**1. Problēmas izpēte un analīze**

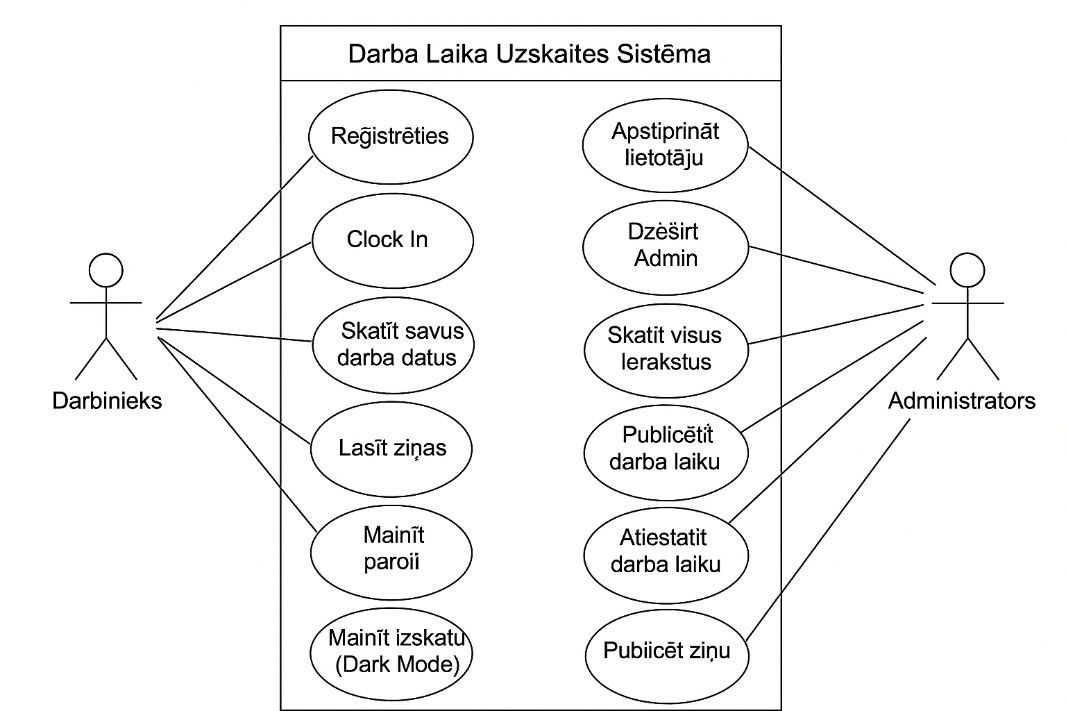
**Izpētes metodes izvēle un pamatojums**

Projekta mērķis ir izveidot efektīvu un lietotājam draudzīgu darbinieku darba laika uzskaites sistēmu, kas nodrošina precīzu laika pierakstu, ziņu publicēšanu un administrācijas iespējas. Lai sasniegtu šo mērķi, tika veikta rūpīga izpēte, izmantojot dažādas metodes un pieejas.

Izpētei tika izmantotas sekojošas metodes:

* **Literatūras un tehniskās dokumentācijas izpēte** – lai saprastu labākās prakses darba laika uzskaites sistēmu izstrādē un izmantotu modernākās tehnoloģijas.
* **Esošo risinājumu analīze** – izpētot eksistējošas sistēmas un to funkcionalitāti, lai identificētu galvenās stiprās un vājās puses.
* **Intervijas ar mērķauditoriju** – lai identificētu prasības un uzlabojumu iespējas, balstoties uz lietotāju pieredzi un vajadzībām.
* **Prototipu testēšana** – agrīnās versijas funkcionalitātes pārbaude, lai uzlabotu lietojamību un pielāgotu interfeisu.
* **Anketēšana** – veicot datu vākšanu no potenciālajiem lietotājiem, lai noteiktu funkcionalitātes prioritātes.

**Izpētes procesa apraksts**



Projekta izstrāde tika veikta šādos posmos:

1. **Sistēmas prasību analīze** – noteiktas funkcijas un mērķauditorijas vajadzības, kā arī izstrādāti lietotāju scenāriji.
2. **Datu apstrāde un struktūras izveide** – izveidotas tabulas SQLite datu bāzē un definēti attiecību modeļi.
3. **Izstrāde un testēšana** – koda rakstīšana un atkļūdošana, kā arī sākotnējie integrācijas testi.
4. **Lietotāju testēšana un pielāgošana** – atgriezeniskās saites ievākšana un uzlabojumu ieviešana.
5. **Galīgā optimizācija un drošības pārbaudes** – sistēmas stabilitātes un veiktspējas nodrošināšana.

