particulier en ce qui concerne l'installation des logiciels GTK+,QT j'ai essayé de les intégrer dans codeblocks mais en vain