<<PROJEKTIN NIMI>>

**TEKNINEN MÄÄRITTELY**

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

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

# TEKNINEN MÄÄRITTELY

Tämä dokumentti sisältää pankkiautomaattiohjelmiston toiminnallisen ja teknisen määrittelyn. Tarkoituksena on esittää:

1. Pankkiautomaatin käyttötapaukset, toiminnallisuudet, vaaditut ominaisuudet ja tietomalli.
2. Järjestelmäarkkitehtuuri, komponentit (Qt-sovellus, REST API, MySQL-tietokanta) ja niiden toteutusratkaisut.
3. Käyttöliittymän lopullinen toteutus kuvakaappauksilla ja tilakaaviolla.

Dokumentissa käytetään UML-kaavioita (käyttötapaus-, komponentti-, luokka- ja tilakaaviot). Tietokanta kuvataan ER-kaaviolla.

# JÄRJESTELMÄARKKITEHTUURI

**Kuvaus**

Järjestelmäarkkitehtuuri esitetään UML-käyttöönottokaaviona:

* **Qt-sovellus:** Käyttöliittymä Windows-ympäristössä.
* **REST API (Express.js):** Node.js-pohjainen rajapinta tietokannan ja sovelluksen välillä.
* **MySQL-tietokanta:** Tallentaa asiakas-, tili- ja tapahtumatiedot.

|  |
| --- |
|  |

**Yleiskuvaus**

1. Pankkiautomaatti toimii Windows-tietokoneella, jossa on tuore Windows-käyttöjärjestelmä ja tietoturvaohjelma (esim. Windows Defender). Ohjelmisto koostuu:
2. **Qt-sovellus:** EXE-tiedosto, joka käyttää Qt-luokkakirjastoa käyttöliittymään ja tapahtumapohjaiseen toimintaan. Kaikki toiminnot suoritetaan painikkeilla ilman PC-näppäimistöä.
3. **REST API:** Node.js- ja Express.js-pohjainen rajapinta, joka tarjoaa kaikki automaatin toiminnot ja CRUD-operaatiot tietokannalle.
4. **MySQL-tietokanta:** Tallentaa tiedot, mukaan lukien korttityypit (debit/credit) ja luottorajat credit-korteille.
5. Käyttäjä kirjautuu syöttämällä kortin ID:n ja PIN-koodin käyttöliittymään.

# KÄYTTÖTAPAUKSET

Pankkiautomaattiohjelmiston keskeiset toiminnot on lueteltu alla. Näistä toiminnoista johdetaan UML-mallinnuskielen käyttötapauskaavio(t) ja jokaisesta käyttötapauksesta käyttötapauskortti.

|  |  |
| --- | --- |
| TOIMINNON NIMI | TOIMINNON KUVAUS |
| Kirjaudu sisään | Kirjautua pankkiautomaatin käyttäjäksi RFID-kortin ja tunnusluvun avulla. |
| Näytä saldo | Näyttää tilin omistajan tiedot, viisi viimeistä tilitapahtumaa ja tilin saldon. |
| Selaa tilitapahtumia | Näyttää näytöllä tilitapahtumia käyttäjän selausvalintojen mukaisesti. |
| Nosta rahaa | Pankkiautomaatti luovuttaa käyttäjälle hänen nostaman summan rahaa, ja vähentää nostetun rahamäärän käyttäjän tililtä. |
| Kirjaudu ulos | Kirjata pankkiautomaatin käyttäjä ulos automaatista. |

Järjestelmän käyttötapauskaavio on allaolevan kuvan mukainen.

|  |
| --- |
|  |

**Käyttötapauskortit**

Käyttötapauskorteilla määritellään tarkemmin mitä jokaisessa toiminnossa tulee

tapahtua. Käyttötapauskorteista voidaan näin ollen johtaa pankkiautomaattijärjestelmän toiminnalliset vaatimukset.

