PANKKIAUTOMAATTISOVELLUS

**PROJEKTISUUNNITELMA**

|  |  |
| --- | --- |
| **Versio** | *1.0* |

|  |  |
| --- | --- |
| **Ryhmä nro** | L19 |
| *Kirill Cheremisin* | |
| *Andrei Bobrenko* | |

# DOKUMENTIN TARKOITUS

Projektisuunnitelmassa esitellään pankkiautomaattisovelluksen kehitysprojekti. Ensisijaisena tavoitteena on toteuttaa pankkiautomaattijärjestelmä, joka vastaa tilaajan tarpeita Qt Widgets -sovelluksella, Node.js-palvelimella ja MySQL-tietokannalla. Toisena tärkeänä tavoitteena on kehittää projektiryhmän osaamista ohjelmistoprojektin toteuttamisesta sekä käytetyistä teknologioista (Qt, Node.js, MySQL).

Dokumentissa esitetään esitutkimuksen tulokset, järjestelmä- ja asiakasvaatimukset, projektin tehtävät (määrittely, suunnittelu, toteutus, testaus ja julkaisu), organisoituminen, käytännöt, rajaukset, aikataulu, laatutavoitteet ja riskit. Projektisuunnitelma kuvaa, miten tavoitteet saavutetaan käytettävissä olevilla resursseilla. Julkaisuksi riittää ryhmän loppuesitys.

Projektisopimuksen allekirjoituksina toimii dokumentin etusivu, jossa on ryhmän numero ja jäsenten nimet.

# ESITUTKIMUS

**Tavoitteet**

Esitutkimuksen tavoitteena oli kartoittaa pankkiautomaattijärjestelmän tuoteideaa ja vaatimuksia. Tutkimuksessa määriteltiin yleiset järjestelmätason vaatimukset ja alustavat asiakasvaatimukset. Keskeistä oli selvittää asiakkaan tarpeet ja kirjata ne riittävällä tarkkuudella. Esitutkimuksen perusteella päätettiin käynnistää toteutusprojekti.

**Tuotteen järjestelmävaatimukset**

Järjestelmään kuuluvat seuraavat fyysiset komponentit:

* **RFID-kortinlukija:** Kiinnitetty pankkiautomaatin tietokoneen sarjaporttiin Qt-sovelluksen ohjaamana.
* **RFID-kortit:** Toimivat järjestelmän pankkikortteina ja kommunikoivat lukulaitteen kanssa.
* **Pankkiautomaatin tietokone:** Sisältää käyttöjärjestelmän ja tietoturvaohjelman. Käyttöliittymä toteutetaan Qt Widgets -sovelluksella.
* **Palvelintietokone:** Käyttöjärjestelmänä Windows 11, Node.js-ympäristö palvelinlogiikalle ja MySQL-tietokanta. Sisältää tietoturvaohjelmapaketin.
* **Tietokoneverkko:** Tiedonsiirto automaatin ja palvelimen välillä käyttää HTTPS-protokollaa.
* **REST-rajapinta:** Node.js-palvelin tarjoaa REST API:n Qt-sovelluksen ja MySQL-tietokannan väliseen kommunikointiin.

**Alustavat asiakasvaatimukset**

Tarkemmat toiminnalliset vaatimukset ja käyttötapaukset selvitetään määrittelyvaiheessa asiakkaan kanssa. Alustavat vaatimukset:

• Debit kortti toteutettava. Debit kortilla ei ole luottoa eli tili ei saa mennä miinukselle. Jos saldo ei riitä käyttäjä saa siitä ilmoituksen.

• Kun Qt-sovellus käynnistetään, avautuu Aloituskäyttöliittymä

• Pankkiautomaatille kirjaudutaan laittamalla ”lätkä” kortinlukijaan jolloin avautuu Tunnuksen syöttökäyttöliittymä, jossa pyydetään PIN-koodi. Huom! Monimuotoryhmissä ei käytetä kortinlukijaa, vaan aloitusikkunasta avataan Tunnuksen syöttökäyttöliittymä painiketta painamalla, jossa annetaan PIN-koodin lisäksi kortin-id.

• Jos PIN-koodia ei anneta 10 sekunnin kuluessa palataan Aloituskäyttöliittymään

• Jos annetaan oikea PIN-koodi avautuu Pääkäyttöliittymä

• Jos annetaan väärä PIN-koodi, ilmoitetaan käyttäjälle ja kysytään uudestaan. Tässä ei vielä vaadita kortin lukitsemista.

