



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Webový informa ní systém pro asistenci rekrea ním lodím
Student:	Markéta Veverková
Vedoucí:	Ing. Vojt ch Jirkovský
Studijní program:	Informatika
Studijní obor:	Web a multimédia
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

Cílem práce je vytvo it systém pro podporu asistence rekrea ním lodím pro p ípad poruchy.

1. Prove te rešerši stávajícího ešení p íp. dalších existujících asisten ních systém
2. Zpracujte analýzu uživatelských požadavk
3. Na základ rešerše a analýzy navrh n te vlastní ešení
4. Návrh implementujte jako webovou aplikaci v PHP frameworku Symfony, která umožní
 - správu klient , kontrakt , asisten ních a technických kontakt v etn zobrazení na map
 - sledovat celý proces asistence
5. Implementaci otestujte z hlediska funk nosti a použitelnosti

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Ji ina, Ph.D.
d kan

V Praze dne 3. listopadu 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Webový informační systém pro asistenci rekreačním lodím

Markéta Veverková

Vedoucí práce: Ing. Vojtěch Jirkovský

29. června 2018

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 29. června 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Markéta Veverková. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Veverková, Markéta. *Webový informační systém pro asistenci rekreačním lodím*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Cílem této bakalářské práce je implementace nové verze již existujícího webového informačního systému Code0. Tento systém je používán francouzskou společností Marine Assistance, jež se specializuje na asistenci rekreačním lodím. Výstupem je webová aplikace, nová verze tohoto informačního systému, implementovaná v PHP frameworku Symfony.

Hlavním přínosem této práce je fakt, že nově implementovaný informační systém je, na rozdíl od jeho staré verze, rozšiřitelný o nové funkce a zároveň má intuitivnější uživatelské rozhraní.

Klíčová slova informační systém, webová aplikace, asistence rekreačním lodím, Symfony framework, PHP

Abstract

The main goal of this Bachelor thesis is to implement a new version of an existing web information system called Code0. This system is used by Marine Assistance, which is a French company specialised in boating assistance. The output of this thesis is a new web information system written in Symfony PHP framework.

The main benefit of the thesis is the fact that the newly implemented information system can, contrary to the former version, be extended by new functions, and it has a more intuitive user interface.

Keywords information system, web application, boating assistance, Symfony framework, PHP

Obsah

Úvod	1
1 Cíle práce	3
1.1 Cíle rešeršní části	3
1.2 Cíle praktické části	3
2 Rozbor stávajícího řešení	5
2.1 Představení systému Code0	5
2.2 Architektura a použité technologie	6
2.3 Datový model	7
2.4 Uživatelské rozhraní	10
2.5 Problémy stávajícího systému	12
2.6 Zdrojový kód	13
3 Analýza uživatelských požadavků	21
3.1 Funkční požadavky	21
3.2 Nefunkční požadavky	22
4 Návrh řešení	23
4.1 Metodika řízení	23
4.2 Datový model	23
4.3 Uživatelské rozhraní	24
5 Implementace	31
5.1 Symfony	31
5.2 Doctrine	31
5.3 Bezpečnost	32
5.4 Twilio	33
5.5 Symfony Workflow	33

6 Testování	35
6.1 Jednotkové a funkční testy	35
6.2 Splnění funkčních požadavků	35
6.3 Testování použitelnosti	36
Závěr	43
Literatura	45
A Seznam použitých zkratk	47
B Výstupní formulář	49
C Obsah přiloženého CD	51

Seznam obrázků

2.1	Průběh asistence	7
2.2	Adresářová struktura původní aplikace	8
2.3	Datový model	16
2.4	Administrace - přidání klienta	17
2.5	Administrace - editace klienta	17
2.6	Asistence - první krok	18
2.7	Asistence - intervence	19
4.1	Datový model	26
4.2	Wireframe administrace - detail klienta	27
4.3	Wireframe administrace - editace klienta	28
4.4	Wireframe asistence - první krok	29
4.5	Wireframe asistence - intervence	30
6.1	Agregované výsledky výstupních dotazníků pro nový Code0	41
6.2	Agregované výsledky výstupních dotazníků pro původní Code0	41

Úvod

Informační systémy jsou v dnešní době ve většině firem téměř nepostradatelné. Ať už se jedná o velké korporace, či střední a malé podniky, výhodou oproti uchovávání informací v klasické papírové formě je nespočet. IS usnadňuje vyhledávání a orientaci v datech, zároveň je možné s daty dále rychle a efektivně pracovat. Důležitá je však kvalita samotného IS. Pokud je daný IS od svého počátku špatně navržen (ať už se jedná o architekturu systému, databázový model či uživatelské rozhraní), míra efektivnosti osob s ním pracujících se razantně snižuje. Zároveň je u takového systému velmi náročná údržba a není snadné jej případně rozšířit o další funkce.

Tato práce se bude zabývat IS s názvem Code0. Tento systém je používán francouzskou společností Marine Assistance, specializující se na asistenci rekreačním lodím. Společnost ho využívá k evidenci svých klientů, jejich kontraktů a proběhlých asistencí. Zároveň je IS využíván k uchovávání informací a kontaktů na technické služby a specialisty. Díky těmto kontaktům je firma schopna poskytovat asistenci lodím až do 500 km od francouzského pobřeží.

Motivací ke zvolení tohoto tématu jsem získala při své 4měsíční stáži v již zmíněné společnosti Marine Assistance. Během této doby jsem měla možnost se s daným IS seznámit. Díky zkušenostem jeho uživatelů a mne samotné jsem byla schopná vyhodnotit, že vytvořením nové verze tohoto systému by společnost získala IS s jednodušším uživatelským rozhraním, který by byl zároveň rozšiřitelný o nové funkce. Prospěšnost mé práce bude tedy hlavně ve zvýšení efektivnosti zaměstnanců zmiňované společnosti.

Jak již bylo naznačeno v předchozích odstavcích, práce se bude zabývat analýzou stávajícího systému, následným návrhem nové verze a její implementací v PHP frameworku Symfony. V první části detailněji popíšu zadání a všechny stanovené cíle práce. V rámci následující části představím IS Code0 a rozeberu jeho stávající řešení, včetně technologií, architektury, databázového modelu a uživatelského rozhraní systému. Třetí část bude věnována analýze uživatelských požadavků. Následovat budou kapitoly věnované návrhu řešení, implementaci a závěrečnému testování nové verze informačního systému.

Důležité je poznamenat, že Code0 se skládá z několika částí:

- **asistenční**, obsahující veškeré funkcionality týkající se samotného průběhu asistence,
- **administrativní**, zahrnující správu klientů, jejich kontraktů, partnerů a technických kontaktů,
- **účetnická**, týkající se správy klientských kontraktů z hlediska financí, exportování dat o klientech do účetních programů a generování faktur,
- **klientská**, poskytující rozhraní klientům pro vyplňování anket spokojenosti.

V rámci této práce se budu zabývat pouze prvními dvěma zmíněnými, tedy asistenční a administrativní částí. Pro účely této práce budeme předpokládat, že IS Code0 obsahuje pouze tyto dvě části, zbylé dvě budou zmiňovány jen velmi okrajově.

Cíle práce

Cíle řešeršní i praktické části se tedy, jak již bylo zmíněno v úvodu, týkají pouze asistenční a administrativní části systému a jsou představeny v následujících odstavcích.

1.1 Cíle řešeršní části

Prvním z cílů řešeršní části je seznámení s již existujícím informačním systémem Code0. Jeho analýza je totiž nezbytná k odhalení chyb, kterých je potřeba se vyvarovat při tvorbě nové verze tohoto systému. Zároveň je též potřeba provést analýzu uživatelských požadavků.

1.2 Cíle praktické části

Praktická část má za cíl návrh nového informačního systému splňujícího uživatelské požadavky definované v řešeršní části. Tento IS bude implementován jako webová aplikace v PHP frameworku Symfony 4. Dalším cílem je samotná implementace a pokrytí automatizovanými testy. Aplikace bude též otestována z hlediska použitelnosti.

Rozbor stávajícího řešení

Následující kapitola obsahuje detailní analýzu systému Code0, jeho představení, analýzu použitých technologií při tvorbě tohoto systému, datový model, uživatelské rozhraní a rozbor nedostatků současného řešení.

2.1 Představení systému Code0

Code0 je informační systém francouzské společnosti Marine Assistance. Ta se již několik let specializuje v oblasti asistence rekreačním lodím. V praxi tato asistence probíhá tak, že klient, který má aktivní kontrakt, společnost v případě poruchy lodi telefonicky kontaktuje.

Po telefonickém kontaktu se zákazníkem operátor v systému ověří, že má zákazník nárok na asistenci (tj. platný kontrakt). Pokud ano, začíná vlastní proces asistence. Ten sestává z identifikace závady a jejího případného vyřešení, či zavolání technika. Celým tímto procesem informační systém operátora provádí.

Je tedy důležité, aby v sobě systém uchovával informace o klientech a jejich kontraktech. Zároveň je systém využíván k ukládání informací o partnerských společnostech, které poskytují technické kontakty. V neposlední řadě také slouží k zobrazení všech dostupných kontaktů, což napomáhá k jednoduššímu nalezení technické pomoci co nejbližší klientovi.

2.1.1 Účastníci

V popisu procesů probíhajících v systému Code0 se vyskytuje několik účastníků, kteří však nemusí být přímo uživateli systému. Pro větší přehlednost jsou účastníci popsáni v následujícím seznamu.

Uživatel, operátor

uživatel systému Code0, zároveň operátor asistence, komunikující se zákazníkem, jediný účastník přímo používající IS,

klient, zákazník

osoba, která má u Marine Assistance zakoupený produkt zajišťující asistenci v případě poškození lodi,

partner, technik

osoba či firma, jež má s Marine Assistance uzavřenou smlouvu o poskytování asistenčních služeb.

2.1.2 Asistence

Proces asistence je hlavním procesem, který v informačním systému probíhá. Začíná kontaktováním asistenčního centra, což je možné dvěma způsoby - buď telefonickým kontaktem, nebo pomocí specializované mobilní aplikace, která do asistenčního centra odešle e-mail se žádostí o asistenci a souřadnicemi klienta. Po této žádosti operátor asistenčního centra klienta telefonicky kontaktuje.

Po zjištění základních informací od klienta (typicky jméno, současná poloha a závada) přichází fáze interakce operátora asistenčního centra s informačním systémem. Tato fáze je znázorněna na diagramu 2.1.