|  |  |
| --- | --- |
| **Nimi** | **Kirjaudu sisään** |
| **Suorittajat** | pankkiautomaattisovellus |
| **Tavoite** | Kirjautua pankkiautomaatin käyttäjäksi |
| **Esiehdot** | Automaatin tietokone toimii, yhteys tietokantaan on kunnossa ja tietokanta on toiminnassa |
| **Kuvaus** | 1. Qt-sovellus käynnistyy Aloituskäyttöliittymään. 2. Käyttäjä painaa "Kirjaudu"-painiketta, jolloin avautuu Tunnuksen syöttökäyttöliittymä. 3. Käyttäjä käyttää RFID-kortin ja sitten syöttää PIN-koodin painikkeilla.   3.1 Jos PIN-koodia ei syötetä 10 sekunnissa, palataan Aloituskäyttöliittymään.   1. REST API tarkistaa tiedot tietokannasta.   4.1. Väärä PIN: Ilmoitus käyttäjälle, pyydetään uudelleen.  4.2. 3 väärää PIN-koodia: Kortti lukitaan, ilmoitus käyttäjälle, paluu Aloituskäyttöliittymään.   1. Oikea PIN: Avautuu Pääkäyttöliittymä, jossa näkyy asiakkaan nimi ja toiminnot. |
| **Loppuehdot** | Käyttäjä on kirjautunut sisään. |
| **Poikkeukset** | Väärä PIN tai kortin lukitus. |

|  |  |
| --- | --- |
| **Nimi** | **Näytä saldo** |
| **Suorittajat** | Pankkiautomaattisovellus |
| **Tavoite** | Näyttää tilin omistajan tiedot ja tilin saldon |
| **Esiehdot** | Kirjauduttu pankkiautomaatin käyttäjäksi, yhteys tietokantaan on kunnossa ja tietokanta on toiminnassa |
| **Kuvaus** | 1. Pääkäyttöliittymässä painetaan "Näytä saldo" -painiketta.  2. REST API hakee saldon, käyttöliittymä näyttää sen.  3. "Paluu"-painikkeella takaisin Pääkäyttöliittymään. |
| **Loppuehdot** | Saldo näytetty oikein. |

|  |  |
| --- | --- |
| **Nimi** | **Selaa tilitapahtumia** |
| **Suorittajat** | Pankkiautomaattisovellus |
| **Tavoite** | Näytetään näytöllä 10 viimeisintä tilitapahtumaa käyttäjän selausvalintojen mukaisesti. |
| **Esiehdot** | Kirjauduttu pankkiautomaatin käyttäjäksi, yhteys tietokantaan toimii ja tietokanta on toiminnassa. |
| **Kuvaus** | 1. Pääkäyttöliittymässä painetaan "Näytä tilitapahtumat" -painiketta. 2. REST API hakee 10 viimeistä tapahtumaa, näytetään käyttöliittymässä.. 3. "Paluu"-painikkeella takaisin Pääkäyttöliittymään. |
| **Loppuehdot** | Tilin omistajan tiedot ja tilitapahtumat on näytetty oikein, ja tilitapahtumia voidaan selata. |

|  |  |
| --- | --- |
| **Nimi** | **Nosta rahaa** |
| **Suorittajat** | Pankkiautomaattisovellus |
| **Tavoite** | Nostaa rahaa (20, 40, 50, 100 € tai muu summa) |
| **Esiehdot** | Kirjauduttu pankkiautomaatin käyttäjäksi, yhteys tietokantaan on kunnossa ja tietokanta on toiminnassa. |
| **Kuvaus** | 1. Pääkäyttöliittymässä painetaan "Nosta rahaa" -painiketta. 2. Näytöllä painikkeet: 20€, 40€, 50€, 100€, "Muu summa". 3. "Muu summa": Käyttäjä syöttää summan painikkeilla. 4. REST API tarkistaa:  * Debit: Saldo riittää (ei miinukselle). * Credit: Luottoraja ei ylity. * 4.1 Jos ei riitä, ilmoitus käyttäjälle, paluu valintaan.  1. Nosto onnistuu: REST API päivittää saldon, vahvistus näytöllä. 2. "Paluu"-painikkeella takaisin Pääkäyttöliittymään. |
| **Loppuehdot** | Käyttäjä on saanut nostetuksi haluamansa summan rahaa ja rahamäärä on veloitettu käyttäjän tilitä. |

|  |  |
| --- | --- |
| **Nimi** | **Kirjaudu ulos** |
| **Suorittajat** | Pankkiautomaattisovellus |
| **Tavoite** | Lopettaa pankkiautomaatin käyttäminen ja kirjautua ulos järjestelmästä. |
| **Esiehdot** | Kirjauduttu pankkiautomaatin käyttäjäksi, yhteys tietokantaan on kunnossa ja tietokanta on toiminnassa. |
| **Kuvaus** | 1. Pääkäyttöliittymässä painetaan Kirjaudu ulos –painiketta  2. Tietokantayhteys suljetaan ja käyttäjä kirjataan ulos pankkiautomaatista.  3. Palataan ohjelman aloituskäyttöliittymään. |
| **Loppuehdot** | Pankkiautomaatin yhteys tietokantaan on suljettu, käyttäjä on kirjattu ulos automaatista. |

**TIETOSISÄLTÖ**

Tässä kuvataan pankkiautomaatiohjelman ja -järjestelmän käsittelemä tieto.

