TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science

Khasanboy Akbarov 144812IVSM

A SELF SERVICE POS SYSTEM USING RFID AUTHENTICATION

Master's Thesis

Supervisor: Juhan-Peep Ernits

PhD

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Khasanboy Akbarov

RFID AUTENTIMIST KASUTAV ISETEENINDUSLIK KASSATERMINAL

Magistritöö

Juhendaja: Juhan-Peep Ernits
PhD

Tallinn 2017

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Khasanboy Akbarov

08.05.2017

Abstract

Using radio frequency identification (RFID) tags is widespread because of the convenience to the user. The technology is often used for access control in work places and in today's world almost everybody carries some RFID.

Point of sale (POS) systems are present in every context where some items or services need to be traded. The idea of a POS system that uses RFID for customer authentication is not new, but creating a system that is convenient for people in a particular context to use requires engineering effort.

In this work, we develop a simple self service POS system for staff members of the Department of Software Science of TTU that enables staff members to register themselves into the system using their staff ID cards, public transportation cards or any other RFID card that can provide a constant user ID (UID). As a customer, a staff member can initially register himself/herself with an RFID card and upon completion of the registration is able to record the purchase of items in common rooms using the application developed in the context of the current thesis. Raspberry Pi computers with touch screens and RFID readers are used as terminals in the system. The administrative interface that allows the system to be managed runs on a separate web server.

The current thesis covers the software development lifecycle of the system from of software requirements specification, software design description until software implementation and deployment.

This thesis is written in English and is 44 pages long, including 6 chapters, 13 figures and 2 tables.

Annotatsioon

RFID autentimist kasutav iseteeninduslik kassaterminal

Kontaktivabad kiipkaardid ja -märgid (RFID) on oma kasutusmugavuse tõttu laialdases kasutuses. Tehnoloogiat kasutataks muuhulgas tihti ka ligipääsude haldamiseks töökohtadel. Seega võib öelda, et tänapäeva maailmas kannab praktiliselt igaüks kaasas mõnd kontaktivaba kiipkaarti.

Kasssaterminale leidub kõigis kontekstides, kus on vaja järge pidada mingite toodete või teenuste müügil. Kontaktivaba kiipkaardiga kliendi autentimine kassaterminalis ei ole uudne, kuid konkreetsesse konteksti sobiva süsteemi loomine eeldab insenerlikku lähenemist.

Käesolevas töös töötasime TTÜ Tarkvarateaduse instituudi töötajatele välja ja implementeerisime lihtsa iseteenindusliku kassaterminali, mis võimaldab töötajal ennast töötõendit või mõnd muud kontaktivaba kiipkaarti kasutades süsteemi registreerida, ning seejärel märkida üles oma kohvitoas sooritatud ostud. Kassasüsteemi terminalid on realiseeritud Raspberry Pi, puutetundliku ekraani ja RFID lugeja abil. Administratiivne liides on veebipõhine ja paikneb eraldiseisvas veebiserveris.

Käesolev magistritöö katab tarkvaraarenduse protsessi alates tarkvara nõuete spetsifitseerimisest tarkvara disaini ja realiseerimiseni ning süsteemi käikuandmiseni.

Lõputöö on kirjutatud [mis keeles] keeles ning sisaldab teksti 44 leheküljel, 6 peatükki, 13 joonist, 2 tabelit.

List of abbreviations and terms

TTU	Tallinn University of Technology
NFC	Near Field Communication
RFID	Radio-frequency identification
CRUD	Create Read Update Delete
DESC	Description
RAT	Rationale
DEP	Dependency
SRS	Software Requirements Specification
SDS	System Design Specification
POS	Point of Sale

Table of Contents

List of figures

List of tables

1 Introduction

POS systems are developed to make business operations to be completed easily and are used in places where customer makes payments for products or services offered by the merchant. The merchant can be a shop or a company that sells products or provides services. There are different kinds of POS systems available in the market and they provide different capabilities for the merchants and customers. As an example of a POS system, we can take a terminal in a shop where customer pays for products he/she is buying. In this kind of terminal POS system has reader to read the barcode on the product and calculates the total amount of money customer should pay. Moreover, it enables the customer to pay for products by using bank transactions.

In this work, we develop a simple POS system for School of Information Technologies of TTU, where faculty members can buy items in the common rooms using terminals based on Raspberry Pis in common rooms. This system does not have a counter to count the amount of items user took and it cannot force faculty members to use this system in order to record consumed items. However, this system is a self service POS system and it is assumed to be used in a group of honest customers.

This system enables faculty members to register themselves to the system by using their staff ID cards, public transportation cards or any other RFID card that can provide a constant UID. Customers of the system put money into their accounts by giving money to system administrators. The balance of the customer is not attached to the registered cards separately; it is attached to a user account.

This system is an online service system where it uses RFID technology for access control. Access control over system's services and resources is something that needs to be present in most systems and needs to be convenient to use yet secure enough.

Three main components of access control are used in most information systems: identification, authentication, and authorization. [1] Identification is a process of indentifying unique user in the system and authentication is checking if user has

provided correct credentials if yes, authenticates user into system. Authorization is where user types will be defined as there is some information or resources that only system administrator has right to use and user don't. Or there are some operations that should only be performed by system administrator or by system user.

The three main approaches to user authentication are: knowledge-based, possession-based, and biometric-based. [1] All of these authentication types have their pros and cons and they are said stronger in ascending order.

But in real world authentication services, sometimes we don't see only one type of authentication used, instead, in authentication process you have to have something unique and you have to know something special. For example in all ATMs we use our bank card which has user identification number and we enter special pin code that only we know. Combining two types of authentications is said two-factor authentication.

However, in our system we develop application using one factor authentication but in the future it can be changed to two-factor authentication easily if needed. In our system we use possession-based authentication approach.

1.1 System description

The eKohvik is a self service point of sale system for the faculty members of Department of Software Science of TTU. It enables faculty members to buy available items in common rooms with the help of minicomputers (Raspberry Pis) located in every common room. Every Raspberry Pi is a station in the system and user can use any station to make purchases.

Furthermore, eKohvik application needs NFC reader connected to the Raspberry Pi to authenticate users with their cards. For example, user can use this application by using his/her public transport card, door card or even bank card as they all have RFID tag attached to them.

An administrator uses the Web Portal in order to administer the system and keep all information accurate in the system. The system administrator can, for example, update items and manage user information, create new station or delete station etc.

Moreover, the software needs Internet connection to interact with backend system and to fetch data from the database. All system information is saved in a database that is running on web server. The eKohvik application also has capability of presenting detailed information of the current user such as his/her balance and all previous purchases.

1.2 Organisation of thesis

- ***Background and Related work:*** Related POS solutions and related technologies and tools are described.
- ***Software Requirements Specification*****:** User Interfaces, Software Interfaces Hardware Interfaces, Communication Interfaces, Functional and Non-functional requirements are specified.
- ***Software Design Description*** Software perspective, user characteristics, class diagram, use case diagrams and sequence diagrams.
- ***Software Implementation*** Choice of technologies, description of software development
- ***Summary***

2 Background and related work

2.1 Related work

Points of sale systems have been developed for decades and we can see a lot of them in use nowadays in different places. However, integrating RFID technology to POS systems is little bit less common compared to traditional POS systems.

Horng-Lin Shieh, Yi-Chun Liao (2011) [2] developed an RFID restaurant POS system in order to reduce the labour cost, improve the service quality and achieve energy saving and carbon reduction for restaurants. The system can improve the service efficiency, lower the operating cost, and reduce environmental burden. System works in a way that each group of customers are provided with an RFID card in the entrance of the restaurant. Each table has a tablet or minicomputer that customer can interact and check the menu via the network. Customers can order meal by using given RFID card and order information is forwarded to the kitchen automatically. Every cook in the kitchen has an RFID card to represent his/her status and he/she swipes the card to change his/her current status. Administrator can see each cook's status and track the working attitude of the cook. Customer can give a comment about food they had at the counter and this information can be used for future improvement.

Gerald G. Abraham (2008) [3] patented Consumer-Centric RFID Point of Sale Transaction System and Method. The system uses Mobile point of sale (MPOS) transaction system based on wireless network, wireless device, RFID card and product with RFID tag attached to it. Moreover, wireless device includes RFID tag reader/writer. RFID card is used for authentication purposes and MPOS system is used to provide product information and purchase options for wireless appliance based on products RFID tag and to perform point of sale transactions. The product information and purchase options can be provided from a content media portal, or a product repository. The system can additionally enable consumer-initiated purchase without a cashier, and can provide pre-purchase product information, either by consumer request or by providing general information of interest, such as items for sale or on special.

