Risinājums: Treniņuzdevums Nr. 1 (Autoserviss "Ātrais Rats")

**1. uzdevums: Iesaistītās puses un prasības**

**Paskaidrojums:** Iesaistītās puses ir visas personas vai grupas, kuru darbu vai ikdienu ietekmēs jaunā sistēma. Ir svarīgi identificēt ne tikai tiešos lietotājus (mehāniķus), bet arī vadību un klientus. Papildu funkcionalitātei ir jābūt loģiskai un atbilstošai katras puses lomai.

1. **iesaistītā puse:** Uzņēmuma vadītājs  
   **Papildu funkcionalitāte:** Atskaišu modulis, kas ļauj analizēt datus par populārākajiem pakalpojumiem, peļņu un mehāniķu noslodzi.
2. **iesaistītā puse:** Mehāniķis  
   **Papildu funkcionalitāte:** Iespēja pievienot remontam fotogrāfijas (piemēram, bojātā detaļa pirms un pēc remonta), kas tiek saglabātas klienta remonta vēsturē.
3. **iesaistītā puse:** Klients  
   **Papildu funkcionalitāte:** Iespēja tiešsaistē apskatīt un apstiprināt remonta tāmi pirms darbu uzsākšanas.
4. **iesaistītā puse:** Administrators (klientu apkalpošanas speciālists)  
   **Papildu funkcionalitāte:** Ātrās meklēšanas funkcija, kas ļauj atrast klientu pēc vārda, tālruņa numura vai automašīnas reģistrācijas numura.

**2. uzdevums: Mērķauditorija un izpētes metodes**

**Paskaidrojums:** Katrai prasībai ir sava galvenā lietotāju grupa (mērķauditorija). Izpētes metode jāizvēlas tāda, kas vislabāk palīdzētu iegūt informāciju no šīs konkrētās grupas.

1. **prasības mērķauditorija:** Autoservisa mehāniķi  
   **Izpētes metode:** **Novērošana.** Vērojot mehāniķi darba procesā, var vislabāk saprast viņa darba specifiku – kādas ir neērtības, kādā secībā notiek darbības, un vai rīks būs jālieto vidē, kur rokas ir netīras (kas ietekmē ierīces izvēli, piem., planšete, nevis dators ar peli).
2. **prasības mērķauditorija:** Autoservisa klienti (esošie un potenciālie)  
   **Izpētes metode:** **Anketēšana.** Tā kā klientu ir daudz, anketēšana ļautu ātri iegūt kvantitatīvus datus par to, kādas funkcijas viņiem šķiet vissvarīgākās (piem., vai svarīgāk ir redzēt remonta vēsturi vai saņemt SMS paziņojumus), un noskaidrot viņu digitālās prasmes.

**3. uzdevums: Darba uzdevumi izstrādātājiem**

**Paskaidrojums:** Šeit problēmvides apraksts jāpārvērš konkrētos, izpildāmos uzdevumos programmētāju komandai. Uzdevumiem jābūt tehniskiem un skaidri definētiem.

1. **darba uzdevums:** Izveidot relāciju datubāzes struktūru (shēmu), kas nodrošina klientu, viņu automašīnu, pierakstu, veikto remontdarbu un noliktavas rezerves daļu datu glabāšanu.
2. **darba uzdevums:** Izstrādāt lietotāja saskarni (front-end) ar drošu autentifikācijas sistēmu administratoriem un mehāniķiem, kas nodrošina piekļuvi pierakstu kalendāram un darbu reģistrēšanas formai.
3. **darba uzdevums:** Implementēt API integrāciju ar SMS sūtīšanas pakalpojumu sniedzēju (piemēram, Twilio), lai realizētu automātisku atgādinājumu nosūtīšanu klientiem.

**4. uzdevums: Izstrādes modeļa izvēle**

**Paskaidrojums:** Izvēle ir jāpamato, sasaistot modeļa īpašības ar projekta specifiku. Šajā gadījumā prasības var mainīties, un ir svarīgi ātri iegūt atgriezenisko saiti.