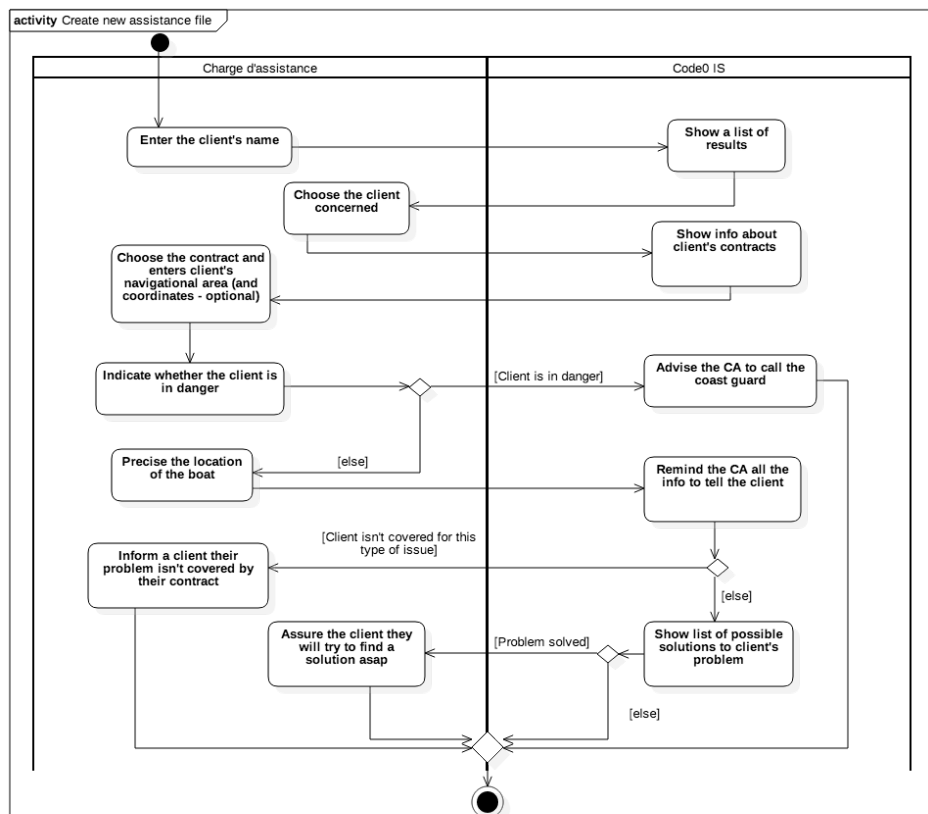
2.2 Architektura a použité technologie

V současné podobě systém není objektově orientovaný, neobsahuje tedy žádné objekty ani šablony. Tvoří jej několik stovek php souborů, které nejsou výrazně strukturované. Adresářová struktura je následující 2.2. Z adresářové struktury je vidět, že projekt působí velmi neorganizovaným dojmem, je velmi těžké se v něm orientovat. Zároveň také obsahuje soubory, které už se nepoužívají (např. `sms.php.old`), což znovu pouze přispívá k celkové chaotičnosti. Samotný systém působí tak, že se kladl důraz spíše na rychlost vytvoření aplikace, než na samotný návrh a případnou použitelnost, údržbu a rozšiřitelnost do budoucna.

2.2.1 PHP

Aplikace je implementovaná v PHP verze 5.3, jež byla jednou z nejpopulárnějších verzí tohoto jazyka [1]. Od roku 2014 je však tato verze na seznamu nepodporovaných [2].

Aplikace používá původní funkce MySQL s předponou `mysql_`. Tyto funkce byly od verze PHP 5.5.0 označeny jako zastaralé, jejich používání způsobovalo chybu `E_DEPRECATED`. Všechny funkce byly odstraněny s novou verzí PHP 7.0.0. [3]



Obrázek 2.1: Průběh asistence

2.3 Datový model

Původní databáze obsahuje celkem 33 entit. Tyto entity mezi sebou nemají žádné relace, veškerá režie týkající se relací mezi jednotlivými objekty se odehrává v aplikaci. Všechny entity jsou zobrazeny v diagramu 2.3.

Chybějící relace zapříčiňují redundanci dat. Například entita reprezentující uživatele je v každé entitě referencována uživatelským jménem. Toto jméno je sice unikátní, pokud by však uživatel chtěl své uživatelské jméno změnit, tato změna by zasáhla několik set řádek v několika různých tabulkách.

Dalším problémem současné databáze je občas velmi nejasné pojmenování jednotlivých entit a jejich atributů. Zároveň je zde velká nekonzistence v názvech primárních klíčů. Vzhledem k tomu, že aplikace není dostatečně zdokumentována, bylo velmi časově náročné celou databázi zmapovat. Jednotlivé entity, jejich primární klíče a popis jsou v následujícím seznamu:

- **assistance (assist_no)** asistenční událost,

2. ROZBOR STÁVAJÍCÍHO ŘEŠENÍ

config	přihlašovací údaje do databáze a ostatní konfigurace
font	sada různých fontů
img	ikony používané v aplikaci
inc	sada funkcí (asistence, mapy, generování menu...)
json	komponenta exportující data ve formátu json
navi	mapy navionics
page_def	mapování všech souborů do pole \$config
std_css	obsahuje 1 CSS šablonu
std_fct	sada funkcí (datum, sql funkce, popup okna)
std_img	ikony používané v aplikaci
assist_content.php	definice jednotlivých kroků asistence
assist_hidden.php	javascriptové funkce pro podporu průběhu asistence
assist_lst_content.php	
assist_lst_hidden.php	zjišťování stavu asistence pro otevření ve správném kroku
cust_hidden.php	
export_content.php	získávání dat z databáze pro export
export_content_bck.php	javascriptové funkce pro frontendovou část exportu
export_hidden.php	notifikace o průběhu exportu
fpdf.php	funkce pro generování pdf
fpdf_cellfit.php	funkce pro generování pdf
index.php	
OvhApi.php	API pro posílání SMS
sms.php	komponenta integrující OVH API do systému
sms.php.old	stará verze komponenty OVH API
std_blank.php	prázdné HTML tagy
std_footer.php	zápatí stránky
std_header.php	záhlaví stránky
std_log_hidden.php	část komponenty pro ukládání log zpráv do databáze
std_login.php	přihlašovací formulář
std_main.php	hlavní stránka
std_main_content.php	
std_main_fct.php	funkce validace formulářů, jejich odesílání, exportování dat, komunikace s databází a spousta dalších funkcí
std_main_hidden.php	notifikace nevalidních formulářů
std_menu.php	základ hlavního menu
std_post.php	routování po odesílání formulářů
std_support_content.php	include souboru std_support.php, pokud existuje
UPDATE_THOMAS_10-2014import_hidden.php	
zzz_update_empty.php	změna některých dat v databázi
zzz_update_mars_2013.php	změna některých dat v databázi

- **assistance_carte_conf (assist_carte_no)** neobsahuje žádná data a v celé aplikaci se nepoužívá, pravděpodobně byla přidána pro konfiguraci mapy navionics, tato možnost však v systému nebyla nikdy implementována,
- **assistance_point_gps (assist_pg_no)** v původním systému byla možnost ukládat souřadnice zvoleného bodu na mapě, tato funkce se však téměř nepoužívala, proto nebude v novém systému implementována,
- **carte_navionics_public_port (port_no)** entita představující přístav,
- **carte_navionics_public_urg (cnp_no)** kontakty na jednotlivé pohotovostní jednotky,
- **client (cust_no)** entita reprezentující klienta a informace o něm,
- **client_contrat (cc_no)** záznam o klientových kontraktech,
- **localisation_user (lu_usr_code)** aplikace čas od času uloží url, na které se uživatel nachází, tato informace se dále v aplikaci nepoužívá,
- **lst_port (port_no)** redundantní entita k entitě **carte_navionics_public_port**,
- **lst_type_bateau (tboat_no)** typ lodi,
- **lst_type_contact (tcon_no)** typ technického partnera, obsahuje však jen jeden záznam a v aplikaci není možné žádný přidat,
- **lst_type_materiaux (tmat_no)** typ materiálu lodi,
- **lst_type_moteur (tmot_no)** typ motoru,
- **lst_type_part (tpart_no)** typ partnerského kontraktu,
- **lst_zone_nav (zone_no)** navigační zóna (region, department, ...),
- **partenaire (part_no)** entita reprezentující technického partnera,
- **partenaire_zone (pz_no)** navigační zóna, ve které daný partner operuje,
- **produit (prod_no)** entita představující produkt,
- **std_log (log_no)** log záznam, poznámka k asistenci, záznam o odeslaném e-mailu nebo SMS,
- **std_log_config (logcfg_module)** entita obsahující SQL dotazy s JOIN příkazy, reprezentující relace mezi tabulkami **std_log** a **client/assistance/partenaire**,
- **std_log_pj (pj_no)** cesta k souboru připojenému k entitě **std_log**; stejná informace je však uložena v atributu **log_file** v entitě **std_log**,

2. ROZBOR STÁVAJÍCÍHO ŘEŠENÍ

- **std_lst_centry (centry_no)** entita reprezentující zemi,
- **std_lst_lang (lang_code)** entita obsahující jedinou informaci o překladu francouzského jazyka do angličtiny a španělštiny; v systému se nevyužívá,
- **std_lst_niveau (part_niveau)** obsahuje tři údaje o kvalitě služeb jednotlivých partnerů,
- **std_lst_part_type (part_type_id)** typ technického partnera,
- **std_lst_pj (pj_no)** typ ukládaného souboru, který se však dá zjistit z přípony souboru,
- **std_lst_prov (prov_no)** provincie, regiony,
- **std_lst_tinyint_val (tval_val)** obsahuje hodnoty *yes* a *no*, které jsou občas používány ve formulářích,
- **std_lst_type_phone (tphone_no)** typ telefonického kontaktu,
- **std_menu (menu_no)** záložky navigačního menu,
- **std_sec_right (secr_usr_code)** nepoužívané navigační menu,
- **std_sec_screen (scr_code)** podzáložky navigačního menu,
- **std_sec_usr (usr_code)** uživatelský účet.

2.4 Uživatelské rozhraní

V současné chvíli pro firmu uživatelské rozhraní nepředstavuje velký problém, jelikož všichni zaměstnanci již byli proškoleni a jsou zvyklí jej používat. Problémem je však velká neintuitivita současného uživatelského rozhraní, což výrazně ztěžuje proces školení nových zaměstnanců.

2.4.1 Administrace

Administrativní část slouží k práci s veškerými daty, kterými informační systém disponuje. Je zde možné prohlížet, přidávat, editovat a mazat údaje o uživateli, technických partnerech, produktech atd. V následujících částech bude popsáno uživatelské rozhraní pro přidání a editaci klientů. Obdobné UI je použito i pro editaci ostatních entit.

2.4.1.1 Přidání klienta

Při vstupu do sekce Klient se zobrazí formulář (obr. 2.4), pomocí kterého je možné nového klienta přidat. Nad formulářem se nachází tlačítka, z nichž ne všechna souvisí se samotným formulářem:

Nouveau

tlačítko pro otevření formuláře pro přidání nového klienta, ve kterém se však zrovna nacházíme,

Enregistrer

uložit údaje klienta,

Supprimer

smazat klienta, což je však při vytváření nového klienta nemožné, jelikož ještě nebyl uložen,

Rechercher

tlačítko pro vyhledávání, vedoucí k formuláři pro vyhledávání klientů, které však nesouvisí s procesem přidávání nového klienta,

Résultat

vede k vypsání všech klientů, též nesouvisející s přidáváním nového klienta.

2.4.1.2 Editace údajů klienta

Po výběru klienta (či po uložení nového klienta) se dostaneme na stránku (obr. 2.5), jež se celkem neliší od té při vytváření nového klienta. Z hlavičky stránky to není jasné, ale jedná se o detail klienta a editační formulář zároveň. Stejně jako u přidávání klienta v předchozí sekci 2.4.1.1 část tlačítek s editací nesouvisí.