U. B. Ceipidor, C. M. Medaglia, A. Marino (2012) [4] proposed solution for a potential vulnerability in mobile proximity payments. As we know mobile proximity payments

are evaluation of card payments and in these kinds of payments payer and payee should be located in the same place in order to perform transaction. The protocol proposed in this solution aims to provide mutual authentication between an NFC phone in card emulation mode and a POS system allowing the phone and POS share a session key to use to perform secure transactions. In smart card case only POS activates RFID tags using radio waves and it can be performed in 10 cm distance. It is thus also possible for third parties to activate the card in any place and information in the card can be shared without card owner's intention. The protocol proposed in this work suggests that both sides should authenticate each other in order to perform transaction.

Michael Dearing, Gediminas Vidugiris, William A. Linton, John Linton, Julia E. Krueger, Less (2010) [5] patented RFID point of sale and delivery method and system**.** In their system, products are delivered to places near to customer and each product has RFID tag attached to it. The places can be containers or refrigerators that can contain multiple amounts of products but each product is kept in separate cell which provides distance between products. Customer has device or card that has RFID tag to open the container and to open the cell to take the product. RFID technology is used for authentication from customer side. Furthermore, RFID is used as a tracking module for products and it can also provide information about need for more products or waste of the products.

2.2 Technological Background and Tools

2.2.1 Technological Background

Radio Frequency Identification (RFID)

During the last decades using radio frequency identification in different fields became mainstream and we can see that more and more industries are adopting and using RFID technologies in their fields. Nowadays, RFID is not only used in production or in retail, it is being used in healthcare, in security, in education or science and more. RFID enables identification from a distance and unlike earlier bar-code technology; it does so without requiring a line of sight [6]. Furthermore, compared to bar codes RFID tags support a larger set of unique IDs and can contain additional data such as manufacturer,

type of the product and also can perform measurements on environmental factors like temperature [7].

There are two types of RFID tags: passive and active [8]. Passive RFID tags are not supplied with battery and are activated by RFID readers. Active RFID tags are supported with battery and are active all the time if they have energy.

On the other hand, RFID technology is divided into low frequency system, high frequency system (frequency is 13.56 MHz), ultra high frequency system (frequency range about 900MHz) and system working in microwave frequency band of 2.4GHz or 5.8GHz [9].

The typical applications of high frequency RFID include electronic ticket, electronic ID card, and electronic locking anti-theft. Relevant international standards are ISO14443, ISO15693, and ISO18000-3 (13.56MHz)

RFID technology that uses high frequency is sometimes called Near Field Communication Protocol or NFC.

In our system we use NFC protocol and cards with ISO14443 international standard are supported.

Electron framework

Electron is a framework for developing desktop applications using web technologies HTML CSS and JavaScript. It is an open source project developed by Github [10].

Electron uses both Chromium Content Module and Node.js runtimes and facilitates developers to build graphical user interfaces using web pages. Moreover Electron gives access to native operating systems capabilities on Windows, OS X and Linux via OS-agonistic API [11] . Applications developed in Electron are cross platform that runs on Windows, OS X and Linux. We use electron to develop the eKohvik application for Raspberry Pi that runs on Rasbian-Jessie.

Angular.js

Angular.js is a client side technology for developing web applications and it is written entirely in JavaScript. It works with HTML, CSS, and JavaScript that makes web app development easier and faster [12] .

Angular.js supports advanced features that developers adopted to use in modern web applications development, such as [13] :

- Support for MVC architecture. It separates model views and controllers
- Ajax services
- Dependency injection
- Browser history (makes bookmarking and back/forward buttons work like normal web apps)
- Testing

Angular.js is used in the Web Portal development and in the eKohvik development together with Electron framework in our system.

Spring Boot

Spring boot is new generation of Spring framework. It is used for building enterprise applications in Java. Using Spring Boot, development of Spring application is much easy and it creates production ready Spring based applications that can just be run [14] . Spring Boot enables developers to create standalone applications that use embedded server, where we can just run them without any external server running.

Spring Boot takes an opinionated view of Spring platform and third party libraries that enable developers to start with minimum fuss. Mostly, in Spring Boot applications Spring configurations are required very little [15] .

Spring Boot has following features [15] :

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Provide opinionated 'starter' POMs to simplify your Maven configuration
- Automatically configure Spring whenever possible
- Provide production-ready features such as metrics, health checks and externalized configuration
- Absolutely no code generation and no requirement for XML configuration

We use Spring Boot for our Backend Application development.

Following frameworks and libraries are used in backend with Spring Boot:

- Spring-data-jpa: Spring Data JPA is a framework for working with entities. It is used to perform CRUD operations in Spring Boot
- Spring - security : Spring Security provides basic spring security in the project
- MySql-connector -java : MySql connector is used to connect to MySql database
- Jwt: Json Web Token is used for authentication. In our system we don't use sessions. When station or web portal connects to the backend, they are redirected to login page. If admin provides correct credentials Json web token is generated in the backend and is sent to the applications. The token will be saved in the applications and every time when request is send from application to backend web token is sent in the header of the request.
- Spring-boot-maven-plugin: Maven plugin is used in order to use maven during the development. Maven is dependency manager application that installs all the dependencies of the project automatically. We can use gradle or ant also.

Pscslite

Pscslite is middleware to access a smart card using SCard API (PC/SC) [16]. It is an open-source project for working with smart cards. It is supported by Windows, OS X, and Linux operating systems.

Nfc-pcsc

Nfc-pcsc is a simple wrapper around pcsclite to facilitate working with NFC tags [17].

2.2.2 Hardware platform

Raspberry Pi (version 2B or 3)

Raspberry pi is a card sized minicomputer originally designed for education purposes. It is popular among digital hobbyists and makers of Internet of Things (IoT) devices. Basic model of Raspberry Pi costs around $35. It has a 64-bit quad-core ARMv8 processor and uses a Raspbian distribution of Linux for its default operating system (OS) [18]. Nowadays, it is also possible to install Windows 10 IoT operating system.

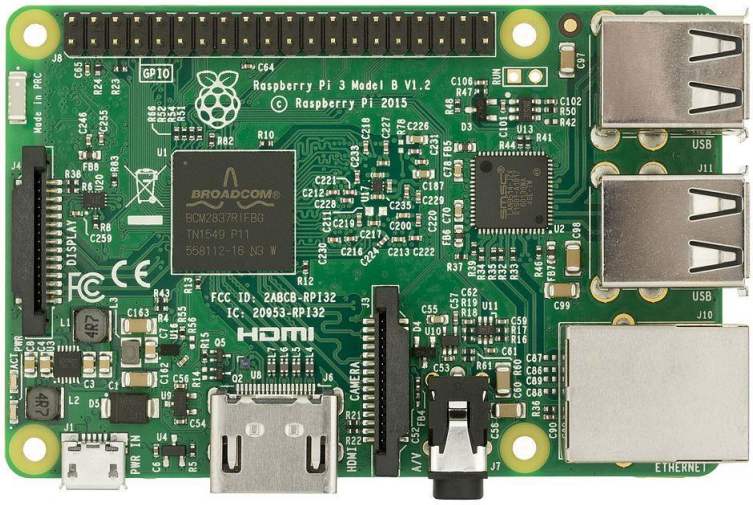

Figure 1 - Raspberry Pi

The 7” Touchscreen Monitor for Raspberry Pi

The 7” Touchscreen Monitor for Raspberry Pi facilitates users to create all-in-one integrated projects such as tablets, infotainment systems and embedded projects. Size of the screen is 800x480 pixels [19].

Figure 2- The 7″ Touchscreen Monitor for Raspberry Pi

ACR1251U USB NFC Reader II

The ACR1251U USB NFC Reader II supports NFC tags and devices. It has Secure Access Module slot and firmware upgradeable. This device can be used for personal identity verification, network login, and online banking [20].

Figure 3 - ACR1251U USB NFC Reader II

3 Software Requirements Specification

This chapter provides outsight to understand what the requirements for the eKohvik system are. It covers user interface requirements, hardware interface requirements, communication interface requirements and functional and non-functional requirements of the system.

3.1 External interface requirements

This section provides detailed description of all inputs into and outputs from the system. It also gives description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1.1 User interfaces

System will be developed as three independent applications. They are the eKohvik Application for Raspberry Pi, the Backend Application and the Web Portal.
Backend application doesn't have user interface so we don't need to write any user interface requirements for it. However, for the eKohvik Application and for the Web Portal we need to specify user interface requirements separately.