**Izpētes datu apkopojums**

Balstoties uz izpēti, tika secināts, ka galvenās prasības ir:

* Droša autentifikācija un autorizācija, izmantojot šifrēšanas metodes.
* Darba laika reģistrēšana un administrācija ar precīzu datu uzglabāšanu.
* Ziņu publicēšanas sistēma, kas ļauj administrācijai izplatīt informāciju.
* Administrācijas panelis lietotāju pārvaldīšanai, nodrošinot iespēju piešķirt un atņemt piekļuves tiesības.
* Responsīvs un lietotājam draudzīgs interfeiss, kas pielāgojas dažādām ierīcēm.

**2. Programmatūras prasību specifikācija**

**Mērķauditorijas izvēle un raksturojums**

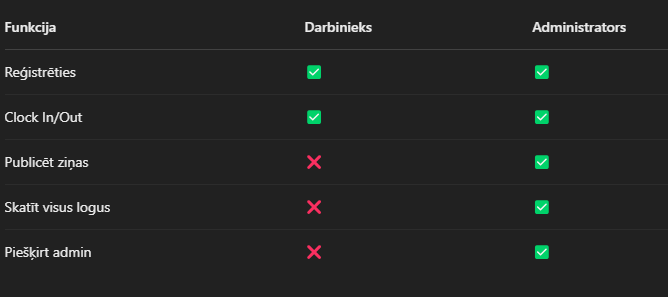
Sistēmas mērķauditorija:

* Uzņēmumi un iestādes ar vajadzību pēc darbinieku darba laika uzskaites, īpaši attālinātā darba kontekstā.
* Administratori, kas vēlas pārvaldīt lietotājus un publicēt ziņas, lai informētu darbiniekus.
* Darbinieki, kas veic laika uzskaiti un skatās jaunākās ziņas, izmantojot lietotājam draudzīgu platformu.

**2.1. Vispārīgs mērķis**

Programmatūras produkta mērķis ir izveidot pilnvērtīgu, drošu, lietotājam draudzīgu un efektīvu darbinieku darba laika uzskaites sistēmu, kas atbalsta darbinieku reģistrāciju, laika uzskaiti, ziņu publicēšanu un detalizētu administratīvo vadību.

**2.2. Detalizēts mērķauditorijas apraksts**

* **Darbinieki:**
  + Vēlas viegli reģistrēties un ātri saņemt piekļuvi darba laika uzskaitei.
  + Nepieciešama iespēja viegli uzsākt un noslēgt darba laika uzskaiti.
  + Pieprasa iespēju lasīt aktuālas ziņas no vadības.
* **Administratori:**
  + Nepieciešams pārskatāms lietotāju vadības panelis, lai varētu apstiprināt jaunus lietotājus un piešķirt pieejas tiesības.
  + Nepieciešama iespēja rediģēt un publicēt ziņas.
  + Nepieciešama detalizēta informācija par katra lietotāja darba laiku.
  + 
* **Vadības darbinieki:**
  + Nepieciešama iespēja pārskatīt darbinieku produktivitāti un darba stundu atskaites.

**2.3. funkcionālās prasības**

**2.3.1. Lietotāja reģistrācija un pieteikšanās**

* Lietotājs ievada lietotājvārdu (3-20 simboli).
* Lietotājs ievada drošu paroli (8-32 simboli, vismaz viens cipars, viens lielais burts).
* Sistēma pārbauda, vai lietotājvārds nav jau izmantots.
* Paroles tiek hash-otas izmantojot bcrypt algoritmu.
* Lietotāja konts tiek atzīmēts kā „neapstiprināts” līdz administratora apstiprinājumam.