Ve formuláři s údaji o klientovi se také nachází textový vstup s popisem ID automatique, jenž by mohl uživatele vést k domněnání, že může nastavit klientovo (automatické) ID, které se používá jako primární klíč v databázi. Po změně hodnoty tohoto políčka se však vyhledá klient s odpovídajícím identifikátorem, či aplikace vypíše chybu, že daný klient neexistuje. Toto chování aplikace je velmi nestandardní a pro nového uživatele systému může být matoucí.

Kromě informací o klientovi lze také zobrazit ostatní informace, které jsou rozděleny do třech záložek:

Client

záložka s formulářem klienta,

Produit

seznam s historií produktů klienta,

Suivi

seznam všech záznamů týkajících se klienta (komentáře k asistencím, SMS zprávy, e-maily atd.).

2.4.2 Asistence

Nejednoznačnosti v části asistence mohou vést k prodlužování celkového času věnovaného interakci operátora asistence se systémem. Času, který by měl být věnován klientovi, či hledání řešení jeho problému.

2.4.2.1 První kroky asistence

Prvních pět kroků asistence sestává z formuláře a pomocných informací k němu. Na obrázku 2.6 se nachází první krok asistence, ve kterém uživatel vybírá kontrakt klienta a zároveň se jej snaží lokalizovat. Ve většině případů klient zná díky palubnímu počítači své přesné souřadnice, není tomu však vždy. Pokud klient své souřadnice nezná, je operátor nucen souřadnice obstarat (otevření nové záložky s mapou, hledání souřadnic dle klientova popisu místa, na kterém se přibližně nachází).

V následujících krocích se systém ptá uživatele, zda je klient na moři či v přístavu, zda je v nebezpečí, jaká je přesná závada na jeho lodi atd. Tento proces je detailněji popsán v diagramu 2.1. Hlavním problémem uživatelského rozhraní v této části je celková nepřehlednost jednotlivých kroků. Oddělením formulářové a informační části by se radikálně zvýšila orientace uživatele v rámci každého kroku.

2.4.2.2 Intervence

Uživatelské rozhraní tohoto kroku asistence je jedním z nejnepřehlednějších. Do této části byla v minulosti přidána možnost odesílat SMS, která byla umístěna do jediného volného místa na obrazovce bez větší logiky a smyslu. Zároveň jsou informace o klientovi umístěny na úplném konci stránky. To se může jevit jako velký problém hlavně v případě, kdy chce operátor klientovi odeslat SMS. Pro tuto akci je nutné dorolovat na konec stránky, ručně zkopírovat klientovo telefonní číslo, vrátit se na začátek stránky a vložit ho do políčka pro příjemce.

2.5 Problémy stávajícího systému

V následující sekci jsou představeny nejzásadnější problémy současného systému, které zároveň slouží i jako kritéria pro pozdější zhodnocení a srovnání nové verze systému.

2.5.1 Dokumentace

Jedním z hlavních problémů současné verze systému je zcela chybějící dokumentace. Tento fakt velmi ztížil analýzu současného řešení a orientaci ve zdrojovém kódu.

2.5.2 Rozšiřitelnost

Dalším zásadním problémem je fakt, že současný systém je téměř nemožné rozšířit o nové funkce. Vzhledem k chybějící dokumentaci, nejasné architektuře systému a chybějícím automatizovaným testům by možná rozšíření bylo velmi náročné navrhnout, natož implementovat a otestovat.

2.6 Zdrojový kód

Jako problém se jeví i samotný styl psaní kódu. Ten je totiž v rámci celého projektu velmi nekonzistentní a jen přispívá k již celkem obtížné orientaci v projektu.

Aspektem, který velmi snižuje možnost orientace v rámci projektu i v rámci jednotlivých funkcí, je především kombinování několika jazyků v jednom souboru. Většina souborů v projektu totiž obsahuje PHP používající SQL, zároveň generující HTML, JavaScript i CSS, což je ve většině případů velmi matoucí a znesnadňuje práci.

2.6.1 Jazyk

V rámci zdrojového kódu je používána angličtina a francouzština. Všechny komentáře jsou psané ve francouzštině, u většiny kódu to však takto jednoznačné není. Názvy funkcí jsou většinou v angličtině, občas se v nich vyskytují zkratky z francouzštiny. U názvů proměnných je však velmi nesourodě používána jak francouzština, tak angličtina, občas jsou oba jazyky v rámci jednoho názvu proměnné dokonce kombinovány. [4]

V následující ukázce kódu můžeme vidět hned několik nedostatků:

- kombinování francouzštiny a angličtiny při pojmenovávání proměnných a v komentářích (`bloc_select`, `Load les données existantes`),
- nejednoznačné pojmenování proměnných (`bloc_lst_fid`),
- kombinování jazyků (PHP, JavaScript, SQL, HTML, CSS, vše na 50 řádcích kódu).

```
1 <?php
2 ...
3 //Load les données existantes
```

2. ROZBOR STÁVAJÍCÍHO ŘEŠENÍ

```
4  $sql = "SELECT * FROM ".$config["bloc_select"][$stab][$bloc]. "
      WHERE ".$config["bloc_id"][$stab][$bloc]. " = '".
      mysql_real_escape_string($get_data["var_id"])."'";
5  $res = mysql_query($sql);
6  if (mysql_num_rows($res) > 0){
7      while($row = mysql_fetch_array($res)){
8          for($i=0;$i<count($data);$i++){
9              if($row[$config["bloc_lst_fid"][$stab][$bloc]] == $data[$i]
10                 [$config["bloc_lst_fid"][$stab][$bloc]]){
11                  $data[$i][$config["bloc_lst_val"][$stab][$bloc]] = $row[
12                     $config["bloc_lst_val"][$stab][$bloc]];
13                  break;
14              }
15          }
16      }
17  }
18  ?>
19  <script language="JavaScript" type="text/javascript">
20  function opt_select_all(sel_value){
21      for(i=0;i<document.getElementsByTagName('input').length;i++){
22          if (document.getElementsByTagName('input')[i].type == '
23              radio'){
24              if(document.getElementsByTagName('input')[i].value==
25                 sel_value){
26                  document.getElementsByTagName('input')[i].checked=true;
27              }
28          }
29      }
30  }
31  </script>
32  <?
33  $head = explode(";", $config["bloc_head"][$stab][$bloc]);
34  $opt = explode(";", $config["bloc_opt_val"][$stab][$bloc]);
35  $opt_counter = 0;
36
37  echo "<table class=\"std_search_table\">\n";
38  echo "<tr class=\"std_search_title\">\n";
39  for($i=0;$i<count($head);$i=$i+2){
40      echo "<td";
41      if($head[$i+1]!="&& $head[$i+1]!=0){
42          echo " style=\"width: ".$head[$i+1]. "px;\"";
43      }
44      echo ">";
45      if($i!=0){
46          $opt_counter++;
47          echo "<a href=\"javascript:void(0);\" onClick=\"javascript:
48              opt_select_all('".$opt[$opt_counter-1]."' )\" style=\"
49                  color:#ffffff;\">".translate($head[$i], $_SESSION["
50                  lang_code"])."</a>";
51      }else{
52          echo translate($head[$i], $_SESSION["lang_code"]);
53      }
54      echo "</td>\n";
55  }
```

```
49 }  
50 echo "</tr>\n";  
51 ...
```

Listing 2.1: Ukázka kombinování jazyků ve zdrojovém kódu

2. ROZBOR STÁVAJÍCÍHO ŘEŠENÍ

assistance assist_no INT(11) assist_step INT(11) assist_lact_step INT(11) assist_cust_no INT(11) assist_cc_no INT(11) assist_lang DOUBLE assist_lact DOUBLE assist_zone_no INT(11) assist_repetre TEXT assist_danger TinyINT(4) assist_lien VARCHAR(10) assist_couverture TinyINT(4) assist_panne_resolue TinyINT(4) assist_part_no INT(11) assist_motif_panne TEXT assist_ref_assurance VARCHAR(60) assist_note TEXT assist_is_rappel INT(11)	carte_navigons_public_urg cmp_no INT(11) cmp_type VARCHAR(10) cmp_name VARCHAR(100) cmp_addr VARCHAR(100) cmp_tel VARCHAR(40) cmp_lien VARCHAR(100) cmp_lang VARCHAR(10) cmp_lng VARCHAR(10) cmp_mtd VARCHAR(1000)	client cust_no INT(11) cc_code VARCHAR(25) cust_ref_no VARCHAR(25) cust_part_no INT(11) cust_firstname VARCHAR(80) cust_name VARCHAR(80) cust_addr1 VARCHAR(80) cust_addr2 VARCHAR(80) cust_cpy VARCHAR(80) cust_cnty_no INT(11) cust_phone1 VARCHAR(25) cust_phone2 VARCHAR(25) cust_phone3 VARCHAR(25) cust_phone_no_1 INT(11) cust_phone_no_2 INT(11) cust_phone_no_3 INT(11)	client_contrat cc_no INT(11) cc_code VARCHAR(25) cc_ref VARCHAR(50) cc_lact_no INT(11) cc_cust_no INT(11) cc_prod_no_1 INT(11) cc_prod_no_2 INT(11) cc_date_start DATE cc_date_end DATE cc_do_not_renew TinyINT(4) cc_boal_name VARCHAR(80) cc_boal_mod VARCHAR(80) cc_boal_modelle VARCHAR(80) cc_boal_boal_no INT(11) cc_boal_fmtd_no INT(11) cc_boal_fmtd_no INT(11) cc_boal_fmtd_no INT(11) cc_boal_fmtd_no INT(11) cc_boal_fmtd_brand VARCHAR(10) cc_boal_fmtd_pwr VARCHAR(10)	ist_type_moteur mtod_no INT(11) mtod_desc_fr VARCHAR(50) mtod_date INT(11)	ist_type_bateau tboat_no INT(11) tboat_desc_fr VARCHAR(100)	ist_type_contact tcon_no INT(11) tcon_desc_fr VARCHAR(50)	ist_type_materiaux mmat_no INT(11) mmat_desc_fr VARCHAR(50)	ist_type_part part_no INT(11) part_desc_fr VARCHAR(80)	ist_type_part_type part_type_id INT(10) part_type_name VARCHAR(100)	ist_lang lang_code VARCHAR(2) lang_name_fr VARCHAR(100) lang_name_en VARCHAR(100) lang_name_sp VARCHAR(100)	ist_log_config logcfg_module VARCHAR(100) logcfg_table VARCHAR(200) logcfg_id VARCHAR(50) logcfg_phone_field VARCHAR(100) logcfg_iphone_field VARCHAR(100)	ist_log_pi pi_no INT(11) pi_no INT(11) chemin_pi VARCHAR(100)	ist_prov prov_no INT(11) prov_cnty_no INT(11) prov_name_fr VARCHAR(80) prov_name_en VARCHAR(80) prov_name_sp VARCHAR(80)	ist_tinynt_val tvai_val VARCHAR(1) tvai_lbl_fr VARCHAR(10) tvai_lbl_en VARCHAR(10)	ist_sec_screen scr_code VARCHAR(10) scr_desc_fr VARCHAR(80) scr_desc_en VARCHAR(80) scr_desc_sp VARCHAR(80)	ist_port port_no INT(11) port_name VARCHAR(50) port_cnty_no INT(11) port_prov_no INT(11)	ist_cnty cnty_no INT(11) cnty_name_fr VARCHAR(80) cnty_name_en VARCHAR(80) cnty_name_sp VARCHAR(80)	ist_sec_right secr_usr_code VARCHAR(25) secr_sec_code VARCHAR(10) secr_right INT(11)	partenaire_zone pz_no INT(11) pz_part_no INT(11) pz_zone_no INT(11)	assistance_carre_conf assist_carre_no INT(11) assist_carre_desc_fr VARCHAR(20) assist_carre_navigons VARCHAR(6)	assistance_point_gps assist_pg_no INT(11) assist_pg_lat VARCHAR(15) assist_pg_lng VARCHAR(15) assist_pg_date DATE assist_pg_heure VARCHAR(10) assist_pg_name VARCHAR(20) assist_pg_ordre INT(11)	std_log log_no INT(11) log_module VARCHAR(25) log_type VARCHAR(25) log_id VARCHAR(25) log_title VARCHAR(255) log_text TEXT log_save_date DATE log_save_closed TinyINT(4) log_save_usr_code VARCHAR(25) log_email_to VARCHAR(255) log_email_from VARCHAR(255) log_email_to_name VARCHAR(255) log_email_from_name VARCHAR(255)	std_list_pi id_pi INT(11) name_pi VARCHAR(20)	std_list_niveau part_niveau INT(10) part_niveau_name VARCHAR(20)	std_list_type_phone iphone_no INT(11) iphone_desc_fr VARCHAR(50) iphone_desc_en VARCHAR(50) iphone_desc_sp VARCHAR(50)	std_list_type_phone iphone_desc_fr VARCHAR(50) iphone_desc_en VARCHAR(50) iphone_desc_sp VARCHAR(50)	std_sec_usr usr_code VARCHAR(25) usr_lusr_no INT(11) usr_pass VARCHAR(255) usr_name VARCHAR(80) usr_firstname VARCHAR(80) usr_lastname VARCHAR(80) usr_city VARCHAR(80) usr_addr1 VARCHAR(80) usr_addr2 VARCHAR(80) usr_prov_no INT(11) usr_cnt_no INT(11) usr_phone_no_1 INT(11) usr_phone_no_2 INT(11) usr_phone_no_3 INT(11) usr_email VARCHAR(255) usr_email VARCHAR(255) usr_lang_code VARCHAR(2) usr_css VARCHAR(25)	std_log_contig logcfg_module VARCHAR(100) logcfg_table VARCHAR(200) logcfg_id VARCHAR(50) logcfg_phone_field VARCHAR(100) logcfg_iphone_field VARCHAR(100)	std_log_pi pi_no INT(11) pi_no INT(11) chemin_pi VARCHAR(100)	partenaire_zone pz_no INT(11) pz_part_no INT(11) pz_zone_no INT(11)
---	--	---	--	--	--	--	--	---	--	--	--	---	--	--	--	---	--	--	---	---	--	--	--	---	---	--	---	--	---	---