3.1.1.1 User interface requirements for the eKohvik Application.

The eKohvik Application runs on Raspberry pi and every the eKohvik Application on Raspberry Pi serves as a station to buy items. When administrator wants to add new station first he/she must create new station object in the database using the Web Portal.

When we run the eKohvik Application for the first time it has to show login page for the administrator to log in using the same username and password used when he/she created the station object. After administrator logged into the eKohvik Application this station is registered in the system and starts operating and enables users to buy items or create account exc.

A first-time user of the eKohvik Application should see a page that asks from him/her to swipe card to NFC reader. If the card is not registered in the system user should be able to create new user account in the system with this card or should be able to add this card

to his/her existing user account. To accomplish this tasks application should show registration options page.

If user is not first- time user and card is already attached to his/her existing user account, items page will be shown to the user. On this page, user's name and current balance of the user is shown on top with one additional button that says check all purchases. If user clicks on check all purchases button application shows list of previous purchases user has made in a page as a list.

On the Items Page, user can choose items from items list that is shown on the left side of the screen. This process is quite easy that user taps on the item he/she wants to buy and this item is added to order list that is shown on the right of the screen. User can buy more than one item or can buy one item in multiple amounts. If by mistake user taps on the item that he/she doesn't need he/she can delete that item from orders list by tapping on it.
There is always Cancel button on every page that if user wants to cancel the process. If user clicks on Cancel button all the process is cancelled and application shows card waiting page.
If user clicks Buy button purchase is send to the backend system and if it was successful Coffee Machine application shows Success page otherwise error page.

3.1.1.2 User interface requirements for the Web Portal

As the Web Portal is single page web application it also has a lot of user interaction pages. First of all when administrator enters url of the Web Portal on his/her browser, login page is shown and as soon as administrator provides correct credentials and submits, application shows full web page for managing the system.
The Web Portal has main menu tab for every entity like users, items, cards, purchases and stations. Every menu tab is attached to entity page. For example if administrator clicks Users tab, Users page is shown.
Users page contains form for creating new user account or editing existing user account information. On the bottom of the form all user account are shown as a list and each of this user account has Edit and Delete buttons. Edit button is to edit information of the account and Delete button is used for deleting the account.
Items page also contains form for creating new item or editing existing item information. Items are shown as a list on the bottom of the form and there is Edit button on each item

that is used to edit the item information. However, items doesn't have Delete button because items can't be deleted. If there is no item available administrator puts the item availability to Not available and item won't be shown on the items list of the eKohvik Application.
Stations page also has form for creating new station. It also lists all the stations at the bottom of the form and each station has Delete button.
Cards page doesn't have any form to create new card or update card information. It shows only list of cards with user email addresses and there is Delete button on each card object.
Purchases page doesn't have any form for creating new purchase or updating purchase information. Purchases are listed and each purchase has Delete button.
Home page contains information about the usage of the Web Portal.

3.1.2 Hardware interfaces

As system divided into three parts every part uses different hardware. They are:

1. The eKohvik Application:

    1.1 NFC reader (USB NFC reader)

    1.2 Touch screen for Raspberry Pi (Raspberry Pi Touch Screen)

    1.3 Raspberry Pi (version 2 B or 3)

2. Backend Application:

    2.1 Web server

    2.2 MySql sever

The NFC reader is connected to Raspberry Pi using USB port. The Touch Screen is installed on top of Raspberry Pi. Connection between Touch screen and Raspberry Pi is maintained by two connections: power from the Pi's GPIO port and a ribbon cable that connects to the DSI port present on all Raspberry Pis. The Web server and the database server are provided by university.

3.1.3 Software interfaces

Our system doesn't use any external software systems. However, Rasbian-Jessie 2017-01-11 is installed to Raspberry Pi and Coffee Machine Application runs on it.
Web server is running on Linux OS and there is a MySql database server.

3.1.4 Communications interfaces

Communication between the eKohvik Application and the Backend System is maintained using internet connection. Data exchange is in JSON format.

3.2 Functional requirements

This section covers the requirements that specify all the fundamental actions of the software system.

3.2.1 User Class 1 – User

ID: FR1

Feature: Create an account

In order to create an account

A user

Should have an NFC card that is supported by the system

Scenario: Required information for registration

Given the user wants to create an account

And the user doesn't have an account

When the user swipes the NFC card on the NFC reader

And clicks the Create new user button

And provides user name

And email address

And card name

Then the user should be able to create an account

Scenario: Confirmed registration

Given the user has provided all the information and clicked the create user button

Then the system should send email to provided email address

Then the system should show page that informs welcoming email is sent to provided email address.

Then new user account should be created

Scenario: Invalid input field

Given the user didn't fill all the input fields

Or provided an invalid email address

Or didn't fill one input field

And clicked the create user button

Then the system should show warning message to the user

And shouldn't continue the process.

ID: FR2

Feature: Add a card to an account

In order to add card to an account

A user

Should have an NFC card that is supported by this system

Scenario: Required information for adding card

Given the user wants to add card to an account

When the user swipes the NFC card on the NFC reader

And clicks Add card to account button

And provides email address

And card name

Then the user should be able to add card to an account

Scenario: Confirmed card addition

Given the user has provided all the information and clicked the add card button

Then system found account with provided email address

Then the system should send email to provided email address

Then the system should show page that informs that welcoming email is sent to provided email address.

Then card should be added to the account

Scenario: Invalid input field

Given the user didn't fill all the input fields

Or provided an invalid email address

Or didn't fill one input field

And clicked the add card button

Then the system should show warning message to the user and shouldn't continue the process.

Scenario: Account doesn't exist with this email address

Given the user has provided all the information and clicked the add card button

Then system didn't find an account with provided email address

Then the system should show page that informs that account doesn't exist

Then card shouldn't be added to any account

ID: FR3

Feature: User log – in

In order to use the system

A user

Should be logged into the eKohvik Application

Scenario: Successful log-in

Given the user wants to log in

When the user swipes a registered NFC card to NFC reader

Then the user should be logged in as a user

Scenario: Unsuccessful log – in

Given the user wants to log in

When the user swipes a not registered NFC card to NFC reader

Then system should show the options that user can create new account, add this card to an account or cancel

ID: FR4

Feature: Buy items

In order to buy items

A user

Should be logged in to the system

Scenario: Make an order list

Given the user wants to make an order

And user is logged in

When user clicks on items on presented item list

Then item should be added to the order list

Scenario: Delete item from order list

Given the user is logged in

And added items to the order list

And added item that he/she doesn't need

When the user taps on the item in order list

Then item should be deleted from the order list

Scenario: Buy items

Given the user is logged in

And made an order list

When user clicks on the Buy button

Then new purchase of the user should be created

Then cost of the purchase should be subtracted from the user's balance

Then success message and users updated balance should be shown on the screen

Scenario: Connection error

Given the user is logged in

And created an order list

And clicked on Buy button

When connection error happened

Then error details are shown on the screen

And all transaction are cancelled

ID: FR5

Feature: Check previous purchases

In order to check previous purchases

A user

Should be logged in to the system

Scenario: Activities to check previous purchases

Given the user wants to check his/her previous purchases

And user is logged in

When the user clicks on check old purchases button

Then all purchases of the user should be shown on the screen as a list

ID FR6

Feature: Cancel process

In order to cancel process

A user

Should be logged in to the system

Scenario: Cancelling process

Given the user wants to cancel the process

And user is logged in

When the user clicks on check Cancel button

Then all processes should be cancelled

Then application state should change to log in page

3.2.2 User Class 2 – Administrator

ID: FR7

Feature: Administrator log in

In order to administer the system

An administrator

Should be logged in

Scenario: Successful log-in

Given the administrator wants to log in

When enters valid username and password

And clicks the login button

Then the administrator should be logged in as administrator of the system

Scenario: Unsuccessful log in attempt

Given the administrator wants to log in

When enters not valid username and password

And clicks the login button

Then the administrator should be informed that he entered wrong credentials

ID: FR8

Feature: Create a new user account without card

In order to create a new account to a user

An administrator

Should be logged in to web portal

Scenario: Create an account to a user without attaching card

Given the administrator wants to create a new account to a user without card

And is logged in to web portal

When the administrator provides user name

And a valid email address

And balance

Then add button is activated

When the administrator clicks the add button

Then new user account is created but it doesn't have card attached to it

ID: FR9

Feature: Update user information

In order to update user information

An administrator

Should be logged in to the Web Portal

Scenario: Update user information

Given the administrator wants to update user name of the user

And logged in to the web portal

When administrator chooses the user and presses the Edit button on users list

Then user information should be presented in users registration form

Then the administrator should be able to update user information for each field

ID: FR10

Feature: Deleting user account

In order to delete user account

An administrator

Should be logged in to the web portal

Scenario: Delete user account

Given the administrator wants to delete a user account

And is logged in to the Web Portal

And knows the account details that he wants to delete

When administrator finds the user account on the users list

And clicks the Remove button

Then purchases that are attached with the account are deleted.

