

SPŠE Ječná

18-20-M/01 Informační technologie

Ječná 30, 120 00, Praha 2

Rezervační systém

(ordinace lékaře)

Jakub Mrázek

Informační technologie

2025

Obsah

1. Cíl práce.....	3
2. Popis projektu.....	3
2.1. Algoritmus	3
2.2. Mechaniky	3
3. Systém requirements.....	4
4. Základní struktura.....	4
5. Testovací data.....	4
6. Uživatelská příručka	5
7. Závěr	5

1. Cíl práce

Cílem práce bylo naprogramovat rezervační systém do ordinance lékaře pro běžného uživatele/pacienta, ale také pro administrátora/zaměstnance ordinace. Každý pacient by měl být schopný se zaregistrovat do systému, přihlásit se pomocí e-mailu a hesla, zarezervovat si termín na vyšetření a odhlásit se. Zaměstnanec/administrátor by měl být schopný si zobrazit veškeré provedené rezervace a má mít možnost jakoukoliv rezervaci, vytvořenou uživatelem/pacientem zrušit a odhlásit se.

2. Popis projektu

Jedná se o plně funkční rezervační systém pro objednání na lékařské vyšetření.

2.1. Algoritmus

Mezi hlavní části algoritmu patří **přihlášení a validace vstupních údajů**, pro jeho kontrolu se používají regulární výrazy. **Rezervace termínu a kontrola dostupnosti** - při vytváření rezervace se kontroluje, zda už tento termín není obsazen předešlými rezervacemi. **Změna a rušení termínu** - při změně se opět kontroluje, zda daný termín není už obsazen a u zrušení se daný termín pouze mění na zrušena=true, toto je bráno v potaz při kontrole, zda daný termín je obsazen; zrušený termín nemizí, protože je následně brán do statistik aplikace. **Filtrování a třídění** - při zobrazení seznamu rezervací se dá třídit a hledat podle jména nebo typu kontroly.

2.2. Mechaniky

Mezi **pacientovy mechaniky** patří **registrace a přihlášení** - registruje se pomocí jména, příjmení, telefonního čísla, e-mailu, typu pojišťovny a hesla, dále se přihlašuje pouze pomocí e-mailu a hesla. **Rezervace termínu** - díky registraci má již pacient předvyplněný formulář, vybírá si pouze druh kontroly, datum a čas rezervace. **Zobrazení rezervací** - díky tomuto pacient vidí veškerou historii svých rezervací, jak zrušených a proběhlých, ale také všechny budoucí rezervace. **Změna hesla** - pacient si může změnit heslo pomocí starého (původního) hesla a zopakováním 2x nového (když staré heslo zapomene, musí napsat na podporu, která je bohužel mimo provoz). **Odhlášení** - pacient se může odhlásit a tím se vrátí do hlavní nabídky menu.

Mezi **mechaniky pro zaměstnance** patří **přihlášení** pomocí jména a hesla. **Zobrazení rezervací** - zaměstnanec má možnost vidět všechny rezervace, filtrovat je a také mazat. **Změna hesla** - funguje stejně jako u pacienta. **Zobrazení statistiky** - zde

je patrné, kolik rezervací proběhlo, jaké typy rezervací byly provedeny a kolik rezervací bylo zrušeno. **Odhlášení** - funguje stejně jako u pacienta.

V hlavní nabídce je také uvedena **základní informace o aplikaci**.

3. Systém requirements

Projekt je programován v programu Java - verze **Java 23**. Pro jeho spuštění je tedy nutné mít nainstalované odpovídající JDK. Mimo samotné JDK je nutné mít zároveň nainstalovnou jednu externí knihovnu - **JDatePicker**, která se využívá k otevření kalendáře a výběru konkrétního dne k rezervaci. Program je možné spustit v libovolném vývojovém prostředí podporující Javu, například **IntelliJ IDEA**, **Eclipse** nebo **NetBeans**.