**Käsiteanalyysi**

* **Yksi kortti kuuluu yhdelle asiakkaalle.**
* **Asiakkaalla voi olla monta tiliä (debit/credit).**
* **Kortti on joko debit (ei miinukselle) tai credit (luottorajalla).**

**Käsiteltävät tiedot**

* **Asiakas:** 
  + **Asiakkaan tunnus**
  + **Etunimi**
  + **Sukunimi**
* **Tili:** 
  + **Tilinumero**
  + **Saldo**
  + **Käyttäjän ID**
* **Kortti:** 
  + **Korttinumero (ID)**
  + **Tilinumero**
  + **PIN-koodi**
  + **Kortin tyyppi**
  + **Luottoraja**
* **Tilitapahtumat:** 
  + **Tilinumero**
  + **Päivämäärä ja kellonaika**
  + **Tili**
  + **Summa**

**Käsitemalli**

Käsiteanalyysin perusteella laadittu tietokannan rakennetta kuvaava ER-mallikaavio liitetään tähän.

|  |
| --- |
|  |

# JÄRJESTELMÄN KOMPONENTIT

Tässä esitetään ensin järjestelmäarkkitehtuuria tarkentava UML-komponenttikaavio. Kaaviossa kuvataan miten järjestelmäkomponentit toteutetaan ohjelmistokomponentten avulla, mitkä ovat niiden käyttämät rajapinnat ja muut yhteydet.

|  |
| --- |
|  |

# KOMPONENTTIEN KUVAUKSET

***Qt-Sovellus***

|  |  |  |
| --- | --- | --- |
| **Tarkoitus ja toiminta** | | Sovellus toimii pankkiautomaattisovelluksena, se odottaa RFID-korttia, pyytää käyttäjää syöttämään PIN-koodi ja sitten lähettää tiedot palvelimelle. Vastaus näytetään visuaalisessa muodossa Qt-sovelluksessa. |
| **Järjestelmäkomponentti** | | Qt-Sovellus |
| **Luokkakaavio (UML)** | | |
| **Rajapinnat**   |  |  | | --- | --- | | **Funktio / Signaali** | **Tarkoitus** | | MainWindow:: readSerialData() | Lukee RFID-kortin tiedot ja käsittelee ne | | PinInputWindow:: onNumberButtonClicked () | Lisää numeron PIN-koodiin | | PinInputWindow:: onNetworkReply() | Käsittelee palvelimelta saadun vastauksen | | PinInputWindow::onClearButtonClicked() | Tyhjentää PIN-koodin | | |  | | --- | | PinInputWindow::onSubmitButtonClicked() |  |  | | --- | |  | | |  | | --- | | Lähettää PIN-koodin REST API:lle |  |  | | --- | |  | | | signal::authenticationCompleted() | Ilmoittaa PIN-tarkistuksen onnistumisesta ja lähettää käyttäjätiedot seuraavalle ikkunalle | | |  | | --- | | WelcomeWindow:: onWithdrawalClicked () |  |  | | --- | |  | | |  | | --- | | Nostotoiminnon |  |  | | --- | |  | | | |  | | --- | | WelcomeWindow:: onTopUpClicked () |  |  | | --- | |  | | |  | | --- | | Tilin lisäystoiminnon |  |  | | --- | |  | | | |  | | --- | | WelcomeWindow:: onBalanceClicked () |  |  | | --- | |  | | Näyttää saldon | | |  | | --- | | WelcomeWindow:: onHistoryClicked () | | Tapahtumahistoria | | |  | | --- | | signal:: actionCompleted() | | Ilmoittaa, että toiminto on suoritettu ja voidaan palata alkuvalikkoon | | ActionWindow:: onSubmitClicked() | Lähettää toimenpiteen nosto /lisäys palvelimelle | | ActionWindow:: onCancelClicked() | Peru toiminnon | | ActionWindow:: onCloseClicked() | Sulkee ikkunan | | ActionWindow:: onNetworkReply() | Käsittelee palvelimen vastauksen | | ActionWindow:: onReAuthenticationCompleted() | Varmistaa käyttäjän uudelleen | | signal:: actionFinished() | Ilmoittaa, että toimenpide on valmis | | signal:: authenticationCompletedForAction () | Lähettää tokenin eteenpäin, kun uudelleenautentikointi onnistuu | | ActionWindow:: onReturnButtonClicked () | Palaa takaisin toimintovalikkoon | | signal:: returnToActionWindow () | Lähettää tapahtuman paluuta varten WelcomeWindow | | signal:: topUpCompleted () | Ilmoittaa, että tilin lisäys on suoritettu onnistuneesti | | | |
| **Vastuuhenkilö(t)** | Suunnitelu: Andrei Bobrenko  Toteutus: Andrei Bobrenko, Kirill Cheremisin  Testaus: Kirill Cheremisin  Dokumentointi: Andrei Bobrenko | |

