# To-Do List projekts

**1) Problēmas skaidrojums, risinājuma izvēle.**

**Problēma –** Cilvēkiem ikdienā bieži vien rodas uzdevumi, kas jāizdara. Tie var būt personiski vai profesionāli uzdevumi. Šie uzdevumi visbiežāk tiek pierakstīti uz papīra lapiņām vai dienasgrāmatā, kas ir neefektīvi, jo tās var viegli pazust un tikt aizmirstas. Kā arīir grūti uzturēt aktīvu sarakstu uz lapas, jo uzdevumi bieži mainās vai tiek papildināti.

**Risinājuma izvēle –** Lai šo problēmu risinātu man radās izvēle izveidot digitālu To-Do list, kas piedāvātu lietotājam ērtu un efektīvu veidu, kā organizēt savus izdarāmos uzdevumus. Šī ideja attīstījās novērojot cilvēku vajadzību pēc labākas uzdevumu pārvaldības un digitāla risinājuma, ko var izmantot jebkurā laikā.

Pirms jauna risinājuma izstrādes ir svarīgi veikt esošo programmu izpēti, lai saprastu kādas funkcijas piedāvā jau esošās programmas.

Izpētot dažādas aplikācijas rezultātā sapratu, ka tajās jau ir daudz un dažādas funkcijas. Tāpēc veidošu tādu aplikāciju, kurai var piekļūt arī bez interneta un kura ir ļoti viegli saprotama. Pārējās To-Do list programmas ir ļoti piepildītas ari liekām lietām, izveidošu pavisam vienkāršu aplikāciju, kuru saprastu jebkurš.

**2) Programmatūras prasību specifikācija**

**Programmas mērķauditorija**:

* privātpersonas, kas vēlas vienkārši pārvaldīt savus uzdevumus vienkāršā, bet efektīva veidā.
* Profesionāļi, kuriem nepieciešama darba organizācija efektīvā uzdevumu vai projekta izpildē un plānošanā.
* Skolēni, kam nepieciešams apkopot savus mājasdarbus un projektus.

Izpētes rezultāts liecina, ka mērķauditorija ir ļoti dažāda, bet ar vienu uzdevumu – izpildīt un apkopot izdarāmos uzdevumus.

**Funkcijas:**

* Pievienot uzdevumu - pievieno jaunu uzdevumu saraksta apakšā.
* Aplūkot visu plānu. - lietotājs var apskatīt visus esošos uzdevumus,
* Aplūkot pabeigtos uzdevumus – apskata visus pabeigtos uzdevumus.
* Atzīmēt uzdevumu, kā izpildītu. - Pabeigto uzdevumu atzīmē, un pēc pogas nospiešanas atzīmē to, kā izpildītu.
* Dzēsts uzdevumu – Var izdzēst uzdevumu no plāna, ja kaut kas mainās un tas vairs nav jādara. Uzdevumu atzīmē un nospiež izdzēst.
* Rediģēt uzdevumus – Ļauj rediģēt jau ievietotu darbu, nemainot ievietoto darbu numerāciju.

Programma vēl sastāvēs no 2 laukiem. Ievades lauka, kur tiks ierakstīts uzdevums, kas jāpievieno. Apskates lauka, kur tiks parādīti visi darāmie uzdevumi vai jau izdarītie – atkarīgs no nospiestās pogas tajā momentā.

Funkcijas skice (jāuztaisa un jāsaprot)