**Izvēlētais modelis: Veiklais modelis (Agile)**

**Pamatojums:**

* **Elastība un pielāgošanās:** Autoserviss ir neliels uzņēmums, un tā vadītājam, visticamāk, radīsies jaunas idejas un prasību izmaiņas izstrādes gaitā, kad viņš redzēs pirmos sistēmas prototipus. Agile modelis ir ideāli piemērots šādām situācijām, jo tas ļauj viegli mainīt prioritātes un pievienot jaunu funkcionalitāti katrā izstrādes ciklā (sprintā).
* **Ātra atgriezeniskā saite:** Izstrāde notiek īsos ciklos, kuru beigās tiek piegādāta strādājoša produkta daļa (piemēram, sākumā tikai pierakstu kalendārs). Mehāniķi un administrators to var uzreiz izmēģināt reālos darba apstākļos un sniegt atsauksmes, kas palīdzēs uzlabot produktu nākamajos posmos.
* **Risku mazināšana:** Sākot ar vissvarīgāko funkcionalitāti (Minimum Viable Product), uzņēmums ātrāk iegūst reālu vērtību. Tas ir mazāk riskanti nekā Ūdenskrituma modelis, kur gala produkts tiek ieraudzīts tikai izstrādes beigās.

**5. uzdevums: Īstenošanas varianti**

**Paskaidrojums:** Jāpiedāvā divi principiāli atšķirīgi risinājumi. Šeit klasisks piemērs ir tīmekļa lietotne pret darbvirsmas lietotni. Katram variantam tehnoloģiju izvēlei (valoda, izvietošana) ir jābūt loģiski saistītai ar aprakstu.

**1. variants:**

* **Īss apraksts:** **Tīmekļa lietotne (Web Application)**, kas pieejama no jebkura datora vai planšetes ar interneta pieslēgumu caur pārlūkprogrammu. Sistēmai ir divas daļas: publiskā daļa klientiem (pierakstīšanās, vēstures apskate) un iekšējā daļa darbiniekiem (kalendārs, remontu uzskaite).
* **Izvietošanas veids:** Mākoņserveris (piemēram, DigitalOcean, AWS) vai tīmekļa mitināšana (web hosting).
* **Programmēšanas valoda:** Python (ar Django vai Flask ietvaru) vai PHP (ar Laravel ietvaru) servera daļai; JavaScript (ar React vai Vue.js) lietotāja saskarnei.
* **Koda redaktors:** Visual Studio Code.

**2. variants:**

* **Īss apraksts:** **Darbvirsmas lietotne (Desktop Application)**, kas tiek instalēta tikai uz autoservisa datoriem. Tā nodrošina ātru un stabilu darbību iekšējā tīklā, bet neparedz tiešu piekļuvi klientiem no mājām. Klientu pierakstīšana notiek telefoniski, un administrators datus ievada sistēmā.
* **Izvietošanas veids:** Lokālais serveris uzņēmuma telpās, uz kura glabājas kopīgā datubāze.
* **Programmēšanas valoda:** C# (izmantojot .NET un WPF/Windows Forms) vai Java (izmantojot JavaFX).
* **Koda redaktors:** Visual Studio (priekš C#) vai IntelliJ IDEA (priekš Java).

Risinājums: Treniņuzdevums Nr. 2 (Mūzikas skola "Do-Re-Mi")

**1. uzdevums: Iesaistītās puses un prasības**

1. **iesaistītā puse:** Skolas vadība  
   **Papildu funkcionalitāte:** Finanšu pārskatu modulis, kas apkopo informāciju par ienākumiem, izdevumiem un parādniekiem.
2. **iesaistītā puse:** Skolotājs  
   **Papildu funkcionalitāte:** Mācību materiālu bibliotēka, kur skolotājs var augšupielādēt notis, audio ierakstus vai video, kas pieejami viņa audzēkņiem.
3. **iesaistītā puse:** Audzēknis/Vecāks  
   **Papildu funkcionalitāte:** Iespēja pieteikties skolas rīkotajiem pasākumiem un koncertiem caur platformu.
4. **iesaistītā puse:** Skolas administratore  
   **Papildu funkcionalitāte:** Rīks jaunu audzēkņu grupu izveidei un pārvaldībai, kas ļauj ērti piešķirt skolotāju un nodarbību laikus.

**2. uzdevums: Mērķauditorija un izpētes metodes**

1. **prasības mērķauditorija:** Mūzikas skolas skolotāji  
   **Izpētes metode:** **Fokusa grupa** vai **Intervija.** Tā kā skolotāju skaits, visticamāk, nav ļoti liels, var organizēt diskusiju (fokusa grupu), kurā kopīgi izstrādātu labāko sekmju vērtēšanas sistēmu. Individuāla intervija palīdzētu noskaidrot katra skolotāja specifiskās vajadzības.
2. **prasības mērķauditorija:** Audzēkņu vecāki  
   **Izpētes metode:** **Prototipēšana.** Izveidojot vienkāršu, interaktīvu paziņojumu sistēmas prototipu (piemēram, Figma), varētu to parādīt vecākiem un lūgt viņiem veikt konkrētus uzdevumus. Tas ļautu novērot, vai sistēma ir saprotama un ērta, un uzreiz saņemt konkrētus uzlabojumu ieteikumus.