**2.3.2. Darba laika uzskaite**

* Lietotājs pēc pieteikšanās nokļūst panelī, kur ir divas skaidri redzamas pogas: „Clock In” (zaļa), „Clock Out” (sarkana).
* Noklikšķinot „Clock In”, sistēma reģistrē precīzu datumu un laiku datubāzē.
* Lietotājs nevar nospiest „Clock In” divreiz pēc kārtas bez „Clock Out”.
* Noklikšķinot „Clock Out”, sistēma atjaunina ierakstu ar beigu laiku.
* Sistēma aprēķina kopējo nostrādāto laiku katram periodam.

**2.3.3. Administrācijas panelis**

* Administratoriem pieejams saraksts ar visiem lietotājiem, norādot lietotājvārdus, statusu (apstiprināts/neapstiprināts), admin statusu.
* Administratoram pieejamas pogas:
  + „Apstiprināt” – aktivizē lietotāju kontu.
  + „Dzēst” – neatgriezeniski dzēš lietotāju.
  + „Piešķirt Admin” / „Noņemt Admin” – maina lietotāja statusu.
* Administrators var redzēt visus darba laika ierakstus tabulā ar filtrēšanas iespējām pēc datuma, lietotāja.
* Ir iespējams atiestatīt darba laika ierakstus ar apstiprinājuma dialogu.

**2.3.4. Ziņu publicēšana**

* Administratoram pieejama ziņu sadaļa ar formu ziņu izveidei:
  + Virsraksts (maks. 100 simboli)
  + Saturs (maks. 5000 simboli)
* Ziņām pievienots automātisks laika zīmogs.
* Ziņas iespējams rediģēt un dzēst, pieejami apstiprināšanas dialogi pirms darbību veikšanas.

**2.3.5. Profila pārvaldība un iestatījumi**

* Lietotājs var mainīt paroli:
  + Jāievada pašreizējā parole un divreiz jaunā parole validācijai.
* Iespēja aktivizēt „Dark Mode”, izmantojot CSS un JavaScript.

**2.3.6. Drošības funkcijas**

* Visas paroles ir bcrypt šifrētas.
* Sesijas pārvaldīšana ar express-session.
* Lietotāju darbības tiek auditētas (pieteikšanās, clock-in/out, paroles maiņa).

**2.4. Detalizētas nefunkcionālās prasības**

* **Ātrdarbība:**
  + Sistēmas reakcija uz pieprasījumu nepārsniedz 500ms.
* **Drošība:**
  + Pilnvērtīga paroļu aizsardzība un sesiju pārvaldība.
  + Datubāzes aizsardzība pret SQL injekcijām.
* **Lietojamība:**
  + Skaidrs, minimālistisks un intuitīvs interfeiss.
  + Adaptīvs dizains, pielāgots mobilajām ierīcēm.

**2.5. Tehniskā infrastruktūra un izstrādes rīki**

* **Backend:** Node.js, Express.js, SQLite, Passport.js, bcrypt
* **Frontend:** HTML5, CSS3, Bootstrap, AJAX
* **Datubāze:** SQLite, iespējama migrācija uz MySQL vai PostgreSQL

**2.6. Testēšanas procedūras**

* **Unit Testing:** Tiek testētas atsevišķas funkcijas.
* **Integration Testing:** Testētas mijiedarbības starp sistēmas moduļiem.
* **User Acceptance Testing:** Testēšana reālos darba scenārijos.
* **Security Testing:** Pārbaudīta paroļu drošība, injekciju novēršana.

**2.7. Sistēmas uzturēšanas prasības**

* Dokumentācija tiek regulāri atjaunota pēc katras būtiskas izmaiņas.
* Administrators atbild par regulāru datu rezerves kopiju izveidi.
* Nepieciešami regulāri drošības auditi un sistēmas atjauninājumi.

**Programmatūras produkta un tā funkciju apraksts**

Sistēma piedāvā sekojošas funkcijas:

* **Lietotāja reģistrācija un pieteikšanās** – izmantojot Passport.js un bcrypt šifrēšanu.
* **Darba laika uzskaite** – clock-in un clock-out iespējas ar automātisku datu saglabāšanu.
* **Ziņu publicēšana** – administratori var pievienot un rediģēt ziņas, kas tiek parādītas visiem lietotājiem.
* **Administrācijas panelis** – lietotāju pārvaldība, admin lomu piešķiršana, tiesību maiņa un lietotāju apstiprināšana.
* **Piekļuves žurnāli** – iespēja sekot līdzi, kad lietotāji ir pieslēgušies un kādas darbības veikuši sistēmā.