Obrázek 2.3: Datový model

Gestion - Client

Nouveau Enregistrer Supprimer Rechercher Résultat			
<< Premier - < Précédent - Nouveau - Suivant > - Dernier >>			
ID automatique		Ne pas importer	<input type="checkbox"/>
Code			
* Apporteur		No de référence	
* Prénom / Cie			
Nom de famille			
* Adresse - Ligne 1			
Adresse - Ligne 2			
* Ville			
* Pays		* Dept. / Prov.	
* Code postal			
Contact			
Téléphone 1		Type téléphone 1	
Téléphone 2		Type téléphone 2	
Téléphone 3		Type téléphone 3	
Courriel			

Obrázek 2.4: Administrace - přidání klienta

Gestion - Client

Nouveau Enregistrer Supprimer Rechercher Résultat			
Client Produit Suivi			
<< Premier - < Précédent - 1 - Suivant > - Dernier >>			
ID automatique	1	Ne pas importer	<input type="checkbox"/>
Code	110001		
* Apporteur	GRAS SAVOYE-ASC	No de référence	
* Prénom / Cie	Jean		
Nom de famille	FRANCAIS		
* Adresse - Ligne 1	1 route des Oiseaux		
Adresse - Ligne 2			
* Ville	Paris		
* Pays	France	* Dept. / Prov.	75 - Paris
* Code postal	12345		
Contact			
Téléphone 1	0605040302	Type téléphone 1	Mobile
Téléphone 2		Type téléphone 2	
Téléphone 3		Type téléphone 3	
Courriel	jean.francais@gmail.com		

Obrázek 2.5: Administrace - editace klienta

2. ROZBOR STÁVAJÍCÍHO ŘEŠENÍ

Assistance

#2174 - Sélection et localisation du bateau

Veillez compléter le formulaire suivant et sélectionner le contrat concerné avant de passer à l'étape suivante. Vous devez compléter les données obligatoires (*) avant de passer à l'étape suivante.

Bateau *	STORM-AWAY - APRIL M	Zone de navigation *	Pays de Loire 85-44
Latitude	0 <<<	Degrés :	Min. : Sec. :
Longitude	0 <<<	Degrés :	Min. : Sec. :
Points de repère	PORT DE PORNICHE		

Vous devez entrer la latitude et longitude en valeur décimale. Si vous avez ces coordonnées en format *degrés minutes secondes*, veuillez utiliser l'outil de conversion prévu à cet effet.

Étape précédente Étape suivante

Afin de vous aider à sélectionner le bateau du client, voici une liste détaillée des contrats du client.

Contrat	Bateau	Action
<i>No contrat :</i> APR-014785 <i>Produit :</i> APRIL MARINE ASSISTANCE <i>Produit secondaire :</i> <i>Apporteur :</i> <i>Date de début :</i> 12-03-2018 <i>Date de fin :</i> 28-02-2019 <i>Renouvellement :</i> Non renouvelable	<i>Nom :</i> <i>Marque :</i> <i>Modèle :</i> <i>Type de bateau :</i> <i>Type de matériaux :</i> <i>Type de moteur :</i> <i>Marque du moteur :</i> HB MERCURY <i>Puissance du moteur :</i> 150 <i>Zone de navigation :</i> <i>Port :</i> PORNICHE France - 44 - Loire-Atlantique	Sélectionner

Liste des appels d'assistance
Le 21-11-2017 à 09:50:28 par ADMINJOEL [Détail]

Obrázek 2.6: Asistence - první krok

2.6. Zdrojový kód

Assistance

#2174 - Intervention

Panne nécessitant une intervention d'un technicien agréé Marine Assistance:

- Appel à un Intervenant le plus disposé à intervenir
- Explication du problème à l'intervenant.
- Evaluation avec lui de son temps d'arrivée et mise en relation directe si possible avec le Client. Transmission des coordonnées du client voix et confirmation par sms si possible ou mail et position du client confirmés à l'intervenant choisis.
- Envois par mail du "bordereau d'intervention" à l'intervenant. Vérifier qu'il soit pré remplis

Veillez sélectionner le prestataire dépannage.

Prestataire dépannage *	MARINE ASSISTANCE IT	Zone de navigation	Pays de Loire 85-44
MARINE ASSISTANCE INTERVENTION 0811031617 (Travail) LE SEXTANT U12 0887337635 (Mobile) 17000 LA TRINQUETTE 0033547744905 (Télécopieur) 17000 LA ROCHELLE contact@marine-assistance.com France Fiche partenaire			
Commentaire			
Ordre d'intervention	<input type="button" value="Imprimer"/>	<input type="button" value="Envoyer"/>	<input type="button" value="Faxer"/>
<input type="button" value="Étape précédente"/>		<input type="button" value="Étape suivante"/>	

SMS

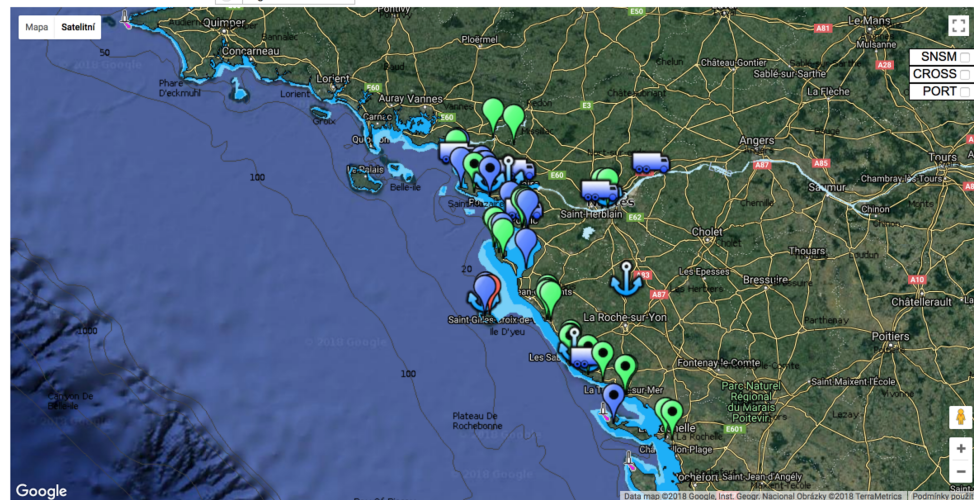
Numéro de téléphone

Message
(Pas de circonflexe ^^)

Suivi de dossier

ID	Type	Date de rappel	Fermé	Fichier	Afficher pour	Créé par	Date	Heure
67185	Doc divers			_local/log_files/67185_5a13e9bdae980.pdf		Joel GUILBAULT	21-11-2017	09:54:21
DEMANDE INTERVENTION IMA								
67184	log					Joel GUILBAULT	21-11-2017	09:53:09
EN ATTENTE DE LA FACTURE DE QUEST NAUTIC								

Coordonnées GPS	Navionics	Type de coordonnées	Distance entre 2 points	
<input type="radio"/> Activé <input checked="" type="radio"/> Désactivé	<input checked="" type="radio"/> Activé <input type="radio"/> Désactivé	<input checked="" type="radio"/> Decimal <input type="radio"/> Degré/Minute <input type="radio"/> Degré/Minute/Seconde	<input type="radio"/> Activé <input checked="" type="radio"/> Désactivé	Lat: _____ Long: _____ Distance: _____ Cap: _____



Informations sur le contrat

Contrat	Bateau	Client
No contrat : APR-014785 Produit : APRIL MARINE ASSISTANCE Produit secondaire : Apparteur : Date de début : 12-03-2018 Date de fin : 28-02-2019 Renouvellement : Non renouvelable	Nom : Marque : Modèle : Type de bateau : Type de matériaux : Type de moteur : Marque du moteur : HB MERCURY Puissance du moteur : 150 Zone de navigation : Port : PORNICHET France - 44 - Loire-Atlantique	Nom : JEAN LUC Adresse : Ville : Pays : Téléphone : Courriel :
		<input type="button" value="Fiche client"/>

Obrázek 2.7: Assistance - intervence

Analýza uživatelských požadavků

Následující kapitola popisuje jednotlivé požadavky, jenž jsou rozdělené na funkční, tedy požadavky na chování systému, a nefunkční, tedy požadavky na vlastnosti systému.

Uživatelské požadavky byly definovány na základě dvou postupů. Prvním z nich byla analýza původního systému a definování jeho jednotlivých funkcí. Následně byly tyto funkce sepsány do podoby požadavků, ze kterých vycházely po diskuzi se společností Marine Assistance výsledné funkční a nefunkční požadavky.