• Käyttäjä voi katsoa saldonsa

• Käyttäjä voi nostaa, joko 20, 40, 50 tai 100 euroa.

• REST APIin toteutettu kaikki automaatilla tarvittavat toiminnot.

* Kukin kortti voi olla joko debit tai credit tyyppinen.
* Credit kortilla voi nostaa, kunhan luottoraja ei ylity. (Credit tileille on siis määritettävä luottoraja)
* Myös "muun summan" (käyttäjä voi siis näppäillä haluamansa summan) nosto mahdollinen.
* Käyttäjä voi myös katsoa 10 viimeistä tilitapahtumaansa
* REST APIin toteutettu kullekin taululle kaikki CRUD-operaatiot (vaikkei niitä tarvita automaatissa).
* Pankkiautomaatin kaikki toiminnot voidaan suorittaa painikkeilla eli siis ilman PC-näppäimistöä.
* Jos PIN-koodi annetaan 3 kertaa väärin, kortti lukitaan ja sitä ei voi enää käyttää. Tästä ilmoitetaan käyttäjälle, jonka jälkeen palataan ohjelman aloitus -käyttöliittymään.

**PROJEKTISUUNNITELMA**

**Toimittaja ja tilaaja**

* **Toimittaja:** OAMK Tietotekniikan opiskelijoiden kurssin TVT24SPL projektiryhmä nro 19. Vetäjänä toimii Kirill Cheremisin.
* **Tilaaja:** OAMK Tietotekniikan tutkinto-ohjelma, edustajana ohjaaja Etunimi Sukunimi.

**Rajaus**

Projekti toteuttaa pankkiautomaattisovelluksesta prototyypin, joka on rajattu toimimaan loppuesityksen ympäristössä. Käteisen nosto simuloidaan, eikä oikeaa rahaa käsitellä.

**Projektin vaiheistus ja aikataulu**

Projekti alkaa periodin ensimmäisenä päivänä ja päättyy viimeistään periodin viimeisenä päivänä (8 kalenteriviikkoa). Tavoite on luovuttaa projekti ohjaajan kanssa sovittuna päivänä periodin viimeisellä viikolla.

**Tehtävät:** Määrittely, suunnittelu, toteutus ja testaus

|  |  |
| --- | --- |
| PROJEKTIN VAIHE | AIKATAULU |
| Määrittelyvaihe | Periodin viikot 1-2 |
| Toteutus- ja testausvaihe | Periodin viikot 2-6 |
| Dokumentointi ja tuotteen toimitus tilaajalle | Periodin viikot 7-8 |

**Kokous- ja palaverikäytännöt**

* Projektiryhmä osallistuu lukujärjestykseen merkittyyn opetukseen ja kokoontuu viikoittain ohjaajan kanssa projektipalaveriin.
* Ennen palaveria ryhmä postaa Teamsiin viikkoraportin:
  + Valmistuneet tehtävät (esittelyvalmiina).
  + Seuraavan viikon suunnitellut tehtävät.
  + Kysymykset ja ongelmat.
  + Versionhallinnan päivitys ajan tasalle.
  + Mahdollinen pyyntö lisäohjaukselle.

**Tiedonvälitys**

* Tiedonvälitys hoidetaan tapaamisissa, Teams-viesteillä ja tarvittaessa sähköpostilla.
* Vastuu sisäisestä tiedottamisesta on projektiryhmän vetäjällä.
* Ohjaajalle osoitetut Teams-viestit tagataan hänen nimellään.

**Laatutavoitteet**

* Noudatetaan sovittuja käytäntöjä ja dokumenttipohjia.
* Koodi ja dokumentaatio pidetään selkeänä ja luettavana.
* Suoritetaan katselmointeja ryhmän kesken ja ohjaajan kanssa, korjaukset tehdään nopeasti.
* Tehtävät jaetaan selkeisiin kokonaisuuksiin (esim. Qt-käyttöliittymä, Node.js REST API, MySQL-tietokanta).
* Testaus noudattaa V-mallia: yksikkötestaus (oma koodi), integrointi- ja järjestelmätestaus (yhdessä), hyväksymistestaus (ennen julkaisua).
* Ryhmä ratkoo ongelmia yhdessä.

**Projektin päätöstoimet**

* Projekti päätetään loppuesityksellä periodin viimeisellä viikolla.
* Dokumentaatio ja koodi (versionhallinnassa) on valmiina ja katselmoituna.
* Itse- ja toveriarviointi tehdään ennen esitystä.
* Lainatut laitteet (RFID-lukija ja kortit) palautetaan ohjaajalle.