Then cards that are attached to the account deleted

Then account is deleted

ID: FR11

Feature: Creating a new item in the system

In order to create new item in the system

An administrator

Should be logged into the Web Portal

Scenario: Create new item

Given the administrator wants to create new item

And is logged in to the Web Portal

When the administrator provides item name

And item price

And item availability in the store

Then administrator should be able to click the Add button

When Add button is clicked

Then new item is added to the system.

ID: FR12

Feature: Update item information

In order to update item information

An administrator

Should be logged in to the Web Portal

Scenario: Update item information

Given the administrator wants to update item information

And is logged in to the Web Portal

When administrator clicks Edit button on the item in items list

And updates the input field with new information

Then Update button should be clickable

When the administrator clicks the Update button

Item information should be updated

ID: FR13

Feature: Delete card from user's account

In order to delete item from user's account

An administrator

Should be logged in to the Web Portal

Scenario: find card to be deleted by user's email address

Given the administrator wants to find card with users email address

And is logged in to the Web Portal

And provided user's email address to the search box

Then cards should be filtered by user's email addresses

Scenario: Delete card from user's account

Given the administrator wants to delete card from user's account

And is logged in to the Web Portal

And clicked on cards tab on the menu

And found the card to be deleted

Then the administrator should be able to click Remove button of the card item

When the administrator clicked on Delete button

Then card should be deleted from attached user's cards list

Then card should be deleted from the system

ID: FR14

Feature: Deleting purchase that is made by mistake

In order to delete purchase that is made by mistake

An administrator

Should be logged in to the Web Portal

Scenario: Delete purchase that is made by mistake

Given the administrator wants to delete purchase that is made by mistake

And is logged in to the Web Portal

And clicked to the Purchases tab of the menu

And has found the purchase to be deleted on purchase details list

When the administrator clicks the delete button

Then the deleted purchase's cost amount is added back to the user's balance who created this purchase

Then purchase is deleted forever from the system and database

ID: FR15

Feature: Adding new station

In order to add new station to the system

An administrator

Should be logged in to the Web Portal

Scenario: Add new station to the system

Given the administrator wants to add new station to the system

And logged in to the Web Portal

And clicked to the Stations tab of the menu

When the administrator provides station address

And station username

And station password

Then administrator should be able to click the Add button

When Add button is clicked

Then new station is added to the system.

ID: FR16

Feature: Delete station from the system

In order to delete station from the system

An administrator

Should be logged in to the Web Portal

Scenario: Delete station

Given the administrator wants to delete station from the system

And logged in to the Web Portal

And clicked to the Stations tab of the menu

And has found the station to be deleted on stations list

When the administrator clicks the delete button

Then the station is deleted from the system

3.3 Performance requirements

This section covers special types of requirements where these requirements are related to user interaction with the software and system performance measurements

ID: QR1

TITLE: Easily buy items feature

DESC: Buying items from the store should be easy and understandable for user

RAT: In order for user to buy items easily

DEP: None

ID: QR2

TITLE: Cancel process

DESC: Cancelling any process and closing any page should be easy and understandable for user

RAT: In order for user to be able to cancel process or close the page

DEP: None

ID: QR3

TITLE: Fault Tolerance of the system

RAT: If the system loses the connection to the Internet or to the NFC Reader or the

system gets some strange input, the user should be informed.

RAT: In order for user to be informed of faults

DEP: None

3.4 Design constraints

This section includes the design constraints on the software caused by the hardware or other systems.

ID: QR4

TITLE: On Screen Keyboard

DESC: The Coffee Machine Application needs On Screen Keyboard. And it should be activated only when user taps on specific input field

RAT: In order for user to user keyboard

DEP: None

3.5 Software system attributes

The requirements in this section specify the availability, security and maintainability of the software system.

Availability

ID: QR5

TITLE: Internet Connection

DESC: Coffee Machine Application should be connected to Internet

RAT: In order for Coffee Machine Application to communicate with Backend system

DEP: None

ID: QR6

TITLE: Reading information from card

DESC: Coffee Machine Application should be connected to NFC reader

RAT: In order for Coffee Machine Application to get card token and authenticate user

DEP: None

Security

ID: QR7

TITLE: User log in security

DESC: The card token should be hashed and encrypted during the communication between Coffee Machine Application and Back-end system

RAT: In order others not to be able to get card token

DEP: None

ID: QR8

TITLE: Admin login to account security

DESC: The messages should be encrypted for log-in communications between Web Portal Frontend application and Backend system

RAT: In order others not to be able to get username and password from those messages

DEP: None

Maintainability

ID: QR9

TITLE: Application extendibility

DESC: The application should be easy to extend. The code should be written in a way that it facilitates implementation of new functions easily.

RAT: In order for future functions to be implemented easily to the application.

DEP: None

4 Software Design Description

The Software Design Description (SDD) is a crucial document for software development lifecycle. This document describes the software design and architecture of the Coffee Machine System.

4.1 Software perspective

This system consists of three parts: one application that runs on Raspberry Pi (CMA), one Backend system that provides Rest API Services and one single page web application (the Web Portal).

The eKohvik will be used to buy items from the store while the Web Portal will be used to manage the system. However, the Backend Application doesn't have user interface and neither user nor administrator interacts with it directly.

Raspberry Pi that the eKohvik Application is running on, will need to be connected with NFC reader, so that NFC reader reads NFC tag from the user's card and sends it to the application. CMA uses card token as identification of the user and if card exists in the database application authenticates the user to make purchases or check user balance. The eKohvik consumes REST API Services provided by backend system and interactions between these systems are done through Internet connection.

The Web Portal is used to control system resources and will be used only by administrator. It enables administrator to perform some actions like creating new items, users, stations and editing entities in the database. The Web Portal also uses REST API Services and will be connected to the Backend Application via Internet. The Web Portal is a single page web application that runs on web browsers.

As this system is data-centric, it uses database to store all system's data. The database is located in database server. Either the eKohvik Application or the Web Portal can't interact or fetch data from database directly. The database will be connected only to the backend system and all the CRUD operations will happen with the help of the Backend System.

The Backend Application is RESTful Web Service Application that has connection with the database and whole business logic is done in it. The Backend Application doesn't have graphical user interface (GUI) but, user interacts with it using the eKohvik

Application and administrator interacts using the Web Portal. The Backend Application will be running on web server.

All applications interactions and running environments are shown on Figure 1.

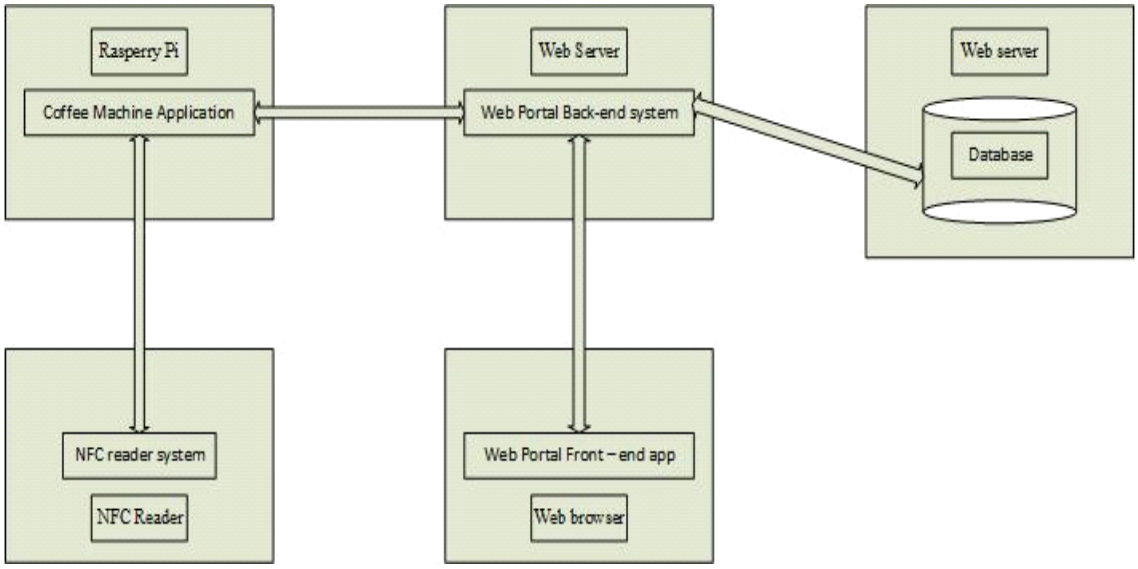

Figure 4 - Block Diagram of the eKohvik System

4.2 Software functions

By the help of the eKohvik Application users will be able to create a new user account, add card to existing user account and buy available items in the store. Furthermore, users can check all the previous purchases he/she has made and current balance in his/her account. After users are authorized items will be shown as a list and users will be able to make purchase.