3.1 Funkční požadavky

- F1 **Uchovávání informací o klientech** Aplikace musí udržovat informace o klientech. Nejdůležitějšími informacemi jsou klientovo jméno, kontaktní telefon a e-mail a partner, který klienta přivedl.
- F2 **Ukládání informací o kontraktech klienta** Aplikace musí uchovávat všechny informace o historii kontraktů jednotlivých klientů. Pro každý kontrakt je potřeba ukládat typ produktu, kterého se kontrakt týká, jeho platnost a informace o lodi (název, typ, značka, materiál, motor).
- F3 **Historie asistencí** Systém musí udržovat informace o všech probíhajících (ale i již ukončených) asistencích.
- F4 **Uchovávání důležitých kontaktů** Pro potřeby asistence je důležité, aby aplikace uchovávala kontakty na partnery, techniky, pobřežní hlídky, přístavy apod.
- F5 **Zobrazení kontaktů na mapě** Aplikace bude pro jednodušší vyhledávání pomoci pro klienta zobrazovat kontakty na mapě.

3. ANALÝZA UŽIVATELSKÝCH POŽADAVKŮ

- F6 **Odesílání SMS** Z aplikace musí být možné kontaktovat klienta či techniky pomocí SMS zprávy.
- F7 **Odesílání e-mailů** Důležité dokumenty, jako například fakturu či žádost k intervenci, bude možné odeslat přímo z aplikace ve formě e-mailu.

3.2 Nefunkční požadavky

- N1 **Webová aplikace** Systém bude dostupný jako webová aplikace implementovaná v PHP frameworku Symfony 4. Zároveň bude používat ORM framework Doctrine.
- N2 **Responzivní design** Aplikace by měla vyhovovat požadavkům responzivního web designu.
- N3 **Dokumentace** Zdrojový kód musí být komentovaný a dokumentovaný v anglickém jazyce.

Návrh řešení

4.1 Metodika řízení

K návrhu i implementaci projektu byly využívány prvky agilní metody řízení [5]. Práce probíhala v iteracích, na začátku kterých byl vždy definován seznam požadavků k vypracování pro konkrétní iteraci. Následně byly v rámci iterace jednotlivé požadavky zpracovány a diskutovány s klientem.

Jelikož nebylo možné se se společností Marine Assistance přímo scházet, přibližně jednou měsíčně docházelo k videokonferencím. Na těch byl jednak představen prototyp obsahující vypracované požadavky za předchozí iteraci, druhak byl naplánován průběh iterace následující.

Tato metodika má oproti tradiční vodopádové metodě výhodu například v tom, že pravidelná komunikace se zákazníkem zvyšuje jeho spokojenost a zároveň kvalitu výsledného produktu.

4.2 Datový model

Jelikož bylo nutné zachovat data z původního systému, musela jsem vycházet také z původního datového modelu. V sekci 2.3 byly popsány všechny problémy, kterých se bylo potřeba při novém návrhu vyvarovat.

Nový datový model má oproti tomu původnímu konzistentnější pojmenování entit, jejich atributů i primárních klíčů. Zároveň byly přidány relace, například entita User je nově referencovaná automaticky generovaným id, na rozdíl od uživatelského jména. Entity, které nebyly používány, byly odstraněny a některé redundantní byly sjednoceny. Struktura nové databáze je zobrazena na diagramu 4.1.

Budoucím velkým vylepšením databázového modelu by bylo vytvoření samostatné entity Boat, jež by sloužila k uchování informací o lodích patřících ke klientovu kontraktu. Tyto údaje se bohužel v současné době musí při každém novém vytváření klientova kontraktu zadávat znovu.

4.3 Uživatelské rozhraní

Při návrhu nového uživatelského rozhraní se vycházelo z jeho původní podoby, aby byl přechod na nový systém pro uživatele co nejjednodušší. Zároveň však bylo v sekci 2.4 definováno několik problémů, kterým se při návrhu nového rozhraní bylo třeba vyhnout.

4.3.1 Wireframes

Pro návrh bylo využito wireframes neboli drátěných modelů, díky kterým se dá jednoduše popsat struktura a rozvržení uživatelského rozhraní. [6]

4.3.1.1 Administrace

Problémy v administrační části byly popsány v sekci 2.4.1. Hlavním problémem zde byla celková orientace na stránce. Ačkoliv jsme se měli nacházet v detailu klienta, byli jsme zároveň v editaci jeho údajů. Dalším problémem byla také tlačítka, jejichž část souvisela s editací klienta, část s vyhledáváním a přidáváním nových klientů. Přesto však nebyla nijak vizuálně oddělena.

Těmto problémům jsem se snažila vyhnout rozdělením administrace klienta do dvou kroků. Po výběru klienta se zobrazí jeho detail, podobný původnímu (obr. 2.4). Na rozdíl od něj však detail netvoří editační formulář, ale souhrn všech informací, co o klientovi máme. Záložky s informacemi o produktech a záznamech o klientovi zůstaly na stejném místě, jelikož nijak nebrání orientaci v rámci stránky. Tlačítka pro vyhledávání a vytvoření nového klienta byla oddělena od zbytku stránky.

V případě, že se rozhodneme informace o klientovi editovat (kliknutím na ikonu vedle klientova jména), otevře se modální okno s editačním formulářem (obr. 4.3).

Funkcionalita mazání, jež byla implementována v původním systému, byla vyžadována společností Marine Assistance. Nedokázala jsem však získat odpověď na otázku, zda někdy tuto funkci některý z operátorů využil či využije. Jelikož jsou na klienta vázané i další informace, týkající se například jeho proběhlých asistencí, nebude tato funkcionalita v nové verzi systému implementována. Údaje o proběhlých asistencích se totiž používají například pro různé statistiky, nedávalo by tedy smysl je mazat spolu s mazáním klienta, zároveň nedává smysl nechávat asistence anonymní, tedy bez klienta, u kterého proběhly.

4.3.1.2 Asistence

Uživatelské rozhraní pro průběh asistence by se oproti původní verzi mělo nejvíce změnit. V původní verzi je často nejasné, co je ve kterém kroku potřeba vyplnit, či co je vlastně podstatou toho kroku. Tomuto problému jsem se snažila vyhnout pomocí rozdělení obrazovky na formulářovou a informační část.

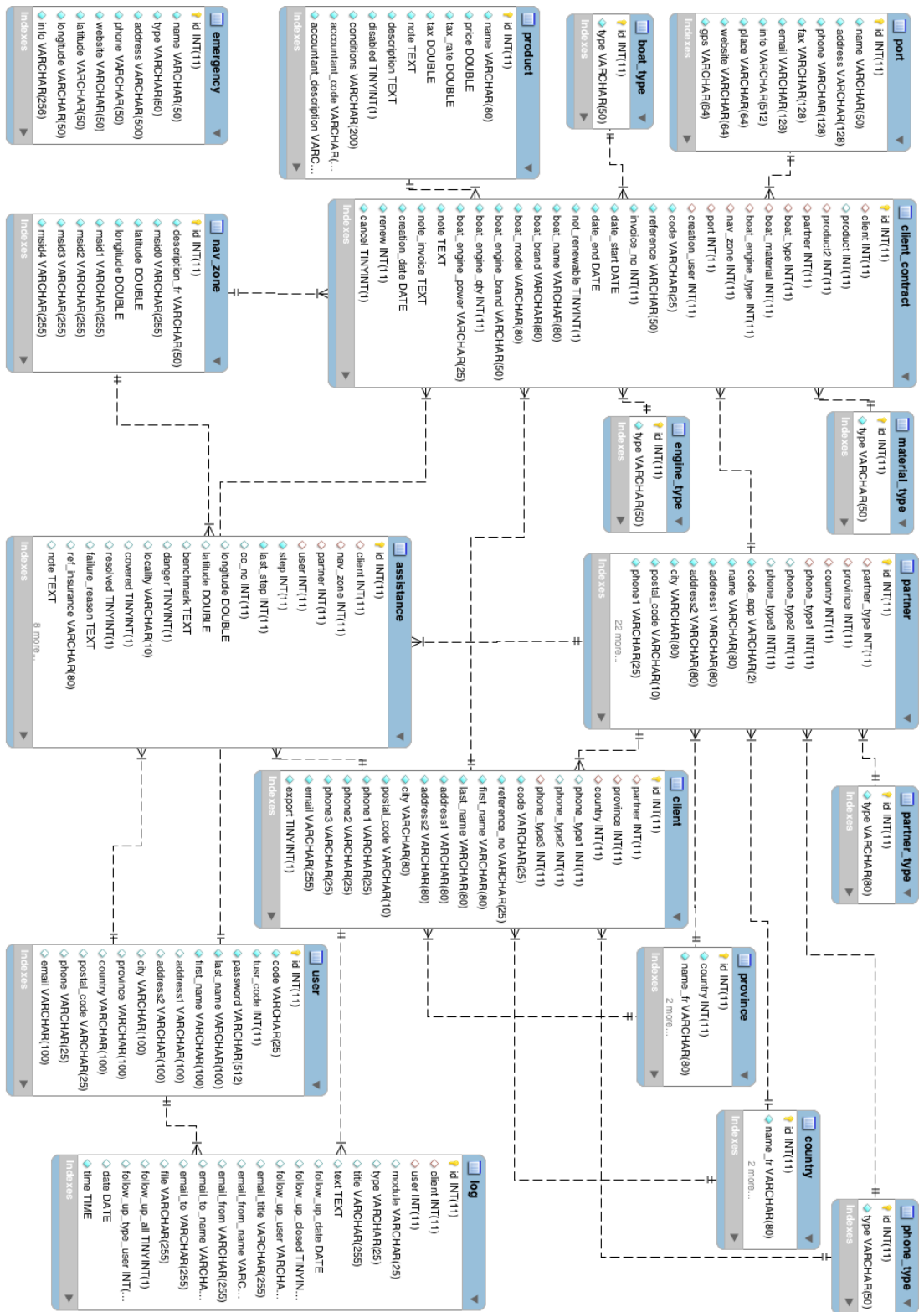
Zároveň jsem přidala možnost schování informací o uživateli a jeho kontraktu, aby nebylo nutné rolovat až na spodní část stránky, kde se původně informace nacházely (obr. 2.7). Tyto změny by měly pomoci uživateli k jednodušší orientaci v rámci každého kroku. Všechny zmíněné změny jsou znázorněny ve wireframu 4.4.

Od prvního do pátého kroku asistence zůstává rozložení obrazovky stejné, jelikož v téměř každém kroku obsahují pouze formulář a informace s ním spojené. V prvním kroku je navíc dostupná mapa, která by měla napomoci snadnější lokalizaci klienta. Rozdílným krokem je krok šestý, nazývaný *intervention*. V tomto kroku se odehrává převážná část samotné asistence: je možné zde na mapě hledat techniky a kontaktovat je. Zároveň můžeme kontaktovat samotného klienta, či případně zanechat k asistenci poznámku nebo přiložit soubor. Wireframe tohoto kroku je na obrázku 4.5 a oproti jeho původní podobě zde bylo změněno velké množství věcí. Snažila jsem se vyřešit všechny problémy představené v sekci 2.4.2 týkající se špatného rozvržení a celkové nepřehlednosti. Vrchní část stránky je rozdělena mezi informace o poskytovateli (technikovi) a informace o klientovi. Pod nimi se nachází další dvě části, které je možné v případě zájmu skrýt: část s mapou a část s tabulkou záznamů v asistenci.

