	Uniwersytet Technologiczno-Przyrodniczy Im. J. J. Śniadeckich w Bydgoszczy Wydział Telekomunikacji, Informatyki i Elektrotechniki		
Przedmiot	Algorytmy i struktury danych		
Prowadzący	dr inż. Michał Kruczkowski		
Temat	Aplikacja połączona z bazą danych z GUI		
Student	*Mateusz Kalksztejn*		
Email	*Matkal003@utp.edu.pl*		
Data wykonania	19.04.2019	**Data oddania**	24.04.2019

Wstęp

Projekt ma za zadanie pogłębić wiedzę w zakresie tworzenia aplikacji GUI. Idea projektu jest stworzenie aplikacji, która może służyć jako wsparcie organizacyjne dla firmy i ich pracowników. Interfejs ma być przejrzysty, intuicyjny dla odbiorcy.

Użyte technologie

Java 8

Java Lambda

JavaFX

SQLite

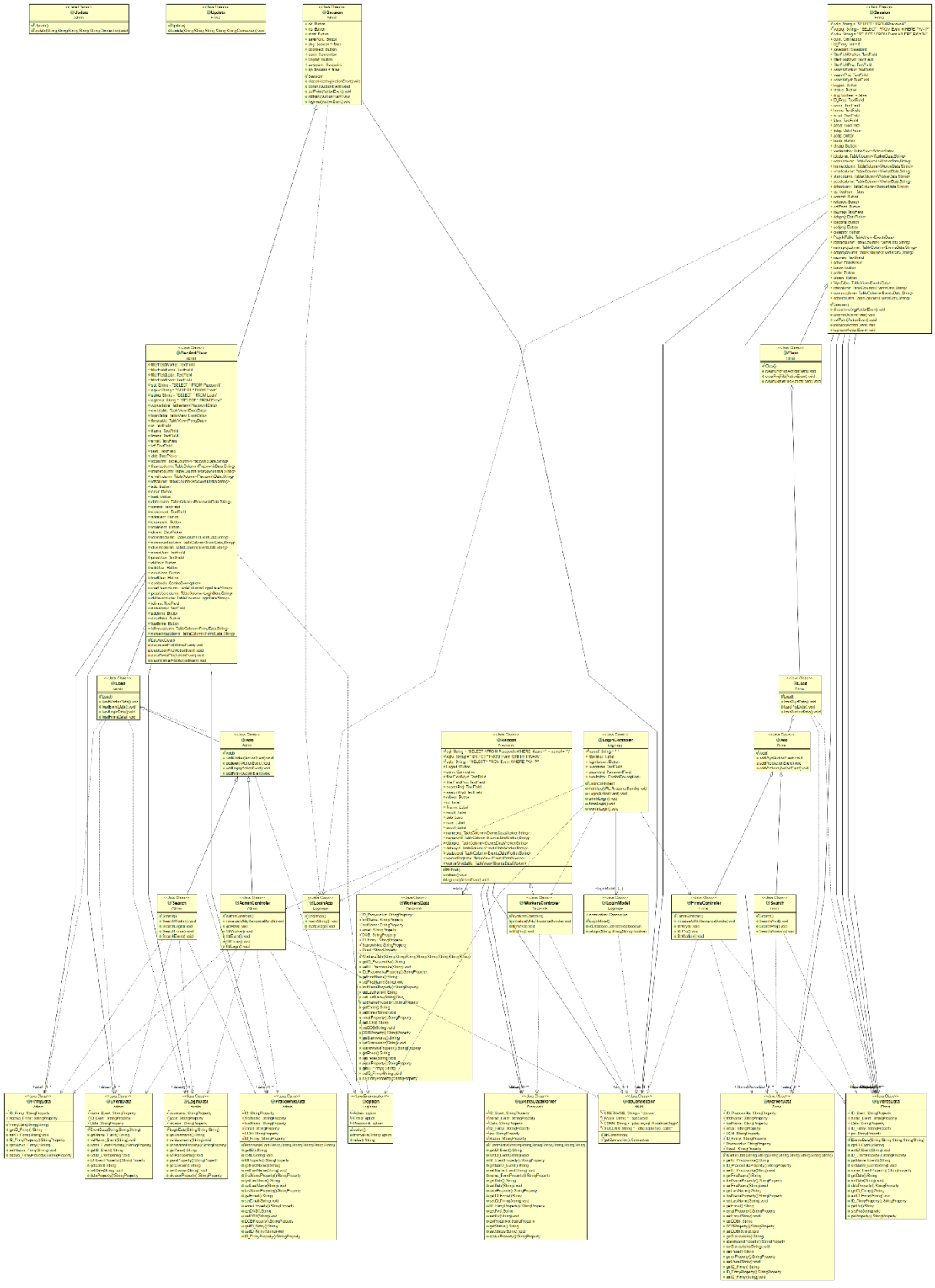
Zarys aplikacji

Aplikacja po uruchomieniu wywołuje okno logowania, w którym mamy wybór na poszczególne sektory: Admin, Firmy, Pracownik (Admin dla konserwacji bazy danych).