Web Portal will provide functionality to control the system and manage all resources. It is the main application that enables administrator to take over the control on whole system. It provides functionalities for creating, editing, deleting entities in the database.

4.3 User characteristics

There are two types of users who interact with the system: user and administrator. Each of these users has their own requirements. The eKohvik Application users can buy items from the store, add card to existing user account and create new user account. Moreover, user can see all the previous purchases that he/she made. The administrator only

interacts with the Web Portal. Administrator manages the overall system so that system doesn't contain any incorrect information.

4.4 Constraints

Internet connection is constraint for the system. Since all three parts of the system uses Internet to interact, it is crucial that there is an Internet connection for the system to operate.

4.5 Assumptions and dependencies

One assumption about the eKohvik Application is that, it will be used on a Raspberry Pi with touch screen. Furthermore, as this application cannot force a faculty member to take item only by using the eKohvik Application, usage environment will be supposed to be dependant of users' honesty. Another point is the eKohvik Application shouldn't make user disappointed by not allowing buying item if user doesn't have enough money in his/her account's balance to buy an item. So balance in user account can become negative.

4.6 Class diagram

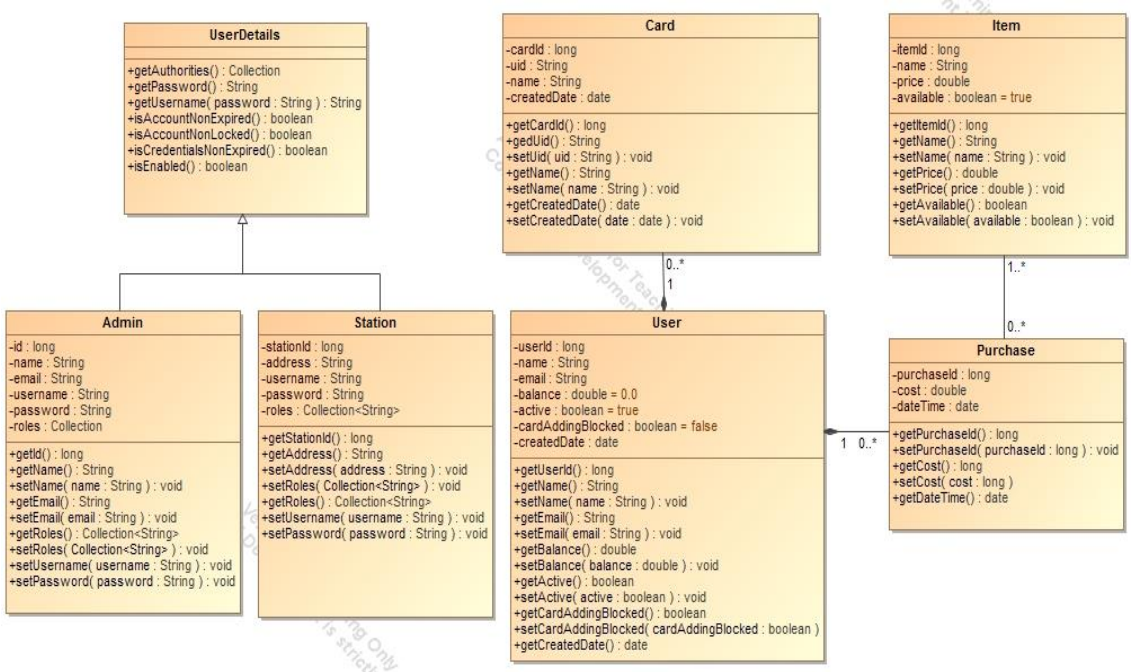

Figure 5 – Class Diagram of the eKohvik Backend System

4.7 Use Cases

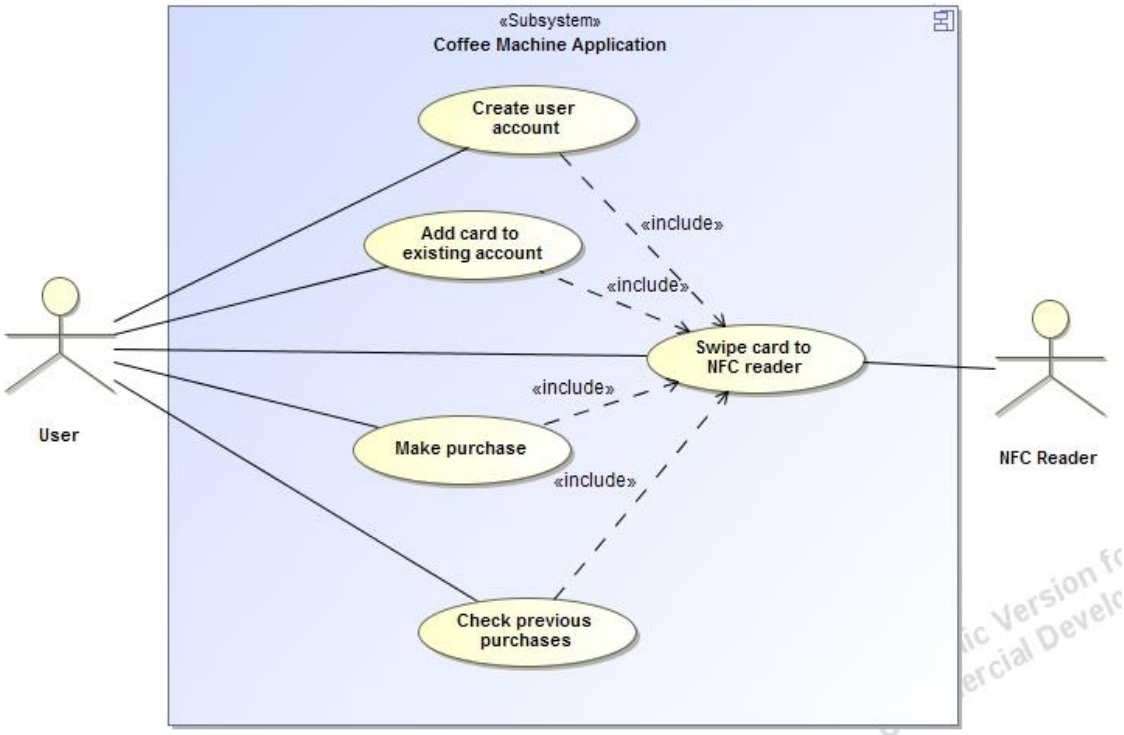

Figure 6 – The eKohvik Application use case diagram

Table 1- The eKohvik Application use case diagram description

Use Case	Description
Swipe card to NFC reader	A user needs to swipe a card that has NFC tag to NFC reader before performing any transaction Actors: User, NFC reader Pre-condition: A user with card that has NFC tag Main flow of events: 1. The user swipes his/her card to NFC reader. 2. The NFC Reader reads the card uid and sends it to the application. 3. The Application hashes and sends the hashed uid to the backend system. 4. The backend system checks the existence of the card in card table of the database. 5. If the card exists in the database, system sends user information with list of items available in the store. 6. The application displays the items page with the user name, current user balance and check previous purchases button on the top of the screen, and items list on the bottom left. 7. Post-condition: The user has been authorized to perform transactions. Alternative flow: 1. The user swipes his/her card to NFC reader.

	2. The system checks the existence of the card in card table in the system. 3. If the card doesn't exist in the database the system shows the page that shows the options to create new user account, add card to existing user account and cancel. 4. Post-condition: The user has not been authorized to perform transactions. Alternative flow: 1. The user swipes his/her card to NFC reader. 2. The system can't read uid of the card. 3. The system shows error page that says card is not supported.
Create user account	A new user needs create user account in the system before performing any transaction Actor: User Pre-condition: A user who doesn't have an account and has card supported by the system Main flow of events: 1. The user swipes his/her card to NFC reader. 2. Coffee Machine Application shows page with three buttons: Create new account, Add card to existing user account, Cancel. 3. The user clicks the Create new account button. 4. The app opens new windows with input fields. 5. The user fills all the input fields and clicks Create account button. 6. The system checks that all of the required information was entered. If yes: The system creates new Card object in card table of the Database. The system creates new User in the user table in the Database. The system sends email to the email address provided in input field. System displays page that says email is sent to the email address. If no: The system displays warning message on current page.
Add card to existing user account	Card should be registered in the system before performing any transaction Actor: User Pre-condition: A user with existing user account in the system and current card is not registered but supported by the system Main flow of events: 1. The user swipes the card to NFC reader. 2. Coffee Machine Application shows page with three buttons: Create new account, Add card to existing user account, Cancel.