**3. uzdevums: Darba uzdevumi izstrādātājiem**

1. **darba uzdevums:** Izstrādāt datubāzes modeli, kas saista audzēkņus, skolotājus, nodarbību grafikus, maksājumus un sekmju žurnālus.
2. **darba uzdevums:** Implementēt lietotāju lomu un tiesību sistēmu (RBAC - Role-Based Access Control), kas nodrošina, ka administratore, skolotājs un vecāks redz tikai sev paredzēto informāciju.
3. **darba uzdevums:** Izveidot moduli, kas katra mēneša sākumā automātiski ģenerē rēķinus visiem audzēkņiem, balstoties uz viņu nodarbību plānu, un integrēt to ar tiešsaistes maksājumu platformu (piem., Stripe).

**4. uzdevums: Izstrādes modeļa izvēle**

**Izvēlētais modelis: Veiklais modelis (Agile)**

**Pamatojums:** Līdzīgi kā iepriekšējā piemērā, mūzikas skolas vajadzības var būt mainīgas. Agile pieeja ļauj izstrādi sadalīt mazākos, pārvaldāmos posmos. Piemēram, pirmajā sprintā var izveidot tikai nodarbību grafiku, kas ir kritiskākā problēma. Pēc tam, saņemot atsauksmes no lietotājiem, var pievienot rēķinu moduli, sekmju uzskaiti utt. Tas nodrošina, ka sistēma tiek veidota ciešā sadarbībā ar skolas darbiniekiem un attīstās atbilstoši viņu reālajām vajadzībām.

**5. uzdevums: Īstenošanas varianti**

**Paskaidrojums:** Šoreiz piedāvāsim atšķirīgus variantus – pilnībā jauna sistēma pret gatavas platformas pielāgošanu.

**1. variants:**

* **Īss apraksts:** **Pielāgota tīmekļa platforma (Custom Web Platform)**, kas izstrādāta no nulles, lai precīzi atbilstu visām skolas "Do-Re-Mi" unikālajām prasībām. Tā nodrošina maksimālu elastību un nākotnes paplašināšanas iespējas.
* **Izvietošanas veids:** Privāts mākoņserveris (VPS - Virtual Private Server).
* **Programmēšanas valoda:** JavaScript (Node.js) servera daļai un JavaScript (Svelte/Vue.js) klienta daļai, izmantojot MongoDB datubāzi elastīgai datu struktūrai.
* **Koda redaktors:** Visual Studio Code.

**2. variants:**

* **Īss apraksts:** **Gatavas mācību pārvaldības sistēmas (LMS) pielāgošana.** Tiek izmantota atvērtā koda platforma, piemēram, **Moodle** vai **WordPress ar LMS spraudņiem** (piem., LearnDash). Tiek konfigurēti esošie moduļi nodarbībām, kalendāram, maksājumiem un saziņai. Risinājums ir ātrāks un lētāks sākotnēji, bet mazāk elastīgs specifiskām vajadzībām.
* **Izvietošanas veids:** Dalītā tīmekļa mitināšana (Shared Web Hosting), kas optimizēta WordPress/Moodle.
* **Programmēšanas valoda:** Galvenokārt konfigurēšana. Nelieliem pielāgojumiem – PHP, CSS.
* **Koda redaktors:** Nav nepieciešams IDE, darbs notiek caur platformas administrācijas paneli un failu redaktoru (piem., Notepad++).

Risinājums: Treniņuzdevums Nr. 3 (Zemnieku saimniecība "Zaļie Pirkstiņi")

**1. uzdevums: Iesaistītās puses un prasības**

1. **iesaistītā puse:** Saimniecības īpašnieks  
   **Papildu funkcionalitāte:** Pārdošanas analītikas panelis, kurā redzams, kuri produkti ir vispieprasītākie, kādi ir vidējie pasūtījuma apjomi un ienākumu dinamika.
2. **iesaistītā puse:** Klients  
   **Papildu funkcionalitāte:** Recepšu sadaļa, kur klienti var atrast idejas, kā pagatavot iegādātos produktus.
3. **iesaistītā puse:** Noliktavas darbinieks  
   **Papildu funkcionalitāte:** Svītrkodu skenēšanas iespēja (izmantojot mobilā tālruņa kameru), lai ātri pievienotu vai noņemtu produktu vienības no sistēmas.
4. **iesaistītā puse:** Piegādes kurjers  
   **Papildu funkcionalitāte:** Mobilajai ierīcei pielāgots piegādes saraksts ar klientu adresēm, kontaktiem un optimizētu maršrutu kartē.