**3) Programmatūras izstrādes plāns**

**Ūdenskrituma modelis:**

* Prasību specifikācija – Saprast lietotāju prasības un nepieciešamo funkcionalitāti. Veikt izpēti par līdzīgas programmatūras izpēti. Uzrakstīt nepieciešamo funkciju aprakstu un uzdevumus.
* Projektēšana - Izstrādāt programmas struktūru un algoritmu. (kā strādās programma)
* Kodēšana - Individuāli izveidot katru funkciju, lai tā darbotos pēc iecerētā.
* Integrācija - Izveidoto kodu apvienot vienā kodā.
* Testēšana - Veikt aplikācijas testēšanu, lai pārbaudītu vai strādā veiktā integrācija. Pārbaudīt lai visi ievadītie dati veic uzdevumu ,kas bija plānot prasību specifikācijā.
* Ieviešana - Apkopot visu programmu vienā, izdzēsts visu lieko. Labot kļūdas, kas tika atklātas testēšanas rezultāta. Izveidot instrukciju un dokumentāciju lietotājam, kā viss strādā.
* Uzturēšana Nodrošināt programmas kvalitatīvu darbību. Veikt labojumus pēc izlaišanas, ja tādi tiek atklāti. Sniegt atbalstu lietotājiem, risinot problēmas un atbildot uz jautājumiem.

**4) Rezultātu analīze**

**Posmu rezultātu sasniegšanas atskaite:**

* Prasību specifikācija – Izveidota prasību specifikācija. Analizēta mērķauditorija un uzrakstītas nepieciešamas funkcijas un to darbība. Tika veikta detalizēta funkciju un prasību noteikšana.
* Projektēšana – Izstrādāta maza sistēma, kā strādās funkcijas un ko katra darīs. (kā apstrādās datus). Tika veikta programmas izveides plānošana, lai pareizi varētu izveidot programmu.
* Kodēšana – Izveidota katra funkcija, tā lai tā darbojas pēc plāna. Funkciju individuālā darbība.
* Integrācija – Viss kods salikts kopā, lai darbotos pēc OOP principa. Pārtaisīts, lai strādā ar “tkinter” vizualizācijas aplikāciju. Tādā veidā lietotājam nav jāraksta informācija terminālī. Jebkurš var saprast aplikāciju.
* Testēšana – Notestēta aplikācija. Uztaisīts, lai parādās kļūdu ziņojumi, ja kaut kas netiek darīts pēc plānotā. Viss strādā pareizi. Tad, kad apskata pabeigtos uzdevumus, liekās pogas nevar izmantot.
* Ieviešana – Apkopots viss kods, liekās rindiņas noņemtas. Viss strukturēts, lai viegli saprotams un pārskatāms.
* Uzturēšana – programmas pārskats, ja parādīsies kļūdas veikšu labojumus, bet pašlaik viss strādā.

Specifikāciju pārskats. Pagaidām visas funkcijas uzrakstītās plāna ir iekļautas un darbojas, kā plānots.

**5) Lietotāja ceļvedis**

Sveiks lietotāj!

Aplikācija izveidota ļoti vienkārša, lai katrs var saprast. Palaižot to atvērsies jauns aplikācijas logs, kur būs redzami 2 lauki un vairākas pogas. Pogu darbību varēs saprast no tā nosaukuma, ja kaut kas tiks darīts nepareizi aplikācija paziņos, kas jādara savādāk.

Iesaku darboties tieši ar mūsu aplikācija, jo tā nesatur liekas funkcijas, tikai pašu vienkāršāko. Rezultātā varat sagaidīt pārskatāmu darāmo uzdevumu aplikāciju, kuru var atvērt jebkurā laikā un vietā. Visu laiku tur būs apkopoti jūsu uzdevumi, kā arī jau pabeigtie darbi, lai zinātu, kas izdarīts.

**6) Pašvērtējums**

Projektā izdevās viss, kas bija ieplānots. Apmēra aplikāciju arī iztēlojos tādu, kur var viegli visu redzēt un apkopot.

Savādāk es laikam darītu vizualizācijas daļu, jo “tkinter” paplašinājumā nebija daudz opciju, lai smuki vizualizētu aplikāciju. Tieši izskats bija tas, ko vēlējos beigās uzlabot, bet līdz galam nesanāca iecerētais.

Kopumā esmu apmierināts ar paveikto. Bija interesanti veidot aplikāciju un izpētīt, kā strādā “tkinter” (iepriekš ar to nebiju strādājis).

**7) Programmas kods**

<https://github.com/ronaldsplostnieks/to-do-list>