**Gabaldarbinieku algas kopotājs**

1. **Problēmas izpēte un analīze.**

Izmantojot novērojumu izpētes metodi, kokapstrādes rūpnīcu darbinieki, kuriem atalgojums par darbu tiek izmaksāts pēc padarītā darba daudzuma, līdz algas dienai grūti izsekot cik ir viņu nopelnītā summa, ja vien nekonsultējas ar grāmatvedi. Šādas konsultācijas aizņem lieku laiku, ko varētu izmantot produktīvāk gan ražotnes darbinieks, gan grāmatvedis. Protams, ka veicamās darbības, lai šo problēmu atrisinātu, ir vienādas un atkārtojas (aiziešana, apjautāšanās, rezultāta saņemšana un došanās prom).

1. **Programmatūras prasību specifikācija.**

Procesu var automatizēt, izveidojot programmu, kur darbinieki paši ievada apstrādāto materiālu daudzumu un, vadoties pēc iepriekš ievadīta koka apstrādes izcenojuma, ir iespējams apskatīt savu nopenīto bruto algu. Mērķauditorija ir jebkurš darbinieks, kura alga tiek rēķināta pēc padarītā darba, balstoties uz iepriekš aprakstīto problēmu, tikai pirms programmas lietošanas uzsākšanas ir vajadzība implementēt kodā darba tarifus.

Programmatūra iespējās ir izdarītā darba atlasīšana un daudzuma ievadīšana, darba algas summa, tāpēc tiks izveidota datubāze, lai varētu apstrādāt statistiku. Pieslēgšanās profilam notiek ar vienkāršu paroli. Dati tiek attēloti tabulā, kurus var sakārtot pēc jebkuras kolonas. Datus var izdzēst no datubāzes, kā arī mainīt summu. Darbu identificē ar ID numuru.

Programmatūra ir rakstīta python valodā, jo ir viegli saprotama un iespēja apstrādāt liela daudzuma datus. Lietotāja saskarne būvēta ar *PySimpleGUI* palīdzību, jo to ir viegli implementēt *python* kodā. Datu bāze nodaļa tiks veidota ar *sqlite3* bibliotēkas palīdzību. Paroles kodēšanu nodrošina *hashib* bibliotēka, bet sakārtošanu *operator* bibliotēka. Kods realizēts *Visual Studio Code* platformā, jo tā piedāvā pajomīgu funkciju daudzumu un paplašinājumus, kas palīdz programmatūras izstrāde un šo saturu piedāvā bezmaksas.

1. **Programmatūras izstrādes plāns**

Programmatūras izstrādē tiks izmantots Iteratīvais modelis. Pēc veiksmīgas problēmas izpētes, prasību specificēšanas un projektēšanas, var veidot nepārtrauktu ciklu ar izstrādi, testēšanu un izvērtēšanu. No sākuma tika atrasta problēma, uz kuras varētu balstīt programmu. Vēlāk izveidoja programmas detalizētāku aprakstu un viss tika dokumentēts sīkāk pa vajadzīgajām nodaļām programmas specifikācijas dokumentā. Uzsākot nebeidzama cikla darbību par turpmāko programmas veidošanu, no sākuma izstrādā konkurētu koda daļu, kas atbilstu par kādu no funkcijām (piemēram, datu ievade, tabulas izveide, datu rediģēšana u.t.t.), tālāk tiek veikta funkcijas testēšana paredzot visus iespējamos scenārijus, veikt visas darbības un pēcāk jau izvērtēt kļūdas un to iespējamās novēršanas. Kad šāds viens cikls ir veiksmīgi pabeigts, atkal sāk no sākuma ar koda izstrādi jau jaunai funkcijai.

1. **Atkļūdošanas un akcepttestēšanas pārskats.**

Datu ievades laukos, kur vajag ievadīt skaitļus, ja ievada nenaturālu skaitli vai burtu, pēc pogas “Ievadīt” nospiešanas parādās logs, ka ievads nav pareizi ievadīts. Ar pogas “Ievadīt” palīdzību pārbauda vai nav kāds no ievades laukiem atstāts tukšs, pretējā gadījumā izlec arī atbilstošs logs. Programma pārtrauc savu darbību, ja ievadītais ID jau ir sastopams datubāzē un summas ievades logā ir ievadīts negatīvs skaitlis, burts vai citādi nepareizi ievadīts decimāldaļas skaitlis. Pie datubāzes datu rediģēšanas logiem arī ir piesaistīta datu pareizības pārbaude, kā tas ir aprakstīts jau iepriekš.

1. **Lietotāja ceļvedis.**

Programma tiek palaista, kad palaiž failu *main.py.* No sākuma parādās logs, kur jāievada lietotājvārds (pieejamie lietotājvārdi ir “janis”, “kokle”, “kubiks”) un parole (vienīgā derīgā parole ir “kubik123”). Ja lietotājvārds un/vai parole ir nederīgi, tad parādās logs ar attiecīgu paziņojumu, bet pretējā gadījumā pieslēgšanās logs pazūd un var uzsākt darbu ar datiem.

Ievades loga pirmā daļa ir paredzēta datu ievadei (jāaizpilda logi ar darba numuru, summu, darba tipu un daudzumu). Pēc veiksmīgas logu aizpildīšanas, ko pārbauda atkļūdošana, kas ir aprakstīta šī dokumenta 4. punktā, nospiežot pogu “Ievadīt”, parādās logs par veiksmīgu datu ievadi. Poga “Exit” pārtrauc programmas darbību. Nākama loga rinda, gadījumā, ja ir vēlme izdzēst datus, tad, ievadot konkurētu darba numuru, tiek no datubāzes izdzēsts darba numurs ar tā attiecīgajiem datiem. Loga pēdējā rinda spēj samainīt summu ierakstot konkurētu darba numuru un summu uz ko lietotājs vēlas mainīt. Šīs abas darbības tiek izpildītas nospiežot attiecīgo pogu rindas beigās un datu validāciju arī pārbauda pēc līdzīga principa kā tas ir aprakstīts šī dokumenta 4. punktā.

Nospiežot pogu “Tabula”, parādās logs ar izveidotu tabulu, kur ir attēloti visi datubāzē pieejamie dati. Virs katras kolonas ir to nosaukumi, pēc kuriem arī ir iespēja kārtot datus augošā secībā. Ja tiek veiktas datubāzes datu rediģēšanas darbības, kamēr tabula ir atvērta, tad, lai atjaunotu tabulas datus, tā ir jāaizver un jāatver vēlreiz.