***MySQL-Tietokanta***

|  |  |  |
| --- | --- | --- |
| **Tarkoitus ja toiminta** | | MySQL-tietokanta tallentaa tiedot, mukaan lukien korttityypit (debit/credit) ja luottorajat credit-korteille, mm. tallentaa asiakas-, tili- ja tapahtumatiedot*.* |
| **Järjestelmäkomponentti** | | MySQL-Tietokanta |
| **Luokkakaavio (UML)** | | |
| **Rajapinnat** *Funktiot mitä tarjoaa muille / signaalit mitä komponentti vastaanottaa tai lähettää muille. (ryhmän itse toteuttamat, ei Qt:n sisäisiä.)*   |  |  | | --- | --- | | **Funktio / Signaali** | **Tarkoitus** | | 'SELECT card\_id, pin\_hash, account\_id, card\_type, credit\_limit FROM cards WHERE card\_number = ?', [card\_number] | Tarkistetaan, onko korttinumero jo olemassa cards-taulussa | | 'SELECT balance FROM accounts WHERE account\_id = ?', [cardRows[0].account\_id] | Tarkistetaan tilin saldo | | 'UPDATE accounts SET balance = ? WHERE account\_id = ?', [newBalance, cardRows[0].account\_id] | Paivitetaan tilin saldo | | 'INSERT INTO transactions (transaction\_time, summa, account\_id) VALUES (NOW(), ?, ?)', [-withdrawalAmount, cardRows[0].account\_id] | Transaktio transaktiotaulukkoon | | |  | | --- | | 'SELECT card\_id, pin\_hash, account\_id FROM cards WHERE card\_number = ?', [card\_number] |  |  | | --- | |  | | |  | | --- | | Tarkistetaan, onko korttinumero jo olemassa cards-taulussa |  |  | | --- | |  | | | |  | | --- | | 'SELECT balance FROM accounts WHERE account\_id = ?', [cardRows[0].account\_id] |  |  | | --- | |  | | |  | | --- | | Saadaan tilin saldo |  |  | | --- | |  | | | |  | | --- | |  |   'SELECT account\_id, customer\_id, balance FROM accounts' | |  | | --- | | Haetaan kaikki tilit accounts-taulusta |  |  | | --- | |  | | | | |
| **Vastuuhenkilö(t)** | Suunnitelu: Kirill Cheremisin  Toteutus: Andrei Bobrenko  Testaus: Andrei Bobrenko, Kirill Cheremisin  Dokumentointi: Andrei Bobrenko, Kirill Cheremisin | |