	3. The user clicks the Add card to existing user account button. 4. The app opens new windows with input fields. 5. The user fills all the input fields and clicks Add card button. 6. The system checks that all of the required information was entered. If yes: The system creates new card object in card table of the Database. The system sends email to provided email address. The system attaches the card to user account and updates the account table with new information. If no: The system displays warning message on current page.
Make purchase	A user can buy items from the store Actor: User Pre-condition: user has logged in Main flow of events: 1. The user taps the item in the item list. 2. The system adds the item to the order's item list. 3. User may choose another item or take more quantities of the same item by just tapping on the item. 4. If item is added by mistake user taps on the item in the orders item list. 5. The system deletes the item from order's item list. 6. User clicks Buy button. 7. The system creates new purchase in purchase table in the database. 8. The system subtracts the purchases cost from the user's balance. 9. The application shows the window that shows the users current balance. Post – condition: The session is over and user is logged out from the system.
Check previous purchases	A user can check previous purchases he/she made Actor: User Pre-condition: User is logged in Main flow of events: 1. The user clicks the check previous purchases button on top. 2. The application fetches all the purchases user has made in a list. 3. User checks the list. 4. User clicks the close button. Post-condition: The session is over and user is logged out.

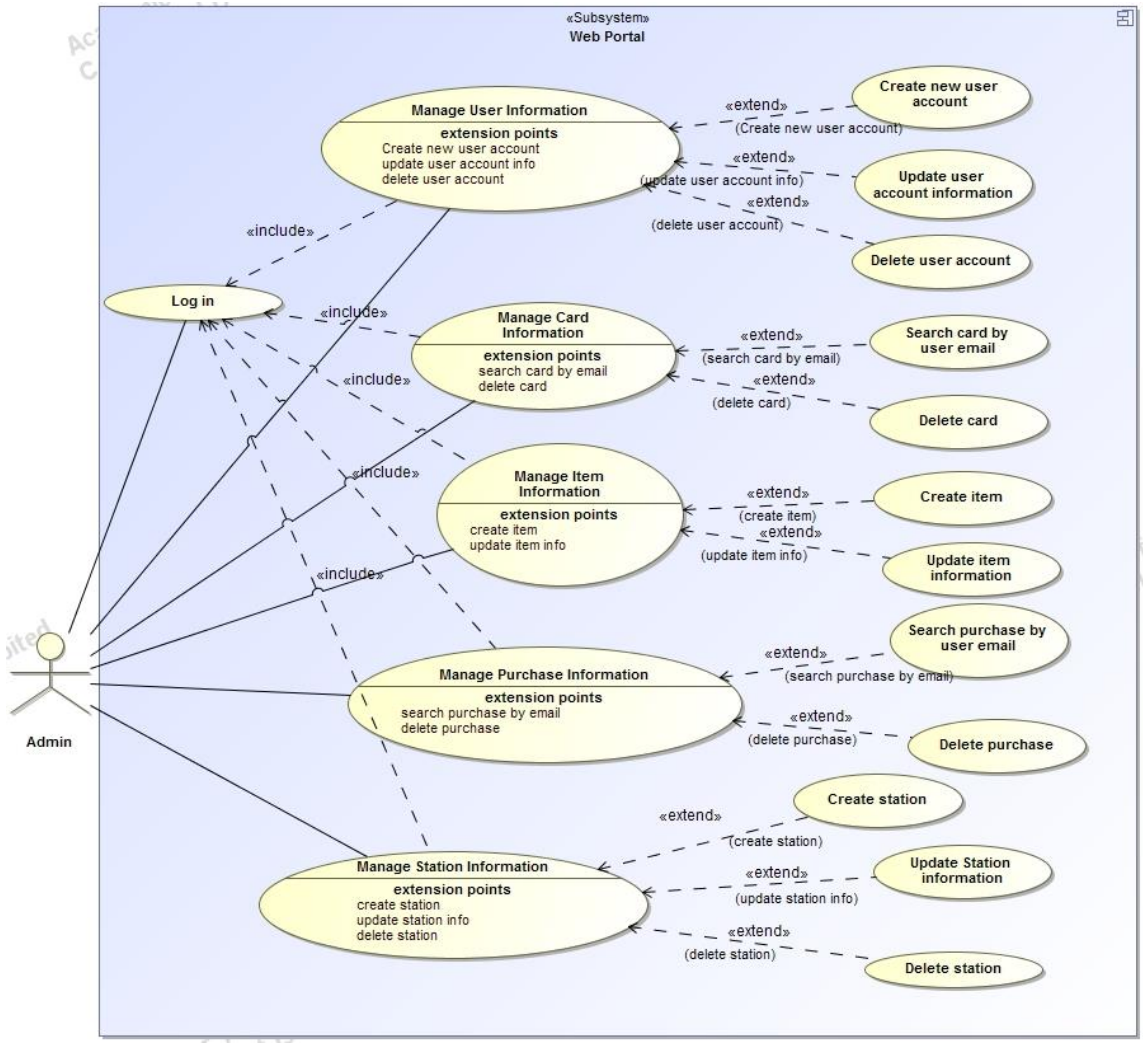

Figure 7 – The Web Portal use case diagram

Table 2 - The Web Portal use case diagram description

Use Case	Description
Log -in	An administrator needs to log in before performing any transaction Actor: Admin Pre-condition: A registered administrator Main flow of events: 1. The system shows Log-in Page. 2. The administrator enters his/her username and password. 3. The administrator clicks the Login button. 4. The system validates the log-in information against the Admin table in the database. 5. The administrator is an authorized admin; the system displays the home page of the web portal. Post-condition: The administrator has been authorized to perform transactions.

	Alternative flow: 1. The system shows the Log-in Page. 2. The administrator enters his/her username and password. 3. The administrator clicks the Login button. 4. The system validates the log-in credentials against the Admin table in the database. 5. Administrator is not authorized admin; the system displays the pop-up message to inform the Admin. Post-condition: The administrator has not been authorized to perform transactions.
Create new user account	An administrator manages user information Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator clicks User tab on the menu. 2. The administrator enters all the required information on the form. 3. The Administrator clicks the Add button. 4. The system creates new user in user table in the database. Post-condition: New user account is created.
Update user account information	An administrator manages user information Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator clicks User tab on the menu. 2. The administrator finds the user account in the users table. 3. The administrator clicks the Edit button. 4. The administrator updates the user account information. 5. The administrator clicks the Update button. 6. The system updates the user row in user table in the database. Post-condition: user account information is updated.
Delete user account	An administrator manages user information Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator clicks Users tab on the menu. 2. The administrator finds the user account in the users table. 3. The administrator clicks the Delete button. 4. The system deletes the cards that are attached to this user account. 5. The system deletes the purchases that this user made. 6. The system deletes the user from the user table in the database. Post-condition: User account is deleted.
Search card by user email	An administrator finds the card by user's email address Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator clicks the Cards tab on the menu. 2. The administrator enters the user's email address on the search box. 3. The web portal filters the cards by user email.