Po zalogowaniu uzyskujemy dostęp do określonych części bazy danych.

Aplikacja jest wyposażona w, wyszukiwarkę, mechanizmy oddawania rekordów do bazy danych czy też edycje po kliknięciu w wybraną komórkę tabeli (Pracownik jest pozbawiony funkcji edycji z wiadomych powodów.

Schemat klas aplikacji UML



Omówienie Klas

Dokładne omówienia po linijce znajdują jako komentarze w kodzie źródłowym

dbConnection

Dana klasa zawiera wszystkie ustawienia i metoda niezbędne do połączenia się z baza danych.

W kolejnych plikach tworzymy obiekt klasy by rozpocząć sesje z bazą danych.

LoginApp

Dziedziczy po klasie aplikacji. Zawiera w sobie początek aplikacji (metoda main) która uruchamia aplikacje GUI.

Metoda start inicjalizuje stworzenia okna logowania, które jest kontrolowanie przez klasę LoginControler

LoginModel

Zawiera metodę służącą weryfikacji poprawności danych logowania.

LoginControler

Klasa służąca jako kontroler okna logowania. Posiada podpięty interfejs inicjalizacji.

Tworzy obiekt typu LoginModel oraz zawiera przeciążenia obiektów z formatki fxml.

Metoda Login

Decyduje do którego sektora przenieść użytkownika

Metody AdminLogin , firmaLogin , workerLogin

Tworzą okna dla poszczególnych działów z użyciem własnych kontrolerów

Option

Jest to klasa typu Enum która służy jako wzór dla obiektów Combobox

Klasy z końcówka Data

Służą jako wzór dla pobranych elementów z bazy danych, koniecznie by efektywnie działać na obserwowalnych listach. Dane klasy zawierają Gettery jaki i Settery oraz Konstruktory.

Paczka Admin

Zawiera w sobie wiele klas które maja na celu uszeregować kod by stał się bardziej przejrzysty.

Zastosowałem model dziedziczenia. Nie jest rozwiązanie idealne gdyż radykalnie wydłuża czas budowania aplikacji oraz w przypadku braku jednego pliku cała aplikacja nie działa .

Sposób dziedziczenia

```
Session->DecAndClear->Load->Add->AdminControler
```

Session

Odpowiada za połączeniem z bazą danych, modelem transakcji jak i wylogowaniem użytkownika

DecAndClear

Zawiera większość zdefiniowanych zmiennych oraz przeciążeń obiektów.

Zawiera metody odpowiadające za czyszczenie pól tekstowych z treści.

Load

Zawiera metody pobrania elementów z bazy danych oraz wczytania je do widzialnej użytkownikowi tabeli. Posiada w danych metodach opcje edycji komórek.

Updata

służy jako skrócenie metoda służąca do zmiany rekordu w bazie danych.

Add

Zawiera metody służące do dodania rekordów do bazy danych

AdminControler

Zawiera metody odpowiedzialne za wyszukiwanie oraz filtrowanie danych z tabeli.

Paczka Firma

Jest bliźniaczo podobna do paczki Admin, klasy o tych samych nazwach służą temu samemu lecz ich konstrukcja nieznacznie się różni.

Dziedziczenie

```
Session->Clear->Load->Add->FirmaControler
```

W tej paczce większość zmiennych jest zdeklarowana w klasie Session

Występuje mechanizm, który filtruje rekordy tylko dla Firmy, która posiada dane rekordy.

(Informacje dla innych firm są poufne).

Paczka Pracownik

Pracownik w porównaniu do firmy czy administratora wypada słabo.

Jest pozbawiony możliwości edycji oraz dodania rekordów jak i prowadzenia transakcji (no chyba oczywiste, dlaczego, brak dodawania - brak cofania 😊)

```
Reboot->WorkerControler
```

Reboot

Zawiera deklaracje zmiennych oraz prosty odczyt danych pozbawiony edycji.

W tej klasie znajduje się także metoda służąca wylogowaniu użytkownika.

WorkerControler

Zawiera konstruktor oraz metody służące wyszukiwaniu oraz filtrowaniu.

Pliki fxml

Służąca jako formatka, do której podpinamy kolejne moduły

Link do repozytorium

https://github.com/Karatonik/Algorytmy-projekt

Wnioski

Tworzenie aplikacji nie jest skomplikowane, wystarczy poświęcić na to trochę czasu oraz umieć biegle posługiwać się wyszukiwarką.

Projekt jest aktualnie jako aplikacja znajdująca się na dysku, ale nie ma większych problemów zmienić ja na aplikacje webową, Aplikacja jest w tym względzie dobrze skalowalna.

Wyrażenia lambda są bardzo pomocne, ponieważ nie musimy tworzyć klas anonimowych których wiele ludzi nie rozumie do końca, posługiwanie się tym narzędziem jest bardzo intuicyjne.

Osoba która biegle posługuje się podstawowymi zapytaniami w języku SQL ma duże pole do zabawy w języku Java dzięki bibliotece java.sql