***REST API***

|  |  |  |
| --- | --- | --- |
| **Tarkoitus ja toiminta** | | Se tarjoa pankkipalveluita, kuten rahan nostoa, saldon tarkistusta ja tilitapahtumien hallintaa verkkorajapinnan kautta. Toiminta perustuu kortin ja PIN-koodin tunnistukseen, ja kaikki tapahtumat tallennetaan tietokantaan turvallisesti. |
| **Järjestelmäkomponentti** | | REST API |
| **Luokkakaavio (UML)** | | |
| **Rajapinnat**   |  |  | | --- | --- | | **Funktio / Signaali** | **Tarkoitus** | | router.get('/get\_accounts') GET | Hakee kaikki tilit tietokannasta ja palauttaa ne JSON-muodossa | | router.post('/create\_accounts') POST | Luo uuden tilin | | router.put('/:accountId') PUT | Paivitetaan tilin saldo | | router.delete('/:accountId') DELETE | Poistaa tilin tietokannasta | | |  | | --- | | router.post('/auth') POST |  |  | | --- | |  | | |  | | --- | | Tarkistetaan korttinumeron ja pinkoodin ja palauttaa JWT-tokenin ja asiakastiedotja |  |  | | --- | |  | | | |  | | --- | | router.get('/get\_cards') GET |  |  | | --- | |  | | |  | | --- | | Hakee kaikki kortit tietokannasta ja palauttaa ne JSON-muodossa |  |  | | --- | |  | | | |  | | --- | |  |   router.post('/create\_card' POST | |  | | --- | | Luo uuden kortin |  |  | | --- | |  | | | router.put('/:cardNumber') PUT | Päivittää olemassa olevan kortin tiedot | | router.delete('/:cardNumber') DELETE | Poistaa kortin | | router.get('/get\_customers') GET | Hakee kaikki asiakkaat tietokannasta ja palauttaa listan JSON-muodossa | | router.post('/create\_customers') POST | Luo uuden asiakkaan | | router.put('/:customerId') PUT | Päivittää asiakkaan | | router.delete('/:customerId') DELETE | Poistaa asiakkaan ID:n perusteella tietokannasta | | router.get('/') GET | Palauttaa etusivun ja näyttaa index-näkymän | | router.post('/withdraw') POST | Nostaa rahaa tililtä, tarkistaa PIN-koodin ja päivittää saldon | | router.post('/balance') POST | Palauttaa tilin saldon | | router.post('/get\_transactions') POST | Hakee tilin kaikki tapahtumat | | router.post('/top\_up') POST | Lisää rahaa tilille | | router.put('/:transactionId) PUT | Päivittää tapahtuman summan | | router.delete('/:transactionId') DELETE | Poistaa tapahtuman | | | |
| **Vastuuhenkilö(t)** | Suunnitelu: Andrei Bobrenko, Kirill Cheremisin  Toteutus: Andrei Bobrenko, Kirill Cheremisin  Testaus: Andrei Bobrenko, Kirill Cheremisin  Dokumentointi: Andrei Bobrenko, Kirill Cheremisin | |

***DLL***

|  |  |  |
| --- | --- | --- |
| **Tarkoitus ja toiminta** | | DLL kirjasto hallita RFID-kortinlukijaa sarjaportin kautta elikkä kirjasto avaa sarjaportin, kuuntelee kortinlukijan lähettämää dataa ja välittää luetut korttinumerot kutsuvalle sovellukselle. |
| **Järjestelmäkomponentti** | | DLL |
| **Luokkakaavio (UML)** | | |
| **Rajapinnat**   |  |  | | --- | --- | | **Funktio / Signaali** | **Tarkoitus** | | PrintDebugMessage() | Tulostaa että toiminto suoritettu | | SetCardReadCallback(callback, userData) | Asettaa takaisinkutsun kortin lukemisen käsittelyyn | | InitReader(portName) | Alustaa RFID-lukijan ja avaa sarjaportin | | StartCardReading() | Käynnistää kortinlukijan ja alkaa kuunnella dataa | | |  | | --- | | StopCardReading() |  |  | | --- | |  | | |  | | --- | | Pysäyttää kortinlukijan ja vapauttaa resurssit |  |  | | --- | |  | | | |  | | --- | | RFIDReader::init(portName) |  |  | | --- | |  | | |  | | --- | | Alustaa ja avaa sarjaportin kortinlukijalle |  |  | | --- | |  | | | |  | | --- | |  |   RFIDReader::startReading() | |  | | --- | | Aloittaa kortinlukijan kuuntelun |  |  | | --- | |  | | | RFIDReader::stopReading() | Pysäyttää kuuntelun ja sulkee sarjaportin | | RFIDReader::readSerialData() | Käsittelee sarjaportilta saapuvan datan ja kutsuu callbackia | | | |
| **Vastuuhenkilö(t)** | Suunnitelu: Kirill Cheremisin  Toteutus: Kirill Cheremisin  Testaus: Kirill Cheremisin  Dokumentointi: Andrei Bobrenko | |

# KÄYTTÖLIITTYMÄN KUVAUS

## Tilakaavio

*Tilakaaviossa kuvataan käyttöliittymän toiminta pääpiirteissään.*

|  |  |
| --- | --- |
| *Kuva tähän.* |  |

## Ohjelman käyttöliittymät

Kuvakaappaukset ryhmän toteutuneesta sovelluksesta.

|  |  |
| --- | --- |
| *Kuvat tänne.* |  |
|  |  |

# LIITTEET

Mahdollinen ryhmän oma lisädokumentaatio mitä projektin aikana on syntynyt.