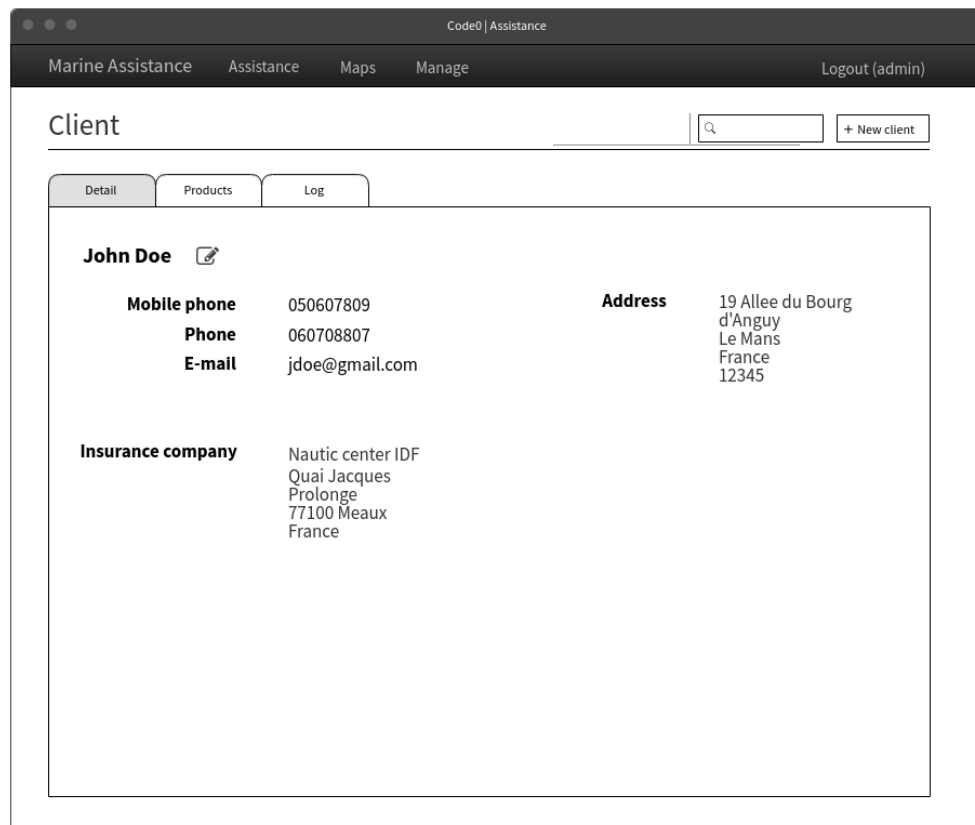
4.3.2 Jazyk

Jelikož bude systém využíván převážně francouzskou společností, výchozím jazykem aplikace je francouzština. Pro potřeby této bakalářské práce a také pro případné budoucí uživatele ze zahraničních firem jsem aplikaci doplnila o anglický jazyk.

4. NÁVRH ŘEŠENÍ



Obrázek 4.1: Datový model



Obrázek 4.2: Wireframe administrace - detail klienta

4. NÁVRH ŘEŠENÍ

Client #1 ✕

First name(s)	<input type="text" value="John"/>	Address - 1st line	<input type="text" value="19 Allee du Bourg d'Anguy"/>
Last name	<input type="text" value="Doe"/>	Address - 2nd line	<input type="text"/>
Mobile phone	<input type="text" value="0506078090"/>	City	<input type="text" value="Le Mans"/>
Phone	<input type="text" value="0607088070"/>	Country	<input type="text" value="France"/>
E-mail	<input type="text" value="jdoe@gmail.com"/>	Postal code	<input type="text" value="12345"/>
Insurance company	<input type="text" value="Nautic center IDF"/>		
Reference number	<input type="text"/>		

Obrázek 4.3: Wireframe administrace - editace klienta

Code0 | Assistance

Marine Assistance Assistance Maps Manage Logout (admin)

Assistance

Client: John Doe Contract: Contract1 ... +

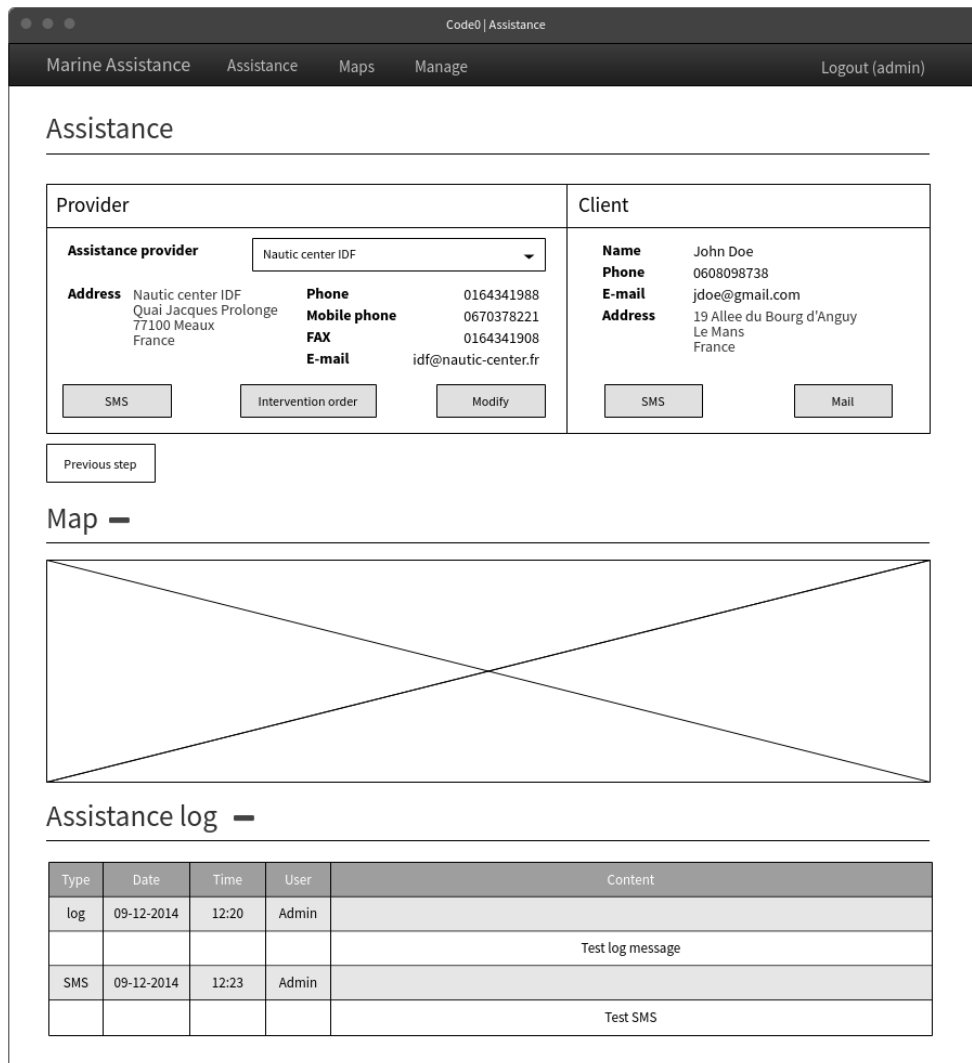
Boat localisation	Help
<p>Boat <input type="text" value="Boat1"/></p> <p>Latitude <input type="text"/> <<< Deg. <input type="text"/> Min. <input type="text"/></p> <p>Longitude <input type="text"/> <<< Deg. <input type="text"/> Min. <input type="text"/></p> <p>Benchmarks <input type="text"/></p>	<p>Important information concerning the current assistance step</p>

Map —

X

Obrázek 4.4: Wireframe asistence - první krok

4. NÁVRH ŘEŠENÍ



Obrázek 4.5: Wireframe asistence - intervence

Implementace

5.1 Symfony

Pro implementaci byl využit PHP Framework Symfony, verze 4 [7]. Důvodů pro zvolení tohoto frameworku bylo několik. Hlavním důvodem byla preference společnosti Marine Assistance k tomuto francouzskému frameworku. Zároveň má jednu z nejrozsáhlejších komunit, což zaručuje stabilitu a dlouhodobou podporu. S tímto frameworkem mám největší zkušenost, zatím jsem se však nesetkala s verzí 4, která přichází s několika novými zajímavými změnami.

Tou největší je nejspíš Symfony Flex, nástroj pro automatickou konfiguraci balíčků [8]. Zároveň Symfony začalo používat novou, přehlednější adresářovou strukturu, která už neobsahuje jednotlivé bundly, ale jediný namespace App. Další výraznou změnou je použití Webpacku Encore místo původní komponenty Assetic.

5.2 Doctrine

Pro práci s databází Symfony používá framework Doctrine. Doctrine používá ORM (objektově relační mapování), tedy automatickou konverzi mezi objekty aplikace a relační databází. Výhodou Doctrine je, že slouží jako abstraktní vrstva nad relační databází, což zaručuje kompatibilitu aplikace s různými relačními databázemi.

5.2.1 Twig

Twig je šablonovacím systémem, který se ve většině případů se Symfony používá. Výhodou Twigu je fakt, že není potřeba kombinovat PHP kód s jazykem HTML (jako k tomu například docházelo v původní aplikaci Code0). Můžeme s jeho pomocí tedy jednoduše oddělit prezentační vrstvu od té aplikační. Twig také escapuje všechny proměnné, díky čemuž předchází XSS útokům [9].

Další výhodou tohoto šablonovacího systému je možnost dědění šablon. V aplikaci existuje základní šablona `base.html.twig`, od které dědí všechny ostatní. Je také možné dědit vícenásobně, v ukázce 5.1 je vidět část `assistance.html.twig` dědící od `base.html.twig` a poté `step1.html.twig` dědící od šablony `assistance.html.twig`.

```
1  {# assistance.html.twig #}
2  {% extends 'base.html.twig' %}
3
4  {% block header %}Assistance{% block step %}{% endblock %}{%
   endblock %}
5
6  {% block body %}
7      {% block client %}{% endblock %}
8
9      <div class="row">
10         <div class="col-md-8">
11             <div class="panel panel-default">
12                 <div class="panel-heading">
13                     <h3 class="panel-title">{% block stepTitle
14                         %}{% endblock %}</h3>
15                 </div>
16                 <div class="panel-body">
17                     {% block stepBody %}{% endblock %}
18                 </div>
19             </div>
20         </div>
21     </div>
22
23 {# step1.html.twig #}
24 {% extends 'assistance/assistance.html.twig' %}
25
26 {% block step %} - step 1{% endblock %}
27
28 {% block client %}
29     <h3>#{{ assistanceId }} {{ client.firstName }} {{ client.
30         lastName }}</h3>
31 ...
```

Listing 5.1: Ukázka vícenásobného dědění šablon

5.3 Bezpečnost

V předchozí části byla zmíněna opatření Twigu proti XSS útokům. K těmto útokům dochází většinou kvůli neošetřeným vstupům, kdy útočník dokáže aplikaci podstrčit vlastní skript, který je poté schopen například získat citlivé údaje uživatele.