**Programmatūras produkta skice**

Sistēma sastāv no:

1. **Servera puses (Node.js, Express, SQLite, bcrypt, Passport.js, dotenv)**
2. **Klienta puses (HTML, CSS, JavaScript, Bootstrap, AJAX)**
3. **Datu bāzes (SQLite) ar lietotāju, laika uzskaites un ziņu tabulām, iekļaujot audita žurnālus**

**3. Programmatūras izstrādes plāns**

1. **Prasību analīze un specifikācija** – 1.-2. nedēļa
2. **Sistēmas arhitektūras izstrāde un datu bāzes struktūras izveide** – 3. nedēļa
3. **Backend izstrāde un sākotnējā testēšana** – 4.-6. nedēļa
4. **Frontend izstrāde, responsīva dizaina pielāgošana** – 7.-8. nedēļa
5. **Integrācija un drošības testi** – 9.-10. nedēļa
6. **Pilnvērtīga testēšana, lietotāju atsauksmju ievākšana** – 11. nedēļa
7. **Dokumentācijas sagatavošana un nodošana** – 12. nedēļa

**4. Atkļūdošanas un akcepttestēšanas pārskats**

**Testēšanas metodes:**

* Vienības testēšana (unit tests) – veikta katra moduļa un funkcijas līmenī, lai garantētu, ka katra sistēmas sastāvdaļa darbojas pareizi neatkarīgi.
* Integrācijas testēšana – pārbaudīta moduļu savstarpējā mijiedarbība un komunikācija, nodrošinot datu korektu nodošanu starp komponentiem.
* Lietotāju testēšana – veikta reālos darba scenārijos, lai pārbaudītu lietotāja saskarnes intuitivitāti, funkcionalitāti un vispārējo lietošanas ērtumu.
* Drošības testēšana – īpaša uzmanība veltīta autentifikācijai, datu privātuma nodrošināšanai un sistēmas aizsardzībai pret biežāk sastopamiem kiberdrošības riskiem, piemēram, SQL injekcijām un paroļu kompromitēšanu.

**Būtiskākie labojumi un optimizācijas:**

* Lietotāju autentifikācijas sistēmas uzlabošana, ieviešot striktākas validācijas prasības un spēcīgāku paroļu politiku.
* Datu bāzes vaicājumu optimizācija, ieviešot indeksēšanu, lai samazinātu atbildes laikus un paaugstinātu sistēmas kopējo veiktspēju.
* Veikta padziļināta lietojamības testēšana, lai nodrošinātu sistēmas intuitīvu lietošanu.

**Būtiski uzlabojumi:**

* Sistēmas autentifikācijas modulis sākotnēji bija lēns un bieži atteica piekļuvi lietotājiem, īpaši intensīvas lietošanas laikā. Ieviešot sesiju pārvaldības optimizāciju un uzlabotu paroļu validāciju, šīs problēmas tika atrisinātas, būtiski uzlabojot lietotāju pieredzi.
* Ziņu publicēšanas modulis sākotnēji nepareizi saglabāja publicēšanas laika zīmogu, kas noveda pie haotiskas informācijas izvietošanas. Šī kļūda tika atrisināta, izmantojot automatizētu servera laika zīmoga validāciju un precīzāku laika ierakstīšanas sistēmu.

**Problēmas un to risinājumi:**

* Reģistrācijas procesa laikā datubāzes sistēma regulāri atgrieza kļūdas, jo lietotāju datu unikālās pārbaudes funkcija nedarbojās korekti. Pēc padziļinātas atkļūdošanas tika izveidota pilnīgi jauna pārbaudes funkcija, kas risināja šo problēmu.
* Ziņu sistēmas ieviešana prasīja vairākus mēģinājumus, jo sistēma nespēja uzturēt vienmērīgu ziņu rediģēšanas funkcionalitāti. Pēc detalizētas integrācijas testēšanas un kļūdu žurnalēšanas tika identificēts un novērsts problēmas cēlonis, uzlabojot gan datu bāzes struktūru, gan lietotāja interfeisa validācijas metodes.

**6. Piemērotās licences pamatojums**

Sistēma tiek izplatīta ar **MIT licenci**, kas nodrošina atļauju brīvi izmantot, modificēt un izplatīt kodu, vienlaikus saglabājot autortiesību paziņojumus.