	Post-condition: Card is found by user's email address.
Delete card	An administrator deletes the registered card from the system Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator clicks the Cards tab on the menu. 2. The administrator finds the card in cards list. 3. The administrator clicks the Delete button. 4. The system detaches the card object from the card user's account. 5. The system deletes the card from the card table in the database. Post-condition: Card is deleted from the system.
Create new item	An administrator creates a new item Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator clicks the Items tab on the menu. 2. The administrator enters all required information to the input fields. 3. The administrator clicks the Add button. 4. The system creates new item in the item table of the database. Post-condition: New item is created.
Update item information	An administrator updates item information Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator finds the item that should be updated from the items list. 2. The administrator clicks the Edit button on the item. 3. The administrator updates all required information to the input fields. 4. The administrator clicks the Update button. 5. The system updates the item information in item table of the database. Post-condition: Item information is updated.
Search purchase by user's email	An administrator finds purchase by user's email address Actor : Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator enters the user's email address on the search box. 2. The system filters the purchases by user's email. Post-condition: Purchases are filtered by user's email address.
Delete purchase	An administrator deletes a purchase Actor : Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator finds the purchase that should be deleted. 2. The administrator clicks the Delete button on the purchase. 3. The system adds the cost amount of the purchase to the user's

	balance. 4. The system deletes the purchase from purchase table in the database. Post-condition: Purchase is deleted from the system and cost is returned to the user's balance.
Create station	An administrator creates a new station Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator enters all the information to the input fields of the form. 2. The administrator clicks the Add button. 3. The system created new station in the station table of the database. Post-condition: New station is created
Update station information	An administrator updates station information Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator finds the station that needs to be updated in the station list. 2. The administrator updates all the information to the input fields of the form. 3. The administrator clicks the Update button. 4. The system updates station's information in the station table of the database. Post-condition: Station information is updated.
Delete station	An administrator deletes station Actor: Admin Pre-condition: Administrator is logged in Main flow of events: 1. The administrator finds the station that needs to be deleted in the station list. 2. The administrator clicks the Delete button. 3. The system deletes the station in station table of the database. Post-condition: Station information is deleted.