Dalším možným útokem je SQL injection, před kterým chrání Doctrine, ovšem pouze v případě, že databázové dotazy jsou jasně definované předem a všechny proměnné dosazujeme pomocí parametrů [10].

5.4 Twilio

Součástí požadavků definovaných v sekci 3.1 je také požadavek na posílání SMS. V původním systému bylo používáno OVH SMS API [11]. Marine Assistance však ve svých ostatních projektech využívá služby společnosti Twilio. Twilio poskytuje API mimo jiné pro posílání a přijímání SMS [12]. I já jsem měla s tímto API několik zkušeností, proto jsme se rozhodli pro jeho použití.

5.5 Symfony Workflow

Pro průběh asistence by se zajisté hodilo v budoucnu použít Symfony komponentu Workflow, jež byla představena v Symfony 3.2 [13]. Tato komponenta velmi ulehčuje práci s objekty, které se v průběhu své existence dostávají do různých stavů ovlivňujících jejich chování. Mezi jednotlivými stavy jsou jasně definované přechody.

S tímto vylepšením je do budoucna počítáno, je však potřeba na tuto změnu připravit již existující data asistencí. To však znesnadňuje fakt, že údaje o stavu asistence nejsou vždy konzistentní a také jsou uloženy v různých attributech. Například údaj o ukončené asistenci je buď uložen v attribute `step` jako hodnota 99, nebo v attribute `last_step` jako hodnota 7. Zároveň však nemusí být splněna ani jedna z těchto podmínek a asistence může být ukončena. Tak je tomu v případě, že má buď attribute `covered` hodnotu `false`, nebo attribute `resolved` hodnotu `true`, nebo také attribute `non_covered` hodnotu `on`.

Jak je vidět, této entitě by tedy použití Symfony Workflow určitě prospělo a, jak už bylo nastíněno v předchozím odstavci, pro další vývoj systému je naplánováno.

Testování

V následující kapitole jsou popsány způsoby, jakými byla vzniklá aplikace otestována.

6.1 Jednotkové a funkční testy

Jak již bylo zmíněno v kapitole 4.1, při návrhu a implementaci se postupovalo podle agilních metod projektového řízení. Tyto metody obsahují mimo jiné automatizované testování.

Pro toto testování byla použita knihovna PHPUnit [14], jež Symfony framework využívá již od své verze 2.7 [15]. Unit testy neboli jednotkové testy byly použity k testování funkčnosti entitních tříd a repositářů. Jednotkové testy slouží k otestování jednotlivých PHP tříd. Pro testování controllerů jsou používány funkční testy, simulující požadavky klienta a testování odpovědí, které controller vrací. Pokrytí entitních tříd, repositářů a controllerů testy je v aplikaci stoprocentní.

6.2 Splnění funkčních požadavků

- F1 Uchovávání informací o klientech - **splněn**
- F2 Ukládání informací o kontraktech klienta - **splněn**
- F3 Historie asistencí - **splněn**
- F4 Uchovávání důležitých kontaktů - **splněn**
- F5 Zobrazení kontaktů na mapě - **splněn**
- F6 Odesílání SMS - **splněn**
- F7 Odesílání e-mailů - **splněn**

6.3 Testování použitelnosti

Původní idea byla taková, že testování použitelnosti bude probíhat s reálnými budoucími uživateli systému, toto řešení by bylo však logisticky velmi náročné. Zvolila jsem tedy možnost jednodušší, a to testovat se čtyřmi testery, kteří mohli být fyzicky přítomni u testování.

Testování probíhalo podle scénářů, které jsou popsány v následující sekci 6.3.1. Před samotným testováním byli testeři zběžně seznámeni s aplikací, poté jsem jako moderátor představovala jednotlivé scénáře, jež následně testeři realizovali. Zároveň byli požádáni o komentování všech kroků a případných nejasností, které by se v průběhu vyskytly. Následně testeři vyplňovali dotazník, dostupný v příloze B.

Stejný průběh byl posléze proveden s každým testerem i na původním systému Code0. Jelikož je tento systém pouze ve francouzském jazyce, trvalo by testování přibližně dvakrát tolik času, na druhou stranu však již byli testeři seznámeni se scénáři. Dle mého názoru lze tedy na obě testování nahlížet jako na ekvivalentní.

Výsledky a porovnání obou testování lze vidět v sekci 6.3.3.

6.3.1 Scénáře pro uživatelské testování použitelnosti

Testovací scénáře kopírují nejdůležitější úkony, které v rámci práce se systémem uživatelé provádí.

Následující scénáře slouží k uživatelskému testování aplikace. Na jeho začátku popis, sloužící k informování testera, co ho v rámci daného scénáře čeká. Následuje požadovaný výchozí a koncový stav aplikace a také přibližná informace o časové dotaci pro daný scénář.

V rámci testování se musí přihlídnout k faktu, že časový odhad není závazující. Je možné, že testera bude zdržovat komentování všech kroků, které dělá. Zároveň se může stát, že tester do požadovaného koncového stavu nikdy nedojde, či dojde, ale s velkými obtížemi a za notného přispění moderátora. Tento průběh je však také velkým přínosem ke zlepšení použitelnosti aplikace.

Poslední sekci každého scénáře jsou jednotlivé kroky, jejichž plněním by se tester měl dostat do cílového stavu. První část (tučně zvýrazněná) je určena pro testera, druhá část slouží jako kontrola pro moderátora. Zde se nachází informace vedoucí k dokončení daného kroku.

6.3.1.1 S01 - Přidání nového klienta

Účelem tohoto scénáře je přidat do systému nového klienta. Klient se jmenuje Michel Doucet, bydlí na adrese 37 Avenue des Minimes, La Rochelle, 17 - Charente-Maritime, 17000, France. Jeho pojišťovna je April Marine. Tyto informace jsou potřebné k vytvoření nového klienta.

- **Výchozí stav:** přihlašovací formulář do aplikace,

- **koncový stav:** detail nově vytvořeného klienta,
- **odhadovaný čas:** 5 minut.

Popis scénáře

1. **Přihlašte se do aplikace.** Pomocí daných přihlašovacích údajů tester vyplní přihlašovací formulář a přihlásí se do aplikace,
2. **přejděte do části s administrací klientů.** Tato část se nachází v hlavním menu v sekci *Manage* -> *Client*. Po kliknutí se zobrazí seznam klientů s možností jejich filtrování a vyhledávání,
3. **vytvořte nového klienta.** Vpravo nahoře se nachází tlačítko *New client*, pomocí kterého se tester dostane k formuláři pro vytvoření nového klienta,
4. **vyplňte informace o klientovi a uložte je.** Po vyplnění povinných informací tester klikne na tlačítko *Save*. Po tomto kroku ho aplikace přesměruje na detail nově vytvořeného klienta,
5. **přidejte ke klientovi informace o jeho e-mailové adrese *mdoucet@hotmail.fr*.** Tester klikne na editační ikonu nacházející se vedle klientova jména a dostane se do editačního formuláře. Zde přidá informaci o klientově e-mailové adrese a kliknutím na tlačítko *Save* informace uloží.

6.3.1.2 S02 - Přiřazení kontraktu klientovi

Klient Martin Bouvier si zakoupil (v e-shopu nebo telefonicky) kontrakt. Tento krok bohužel zatím není v aplikaci automatizovaný, proto je potřeba zadat tuto informaci do systému ručně. V rámci tohoto scénáře je potřeba nalézt daného klienta a přiřadit mu zakoupený kontrakt.

- **Výchozí stav:** domovská stránka aplikace,
- **koncový stav:** výpis kontraktů klienta s nově přidaným kontraktem,
- **odhadovaný čas:** 5 minut.

Popis scénáře

1. **Přejděte na seznam všech klientů a najděte klienta jménem Martin Bouvier.**
2. **Přiřadte klientovi kontrakt Serenity Gold 222.**

6.3.1.3 S03.1 - Založení nové události asistence

Přijali jste telefonát od klienta, který se dostal do potíží. Má platný kontrakt a informace o něm se nachází v systému. Vaším úkolem je klienta najít a založit novou asistenční událost.

- **Výchozí stav:** domovská stránka aplikace,
- **koncový stav:** první krok nově vytvořené asistence,
- **odhadovaný čas:** 2 minuty.

Popis scénáře

1. **V sekci asistence najděte klienta Martin Bouvier** Tester v hlavním menu klikne na položku *Assistance*, která zobrazí formulář pro vyhledávání napříč klienty a informacemi o nich,
2. **pokuste se stejného klienta najít pomocí jeho telefonního čísla 0908070605,**
3. **otevřete novou asistenční událost.** Po kliknutí na tlačítko *Begin intervention* se tester dostane do prvního kroku asistence.

6.3.1.4 S03.2 - Založení nové události asistence pro neexistujícího klienta

Přijali jste telefonát od klienta, který se dostal do potíží. Tento klient však zatím nemá platný kontrakt a není o něm záznam v systému. Tento klient je však ochoten si zakoupit kontrakt, je tedy možné ho v prvním kroku asistence přidat do systému.

- **Výchozí stav:** domovská stránka aplikace,
- **koncový stav:** první krok nově vytvořené asistence,
- **odhadovaný čas:** 5 minut.

Popis scénáře

1. **V sekci asistence vytvořte nového klienta** Tester v hlavním menu klikne na položku *Assistance*, která zobrazí formulář pro vyhledávání napříč klienty a informacemi o nich a zároveň tlačítko *New client*,
2. **zadejte informace, které vám klient sdělil,**
3. **přejděte do prvního kroku asistence.** Po kliknutí na tlačítko *Begin intervention* se tester dostane do prvního kroku asistence.

6.3.1.5 S03.3 - Průchod do závěrečné fáze asistence

V průběhu rozhovoru s klientem získáváte informace o jeho lokaci, stavu jeho lodi, apod. Klient se nachází na souřadnicích 46.1536, -1.8191 na širém moři. Jeho závada je nemožnost nastartovat motor, žádné bezprostřední nebezpečí však nehrozí. Jediným řešením pro klienta bude vybrat partnera, který má možnost za klientem dojet a dopravit ho bezpečně do přístavu. Projděte s klientem všechny kroky asistence a na závěr vyberte nejbližšího partnera a odešlete mu žádost o intervenci.

- **Výchozí stav:** úvodní krok asistence,
- **koncový stav:** závěrečný krok asistence,
- **odhadovaný čas:** 10 minut.

Popis scénáře

1. **Zadejte do systému klientovy souřadnice.**
2. **Postupujte podle instrukcí systému, až dojdete do závěrečného kroku asistence.**

6.3.1.6 S04 - Přidání nového partnera

Cílem tohoto scénáře je přidat do systému informace o novém partnerovi. Tento partner je společnost nazývající se Marine Assistance, která sídlí na adrese 1 Rue de la Trinquette, 17000 La Rochelle, Francie.

- **Výchozí stav:** domovská stránka aplikace,
- **koncový stav:** detail nově vytvořeného partnera,
- **odhadovaný čas:** 5 minut.

Popis scénáře

1. **Přejděte do části s administrací partnerů.** Tato část se nachází v hlavním menu v sekci *Manage* -> *Partner*. Po kliknutí se zobrazí seznam partnerů s možností jejich filtrování a vyhledávání,
2. **vytvořte nového partnera.** Vpravo nahoře se nachází tlačítko *New partner*, pomocí kterého se tester dostane k formuláři pro vytvoření nového partnera,
3. **vyplňte informace o partnerovi a uložte je.** Po vyplnění povinných informací tester klikne na tlačítko *Save*. Po tomto kroku ho aplikace přesměruje na detail nově vytvořeného partnera.

