Opis procedur i funkcji dostępnych dla poszczególnych typów użytkowników :

**Zwykły pracownik :**

Podczas wywoływania dostępnych procedur korzysta z paremetru @prac , który jest typu Integer oraz oznacza indeks pracownika. **ZWYKŁY PRACOWNIK NIE MOŻE GO ZMIENIAĆ !**

* Liczba\_godzin\_pracownika(@prac) : Funkcja zwraca łączną sumę godzin pracownika oraz korzystając z relacji wyświetla ile godzin tygodniowo pracownik pracuje dla poszczególnego zespołu.
* Wynagrodzenie\_pracownika(@prac) : Funkcja zwraca łączną sumę wynagrodzenia pracownika oraz korzystając z relacji wyświetla ile pracownik zarabia podczas pracy w poszczególnym zespole.
* Zespoły\_pracownika(@prac) : Procedura wyświetla wszystkie zespoły do, których należy oraz należał pracownik wraz ze swoim obecnym statusem w nich.
* Projekty\_pracownika(@prac) : Procedura wyświetla wszystkie projekty , które realizuje oraz realizował pracownik wraz z obecnym statusem tych projektów.
* Dane\_pracownika(@prac) : Procedura wyświetla dane pracownika , w przypadku błędu w danych należy skontaktować się z odpowiednią instancją.
* Historia\_pracownika(@prac) : Procedura wyświetla wszystkie zespoły oraz realizowane przez nich projekty wraz ze statusem projektów. (Jest to przydatne w momencie , kiedy pracownik np. : ubiega się o dołączenie do nowego zespołu w firmie i wraz z podaniem wysyła wygenerowaną przez aplikacje historie).

**Zarządca zespołu :**

Należy pamiętać o tym , że zarządca zespołu także jest pracownikiem => może korzystać z funkcjonalności aplikacji jak zwykły pracownik. Natomiast z powodu posiadanych obowiązków dostaje także nowe możliwości. Podczas wywoływania swoich procedur korzysta z 3 parametrów : @prac - do funkcjonalności dla zwykłego pracownika , @zesp - jest to parametr typu Integer oraz oznacza indeks zespołu , @cz - jest to parametr typu Integer oraz oznacza indeks członka zespołu, @wyn - Nie ujemny integer określający wysokośc wynagrodzenia dla poszczególnego członka , @czas - Nie ujemny integer określający czas pracy. Zarządca może ustalić wartość parametru @cz , @wyn oraz @czas jednak **NIE MOŻE ZMIENIIAĆ PARAMETRÓW @prac i @zesp!**

* Budżet\_zespołu(@zesp) : Funkcja zwraca łączny budżet zespołu , który jest sumą przydzielonych budżetów do wszystkich projektów realizowanych przez zespół.
* Członkowe\_zespołu(@zesp) : Procedura korzystając z tabeli Grupy\_zespołów wyświetla wszystkich pracowników , którzy znajdują się w danym zespole.
* Ustal\_wynagrodzenie(@zesp,@cz,@wyn) : Procedura modyfikuje tabele Grupy\_zespołów, do atrybutu "wynagrodzenie" wpisuje wartośc @wyn w krotce gdzie atrybut "zespół" = @zesp oraz atrybut "pracownik" = @prac
* Ustal\_czas\_pracy\_członka(@zesp,@czas,@wyn) : Procedura modyfikuje tabele Grupy\_zespołów, do atrybutu "godziny" wpisuje wartośc @czas w krotce gdzie atrybut "zespół" = @zesp oraz atrybut "pracownik" = @prac.
* Ustal\_czas\_pracy(@zesp,@czas) : Procedura modyfikuje tabele Zespoły , do atrybutu "godziny\_tygodniowo" wpisuje wartośc @czas w krotce gdzie atrybut "zespół" = @zesp.
* Zatwierdź\_wynagrodzenie(@zesp) : Funkcja działa jako transakcja do atrybutu "wypłata" w tabeli Pracownik dodaje wartość atrybutu "wynagrodzenie" z tabeli Grupy\_zespołów , wszędzie tam gdzie Pracownik.id\_pracownika = Grupy\_zespołów.pracownik i Grupy\_zespołów.zespol = @zesp. Jeżeli jednak suma wynagrodzeń przekracza budżet zespołu transakcja kończy się niepowodzeniem.
* Zatwierdź\_czas\_pracy(@zesp) : Funkcja działa jako transakcja do atrybutu "godziny\_tygodniowo" w tabeli Pracownik dodaje wartość atrybutu "godziny" z tabeli Grupy\_zespołów , wszędzie tam gdzie Pracownik.id\_pracownika = Grupy\_zespołów.pracownik i Grupy\_zespołów.zespol = @zesp. Jeżeli jednak suma godzin przekracza budżet zaplanowana sumęgodzin pracy dla całego zespołu transakcja kończy się niepowodzeniem.
* Dodaj\_do\_zespołu(@zesp,@cz) : Procedura dodaje do zespołu pracownika o indeksie @cz. Atrybut "status" w tabeli Grupy\_zespołów w krotce gdzie "pracownik" = @cz oraz "zespol" = @zesp zmienia się na "Obecny". (W przypadku dodania nie odpowiedniej osoby należy skontaktować się z odpowiednią instancją aby poprawić zmiany w historii).
* Usuń\_z\_zespołu(@zesp,@cz) : Procedura usuwa z zespołu pracownika o indeksie @cz. Atrybut "status" w tabeli Grupy\_zespołów w krotce gdzie "pracownik" = @cz oraz "zespol" = @zesp zmienia się na "Usunięty". (W przypadku usunięcia nie odpowiedniej osoby należy skontaktować się z odpowiednią instancją aby poprawić zmiany w historii).
* Przenieś\_z\_zespołu(@zesp,@cz) : Procedura usuwa z zespołu pracownika o indeksie @cz. Atrybut "status" w tabeli Grupy\_zespołów w krotce gdzie "pracownik" = @cz oraz "zespol" = @zesp zmienia się na "Przeniesiony". (W przypadku przeniesienia nie odpowiedniej osoby należy skontaktować się z odpowiednią instancją aby poprawić zmiany w historii).
* Projekty\_zespołu(@zesp) : procedura wyświetla wszystkie projekty w , których bierze bądź brał udział zespół @zesp.