# RISKIENHALLINTA

Riskien hallinta jakautuu riskien tunnistamiseen, niiden analysointiin ja niihin varautumiseen laatimalla ratkaisuehdotukset. Tässä projektissa projektiryhmä on kartoittanut seuraavat riskit.

**Riskilista ja riskien analyysitaulukot**

|  |  |
| --- | --- |
| **Riskinumero** | **Riski** |
| 1 | Projektin tehtävien valmistuminen projektin aikataulun mukaisesti. |
| 2 | Toiminnalliset vaatimukset ovat epäselviä. |
| 3 | Projektin viestintä ei toimi. |
| 4 | Palvelimen ja tietokannan yhteys katkeilee. |

**RISKI 1**

|  |  |
| --- | --- |
| **Riskialkio** | Projektin tehtävien valmistuminen projektin aikataulun mukaisesti. |
| **Riskitekijä** | Ryhmässä ei ole sovittu miten kommunikoidaan päivittäin, miten yhdessä autetaan toisia tarvittaessa projektin tehtävien tekemisen yhteydessä. Ryhmän jäsenen sitoutuminen projektiin. |
| **Riskitapahtuma** | Ryhmän jäsen ei tee projektin tehtäviä sovitun mukaisesti. |
| **Riskin seuraamus** | Ryhmän jäsenelle osoitetut tehtävät jäävät tekemättä, ja muut joutuvat ottamaan tehtävät tehtäväksi. Projektin tehtäviä ei ehditä tekemään aikataulun mukaisesti. |
| **Toimenpiteet/ratkaisu** | Sovitaan päivittäisestä yhteydenpidosta, ja siitä, että miten autetaan toisia ryhmän jäseniä tehtävien tekemisessä. |

**RISKI 2**

|  |  |
| --- | --- |
| **Riskialkio** | Toiminnalliset vaatimukset ovat epäselviä. |
| **Riskitekijä** | Toiminnalliset vaatimukset kirjoitetaan  epäselvästi ja liian yleisellä tasolla. |
| **Riskitapahtuma** | Toiminnallisia vaatimuksia ei täysin ymmärretä, koska niissä ole tarpeeksi riittävää informaatiota  vaatimuksen toteuttamiselle. Vaatimuksista joudutaan keskustelemaan uudestaan toteutusvaiheessa, ja ne joudutaan  kirjoittamaan uudestaan. |
| **Riskin seuraamus** | Koska toiminnalliset vaatimukset ovat  epäselviä, niin niitä joudutaan päivittämään.  Tämä vie aikaa projektin toteutusvaiheelta, ja  alkaa myöhästyttämään projektia. |
| **Toimenpiteet/ratkaisu** | Projektin ohjaavaa opettaja käy ryhmän kanssa  toiminnalliset vaatimukset läpi Trellossa. |

**RISKI 3**

|  |  |
| --- | --- |
| **Riskialkio** | Projektin viestintä ei toimi. |
| **Riskitekijä** | Projektiryhmän jäsenet eivät kommunikoi projektissa sovitulla tavalla. |
| **Riskitapahtuma** | Ryhmän jäsenet ja ohjaava opettaja eivät tiedä missä tilassa projekti on. |
| **Riskin seuraamus** | Projektin tehtävien tila on epäselvä. |
| **Toimenpiteet/ratkaisu** | Kommunikoidaan projektissa sovitulla tavalla päivittäin. |

**RISKI 4**

|  |  |
| --- | --- |
| **Riskialkio** | Palvelimen ja tietokannan yhteys katkeilee. |
| **Riskitekijä** | Node.js REST API:n ja MySQL:n konfigurointi epäonnistuu. |
| **Riskitapahtuma** | Tilitapahtumat tai saldo ei näy. |
| **Riskin seuraamus** | Sovellus ei toimi odotetusti. |
| **Toimenpiteet/ratkaisu** | Testataan yhteys erillisenä osana ennen integrointia. |

**RISKI 5**

|  |  |
| --- | --- |
| **Riskialkio** | RFID-lukijan integrointi epäonnistuu |
| **Riskitekijä** | Sarjaportin käyttö Qt:ssa on haastavaa. |
| **Riskitapahtuma** | Kortinlukeminen ei toimi. |
| **Riskin seuraamus** | Kirjautuminen ei onnistu, perustoiminto puuttuu. |
| **Toimenpiteet/ratkaisu** | Testataan lukija ajoissa ja konsultoidaan ohjaajaa. |