4. Základní struktura

Program je rozdělen do několika tříd, které spolu navzájem komunikují. Třída **Main** slouží jako výchozí bod spuštění aplikace. Třída **HlavníOkno** slouží jako výchozí bod pro přihlášení, registraci a navigaci. Třída **FormularZmenaTerminu** využívá externí knihovnu **JDatePicker** a slouží ke změně termínu rezervace. Enum **Pojistovna**, reprezentuje typy pojišťoven. Třída **PrihlaseniFormular**, slouží pro přihlášení pacienta do systému. Třída **RegistraceFormular**, slouží k registraci uživatele do systému. Třída **Rezervace** reprezentuje pacienta v systému – (jméno, příjmení, telefon...). **RezervaceFormular** je třída která vytváří formulář pro rezervaci. **RezervaceSpravce**, slouží k správě rezervací. Třída **Spusteni**, zodpovídá za spuštění celého kódu, obsahuje metodu *spusteni()*, která je pak i v Mainu. **StatistikaOkno**, slouží k zobrazení statistik o rezervacích. Enum **TypNavstevy**, obsahuje 3 typy kontroly pacienta.

Třída **Uzivatel** reprezentuje pacienta/zaměstnance v systému.

ZobrazitRezervace vytváří okno pro zobrazení seznamu rezervací s možností filtrování a hledání.

5. Testovací data

Program je možné testovat manuálně, ale zároveň je testován pomocí Unit testů. Mezi základní testy patří kontrola, zda jsou telefonní číslo, ale i e-mail správně zadané. Dále probíhá kontrola, zda na stejný termín nelze provádět 2 rezervace. Když se rezervace zruší, proběhne kontrola, zda je možné na zrušený termín provést novou rezervaci.

6. Uživatelská příručka

Tato aplikace slouží k objednávání pacientů na vyšetření k lékaři. Po spuštění programu se zobrazí okno, kde si uživatel může vybrat, zda se chce přihlásit, zaregistrovat nebo si přečíst základní informace o aplikaci. Nový uživatel se nejprve zaregistruje vyplněním svého jména, příjmení, e-mailu, telefonního čísla, hesla a výběrem své zdravotní pojišťovny. Po registraci se může přihlásit a začít systém používat.

Po přihlášení má pacient možnost si vybrat typ lékařské návštěvy, zvolit si konkrétní datum a čas pro vyšetření. Pokud je některý den již zcela obsazen, systém na to uživatele upozorní. Uživatel vidí jen skutečně volné časy, které si může rezervovat. Uživatel si může také zobrazit přehled svých již vytvořených rezervací. V tomto přehledu může rezervace upravovat, například změnit termín. Všechny informace, které uživatel do systému zadá, se automaticky ukládají a při opětovném spuštění aplikace jsou opět dispozici a již není nutné znovu zadávat údaje. Do systému se může přihlásit i administrátor, který má přístup ke všem rezervacím bez omezení a může také zobrazit statistiku. Správce má navíc možnost rušit rezervace ostatních uživatelů.

7. Závěr

Během vývoje projektu jsem narazil na několik problémů. Nejdříve jsem nevěděl, že Java nemá samostatnou knihovnu s kalendářem, kde by bylo umožněno plánování data (pro termín vyšetření). Proto jsem musel použít externí knihovnu JDatePicker. Chtěl jsem také vytvořit možnost výběru různých druhů ordinací podle oboru lékaře, ale nakonec jsem tento výběr neprovedl. Další nápad byl přidání mapy ordinací v okolí, ale takovou funkci neznám a nevěděl jsem, jak mapu vložit, proto jsem tuto možnost do projektu nepřidal. Několikrát se mi stalo, že jsem se snažil nějakou již vytvořenou funkci opravit, ale při úpravě přestal fungovat jiný algoritmus. Celkově svou práci hodnotím kladně protože si myslím, že takový rezervační systém by měl reálné využití.