**2. uzdevums: Mērķauditorija un izpētes metodes**

1. **prasības mērķauditorija:** Noliktavas darbinieks  
   **Izpētes metode:** **Novērošana un Intervija.** Vērojot, kā darbinieks šobrīd veic uzskaiti, var identificēt "šaurās vietas". Intervijā var noskaidrot specifiskas vēlmes, piemēram, vai būtu ērtāk lietot planšeti vai stacionāru datoru, un kādi ir lielākie izaicinājumi.
2. **prasības mērķauditorija:** Saimniecības klienti  
   **Izpētes metode:** **Konkurentu analīze.** Izpētot citus līdzīgus pārtikas piegādes e-veikalus, var identificēt labās prakses piemērus un populāras funkcijas (piemēram, "iecienītāko produktu" saraksts, abonēšanas iespējas), par kurām pēc tam varētu veikt mērķētu anketēšanu klientu vidū.

**3. uzdevums: Darba uzdevumi izstrādātājiem**

1. **darba uzdevums:** Izstrādāt publisku produktu katalogu ar meklēšanas, filtrēšanas un kārtošanas iespējām, kas reāllaikā attēlo datus no noliktavas datubāzes.
2. **darba uzdevums:** Implementēt pilnu e-komercijas ciklu: preču grozu, pasūtījuma noformēšanu, lietotāja reģistrāciju un drošu integrāciju ar tiešsaistes maksājumu apstrādes sistēmu (piem., Swedbank Payment Portal).
3. **darba uzdevums:** Izveidot administrācijas moduli, kas, balstoties uz piegādes dienā reģistrētajām adresēm, izmanto kartēšanas API (piemēram, Google Maps API), lai izveidotu optimālu piegādes maršrutu.

**4. uzdevums: Izstrādes modeļa izvēle**

**Izvēlētais modelis: Veiklais modelis (Agile) kombinācijā ar Prototipēšanu**

**Pamatojums:** E-komercijas projekts ir ļoti orientēts uz lietotāja pieredzi (UX).

* **Agile** pieeja ļaus projektu attīstīt pakāpeniski. Sākumā var izveidot minimālu, bet strādājošu veikalu (produktu katalogs un pasūtīšanas forma), lai ātri sāktu tirdzniecību. Pēc tam katrā sprintā var pievienot jaunas funkcijas – tiešsaistes maksājumus, recepšu sadaļu, atlaižu kodus utt.
* **Prototipēšana** ir kritiski svarīga, lai izstrādātu ērtu un intuitīvu iepirkšanās procesu. Pirms kodēšanas var izveidot interaktīvus saskarņu prototipus (piem., Figma) un testēt tos ar potenciālajiem klientiem, lai pārliecinātos, ka iepirkšanās process ir loģisks un patīkams.

**5. uzdevums: Īstenošanas varianti**

**Paskaidrojums:** Šeit labs kontrasts ir starp tīmekļa platformu un uz mobilo lietotni centrētu risinājumu.

**1. variants:**

* **Īss apraksts:** **Adaptīva dizaina e-komercijas vietne (Responsive Web Platform).** Tiek veidota viena platforma, kas automātiski pielāgojas jebkuram ekrāna izmēram – datoram, planšetei un viedtālrunim. Šis ir universāls un visizplatītākais variants.
* **Izvietošanas veids:** Mākoņplatforma (piemēram, Heroku) vai jaudīga tīmekļa mitināšana.
* **Programmēšanas valoda:** Ruby (ar Ruby on Rails ietvaru, kas ir ļoti populārs ātrai e-komercijas projektu izstrādei) vai PHP (izmantojot gatavu e-komercijas ietvaru kā Magento vai pielāgojot WooCommerce).
* **Koda redaktors:** Visual Studio Code, RubyMine.

**2. variants:**

* **Īss apraksts:** **Mobilā lietotne (Mobile-first Application).** Galvenais fokuss ir uz lietotni Android un iOS ierīcēm, caur kuru notiek visa pasūtīšana. Papildus tiek izveidota vienkārša tīmekļa lapa (landing page) ar informāciju un saitēm uz lietotņu veikaliem. Šī pieeja ļauj izmantot ierīces funkcijas (piem., push paziņojumus par jauniem produktiem).
* **Izvietošanas veids:** Lietotne tiek izvietota **Google Play Store** un **Apple App Store**. Servera daļa (API) tiek izvietota mākoņserverī.
* **Programmēšanas valoda:** **Flutter** vai **React Native**, lai ar vienu koda bāzi izveidotu lietotni abām platformām (Android un iOS). Servera daļai (API) – Node.js vai Python.
* **Koda redaktors:** Android Studio (ar Flutter spraudni) vai Visual Studio Code.