6.3.2 Výstupní dotazník

Výstupní dotazník obsahoval tvrzení, na která tester odpovídal na pětibodové škále (1 - úplně nesouhlasím, 2 - spíše nesouhlasím, 3 - nevím, 4 - spíše souhlasím, 5 - úplně souhlasím). Kritéria hodnocení byla následující:

intuitivita - aplikace mi přišla intuitivní,

jednoduchost - používání aplikace bylo snadné,

atraktivita - aplikace mi připadala atraktivní,

konzistence - vzhled aplikace byl konzistentní,

srozumitelnost - chápal/a jsem, k čemu který ovládací prvek aplikace slouží,

přehlednost - uživatelské rozhraní aplikace je přehledné.

6.3.3 Vyhodnocení

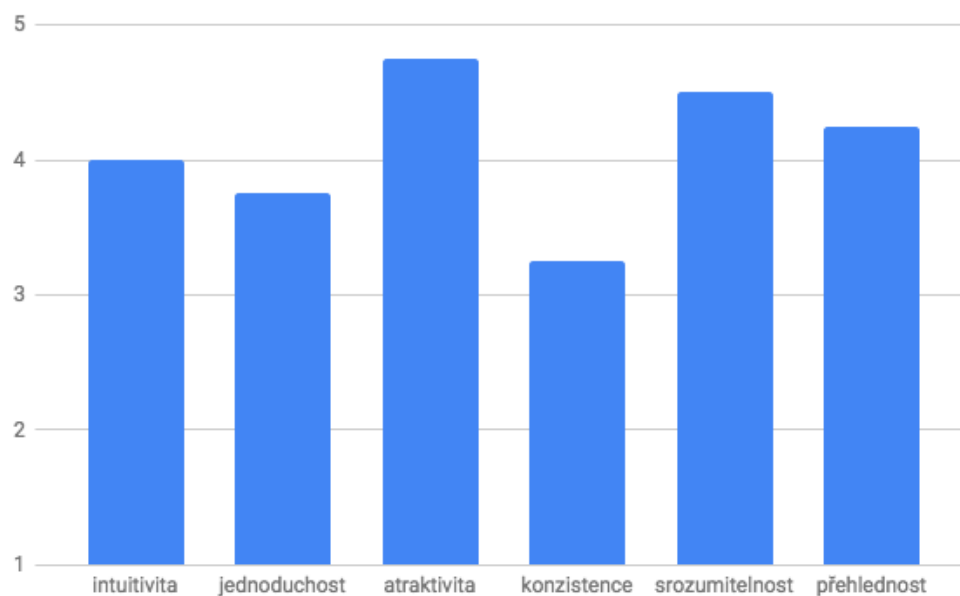
Uživatelské testování dopadlo velmi kladně pro novou verzi systému. Všechny testované aspekty byly hodnoceny známkou 3 a výše, výsledky pro jednotlivá kritéria je možné porovnat v grafu 6.1. V porovnání s hodnocením uživatelského rozhraní původního systému skončil původní systém v každém aspektu v průměru o 1.8 bodu hůře, než nová verze. Výsledky výstupních dotazníků pro původní systém jsou dostupné na grafu 6.2.

6.3.3.1 Testovací log

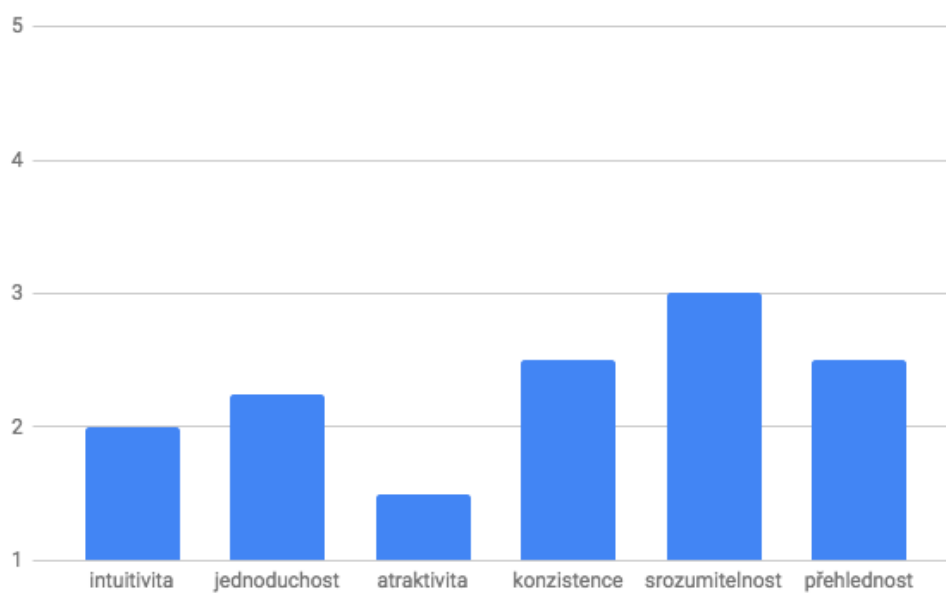
Testování však také odhalilo několik chyb v novém uživatelském rozhraní:

- navigační tlačítka v asistenčním procesu nejsou dostatečně výrazná,
- na několika místech v aplikaci chybí překlady,
- v každém kroku asistence se v pravé straně vyskytuje informace o aktuálním kroku a možnostech uživatele, kromě kroku číslo 3, což pro uživatele působí nekonzistentně,
- ve formuláři pro vytváření nového klienta nebyly zvýrazněny povinné položky.

Všechny výše zmíněné chyby byly opraveny, což přispělo ke zvýšení kvality uživatelského rozhraní.



Obrázek 6.1: Agregované výsledky výstupních dotazníků pro nový Code0



Obrázek 6.2: Agregované výsledky výstupních dotazníků pro původní Code0

Závěr

Cílem této práce bylo analyzovat stávající implementaci informačního systému Code0, následně provést analýzu uživatelských požadavků, navrhnout nové řešení, implementovat ho v PHP frameworku Symfony a na závěr jej otestovat z hlediska funkčnosti a použitelnosti.

V rámci analýzy původní implementace bylo odhaleno množství chyb týkajících se návrhu aplikace, datového modelu, implementace, návrhu uživatelského rozhraní a také organizace a stylu psaní zdrojového kódu. Analýza původního systému a také definice uživatelských požadavků mi poté pomohly se většiny zmíněných chyb v rámci návrhu i implementace vyvarovat.

Navrhnut byl nový datový model, jenž však stále obsahuje jistá omezení vycházející z požadavku na zachování veškerých původních dat aplikace. Díky důkladné analýze a testům původního uživatelského rozhraní mohlo být kompletně navrženo nové uživatelské rozhraní.

Výsledná aplikace, implementovaná v PHP frameworku Symfony, byla na závěr podrobena uživatelskému testování použitelnosti. Toto testování dopadlo velmi kladně, všichni testéři sympatizovali spíše s novým uživatelským rozhraním, než s původní verzí Code0.

Je důležité připomenout, že původní aplikace obsahovala navíc ještě účetní a klientskou část (jak bylo zmíněno v Úvodu). Po otestování současné verze aplikace se společností Marine Assistance budu tedy pokračovat návrhem a implementací chybějících dvou částí.

Literatura

- [1] UpdraftPlus: Does your PHP version matter? [online], [cit. 2018-05-12]. Dostupné z: <https://updraftplus.com/php-version-matter/>
- [2] The PHP Group: Unsupported Branches. [online], [cit. 2018-05-12]. Dostupné z: <http://php.net/eol.php>
- [3] The PHP Group: General Changelog for the ext/mysql extension. [online], ©2001-2018, [cit. 2018-03-12]. Dostupné z: <http://php.net/manual/en/changelog.mysql.php>
- [4] Martin, R. C.: *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008, ISBN 0132350882.
- [5] Drahoslav Dvořák, Martin Mareček: Agilní project management. [online], [cit. 2018-06-05]. Dostupné z: <https://www.systemonline.cz/rizeni-projektu/agilni-project-management.htm>
- [6] Amy Smith: Basic UI/UX Design Concept Difference Between Wireframe & Prototype. [online], [cit. 2018-05-01]. Dostupné z: <https://uxplanet.org/basic-ui-ux-design-concept-difference-between-wireframe-prototype-e38cd3580543>
- [7] SensioLabs: Symfony: Symfony 4. [online], [cit. 2018-06-15]. Dostupné z: <https://symfony.com/4>
- [8] Jani Tarvainen: Differences in Symfony 3 vs. Symfony 4. [online], [cit. 2018-06-25]. Dostupné z: <https://symfony.fi/page/symfony-3-vs-symfony-4-and-flex>
- [9] SensioLabs: Twig: Twig. [online], [cit. 2018-05-10]. Dostupné z: <https://twig.symfony.com/>

LITERATURA

- [10] The Doctrine Project: Security. [online], [cit. 2018-06-15]. Dostupné z: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/reference/security.html>
- [11] OVH: PHP OVH SMS. [online], [cit. 2018-06-15]. Dostupné z: <https://github.com/ovh/php-ovh-sms>
- [12] TWILIO, INC.: Twilio API Reference. [online], [cit. 2018-06-25]. Dostupné z: <https://www.twilio.com/docs/api>
- [13] SensioLabs: Symfony: The Workflow Component. [online], [cit. 2018-06-15]. Dostupné z: <https://symfony.com/doc/3.2/components/workflow.html>
- [14] Sebastian Bergmann: PHPUnit. [online], [cit. 2018-06-22]. Dostupné z: <https://phpunit.de/>
- [15] SensioLabs: Symfony: Testing. [online], [cit. 2018-06-20]. Dostupné z: <https://symfony.com/doc/current/testing.html>

Seznam použitých zkratk

CSV Comma-separated values

IS Informační systém

ORM Object Relational Mapping

PHP PHP: Hypertext Preprocessor

SQL Structured Query Language

UI User Interface

XML Extensible markup language

Výstupní formulář

Pro každou z otázek označte, prosím, jednu z odpovědí 1-5:

- 1 - úplně nesouhlasím,
- 2 - spíše nesouhlasím,
- 3 - nevím,
- 4 - spíše souhlasím,
- 5 - úplně souhlasím.

	1	2	3	4	5
Aplikace mi přišla intuitivní					
Používání aplikace bylo snadné					
Aplikace mi připadala atraktivní					
Vzhled aplikace byl konzistentní					
Chápal/a jsem, k čemu který ovládací prvek aplikace slouží					
Uživatelské rozhraní aplikace je přehledné					

Ke každé z otázek, prosím, napište stručný komentář.

Co se Vám na aplikaci líbilo/nelíbilo?

Byla během testování nějaká chvíle, kdy se aplikace chovala jinak, než byste si představoval/a?

B. VÝSTUPNÍ FORMULÁŘ

Narazil/a jste na něco nelogického?

Je něco, co byste na aplikaci změnil/a?

Další poznámky:

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	code0.....	zdrojové kódy implementace
	text	
	thesis.pdf.....	text práce ve formátu PDF
	thesis.....	zdrojové kódy práce