4.8 Sequence Diagrams

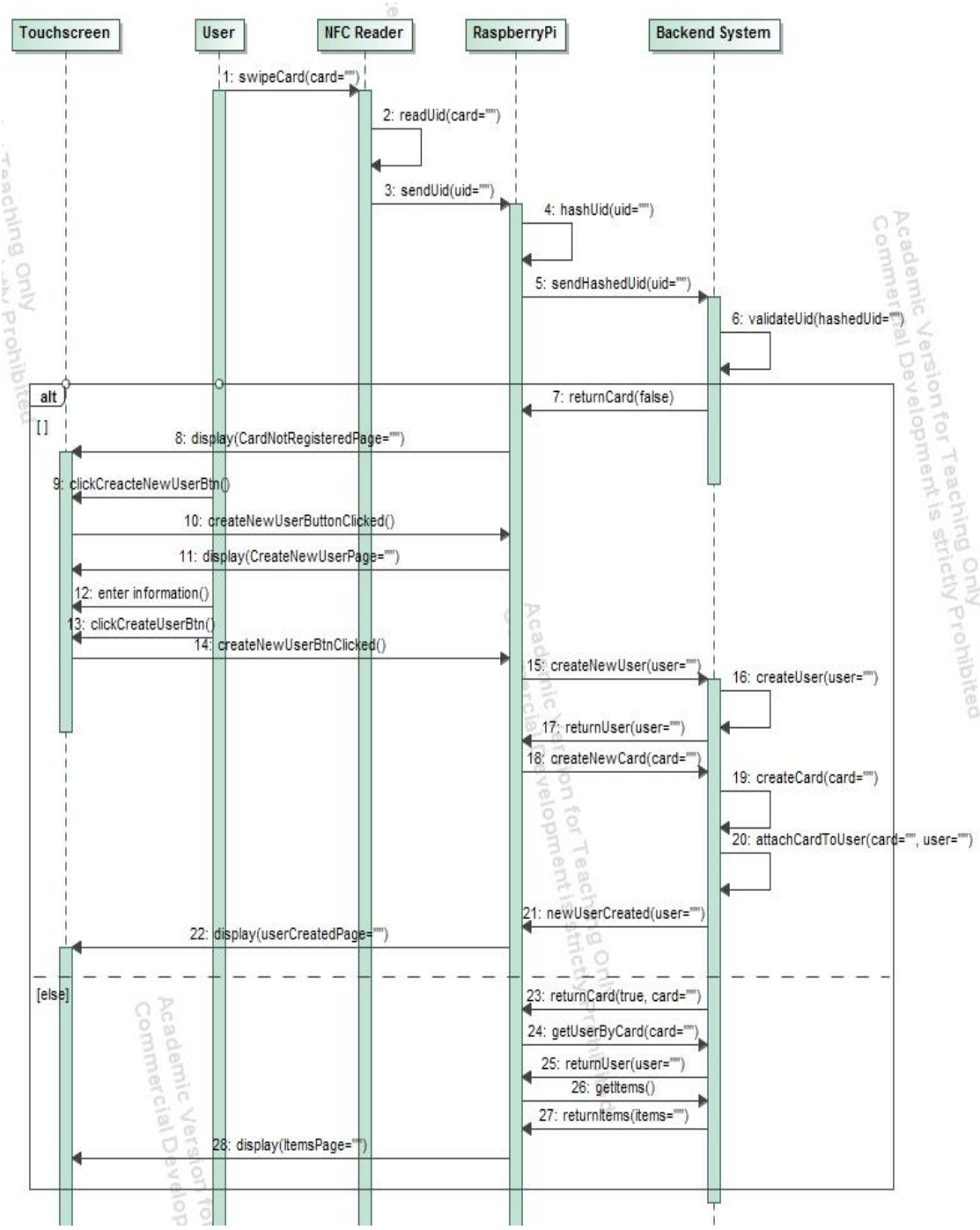

Figure 8 - Create new user account sequence diagram

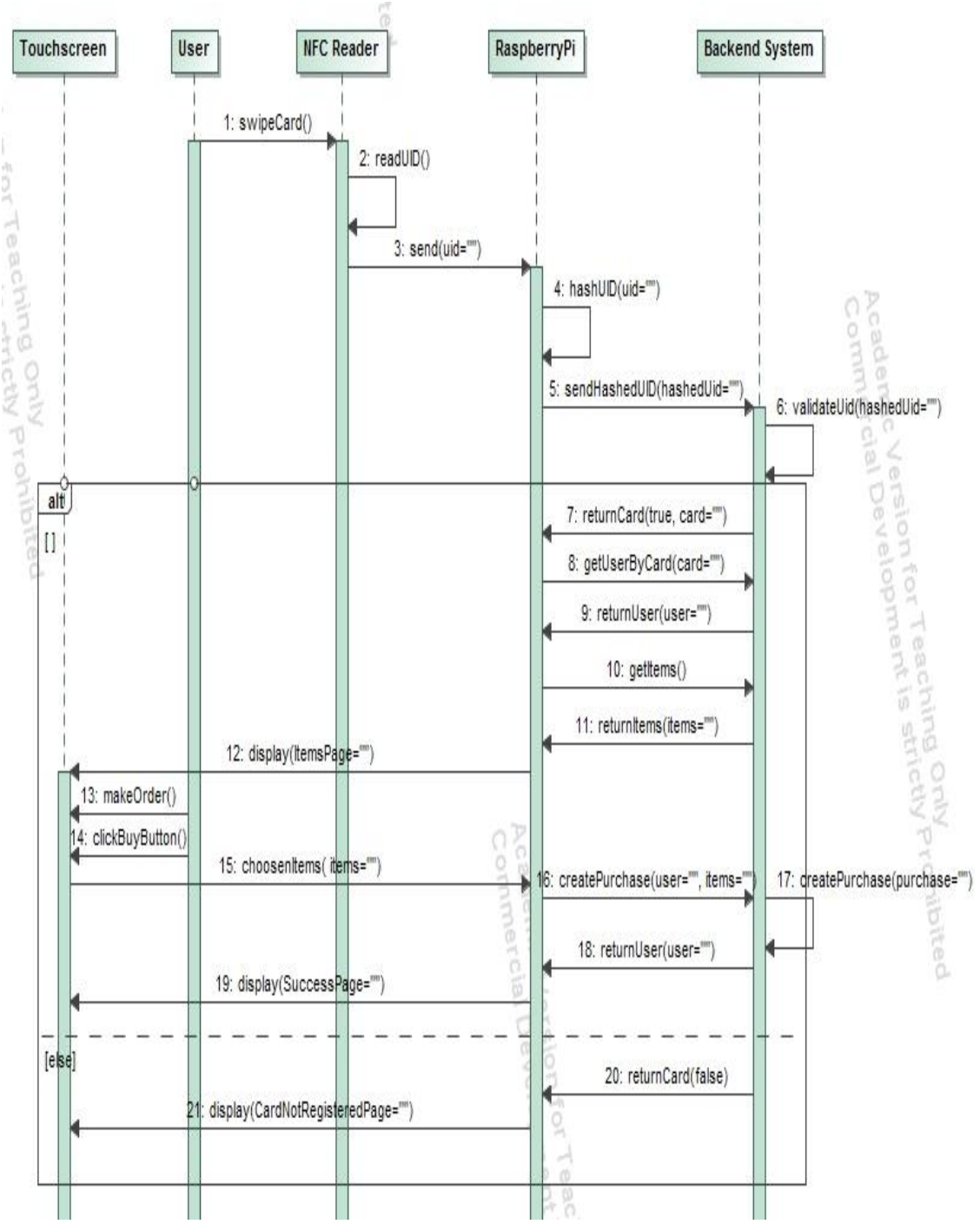

Figure 9- Create purchase sequence diagram

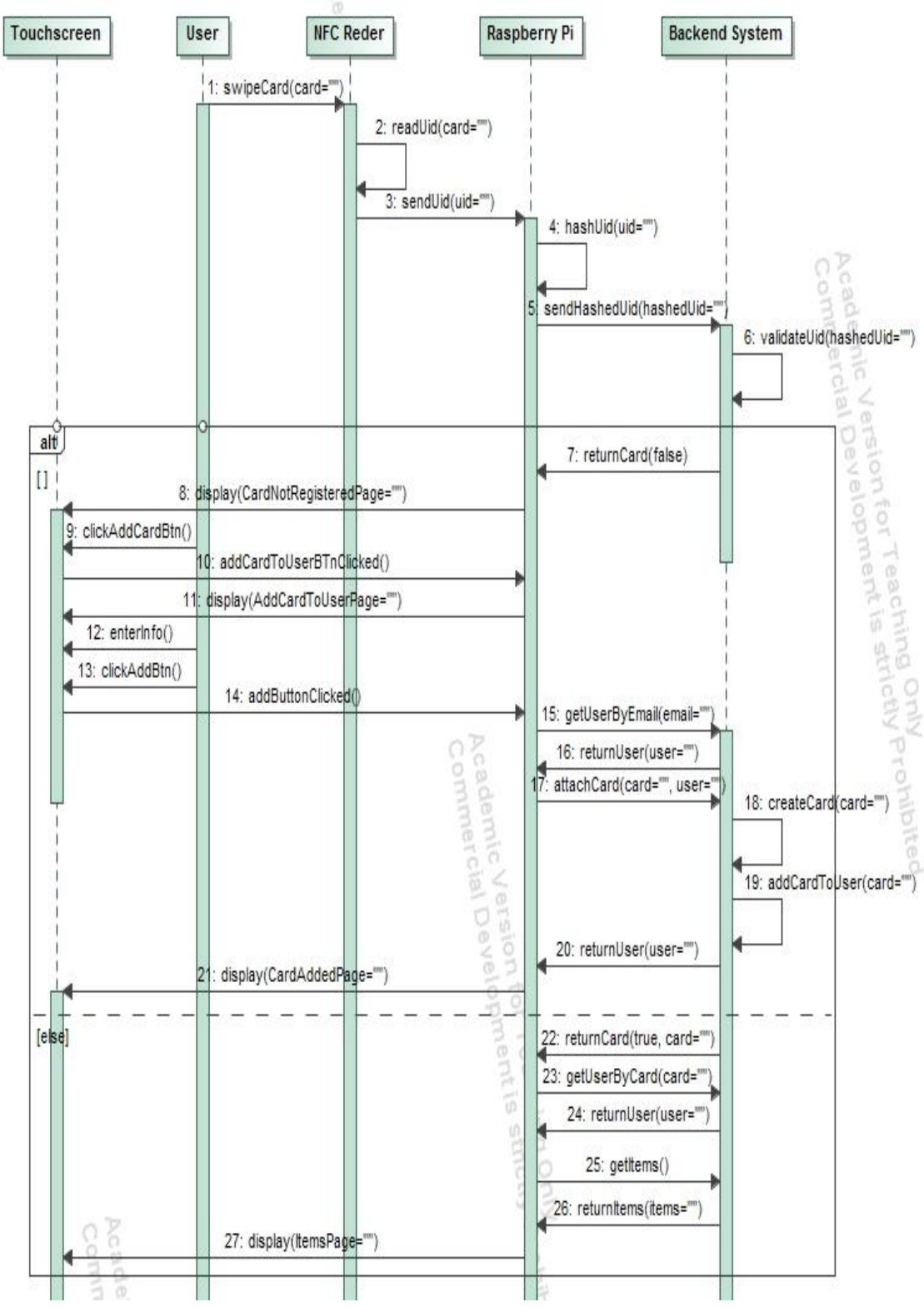

Figure 10 - Add card to user account sequence diagram

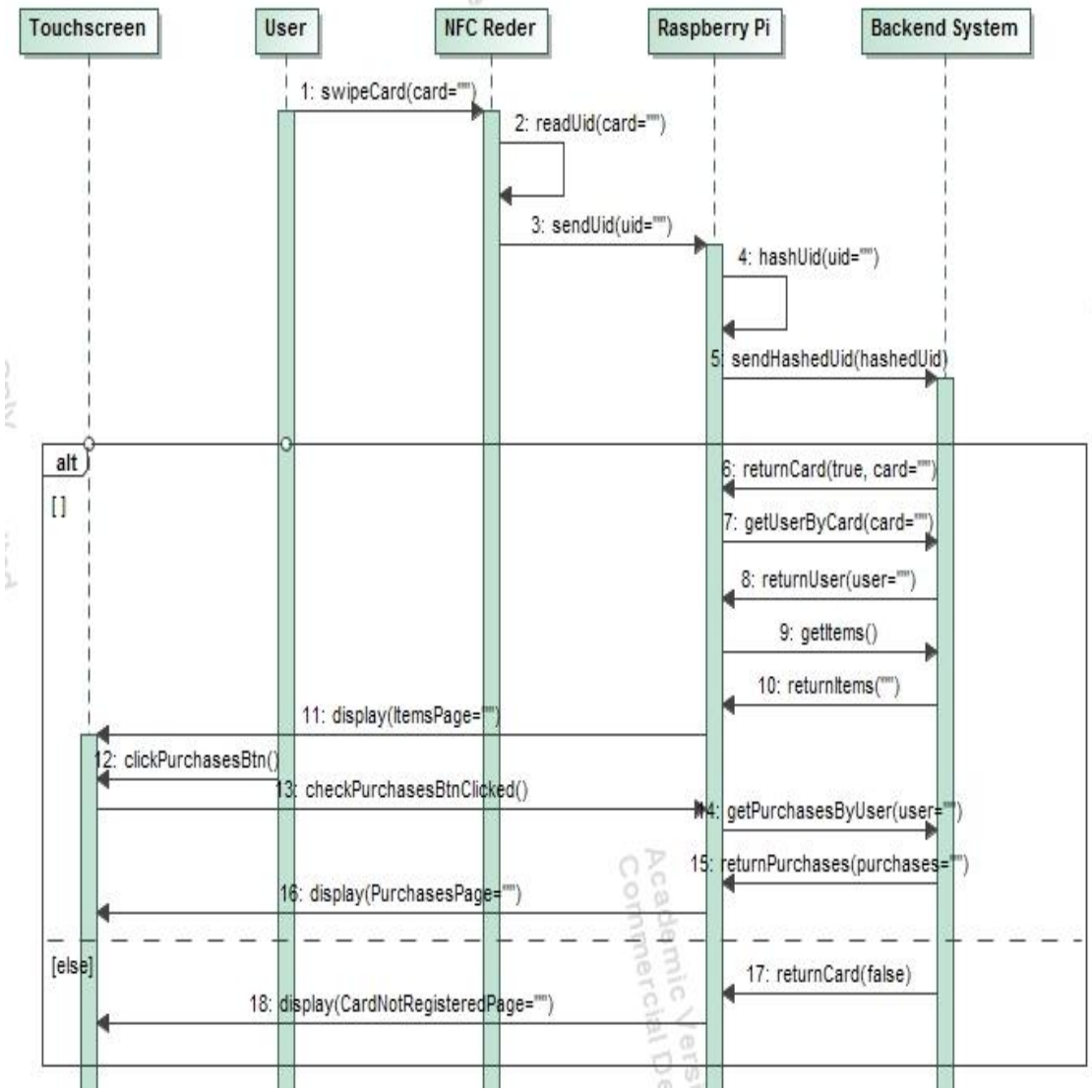

Figure 11– Check previous purchases sequence diagram

5 Software implementation

5.1 Choice of technologies for development

Choosing appropriate technology for a particular task is one of the most crucial parts of a successful project. As I am developing three different applications I used different programming languages and technologies for each of them.

5.1.1 Choice of technology for the eKohvik Application

Before starting the eKohvik Application development I did research, even tried with some of the following programming languages and technologies. First of all, I installed windows IoT (Internet of Things) on Raspberry Pi and tried to develop application using Universal Windows Platform (UPW). In this technology apps are mostly written with C# programming language, and we use Visual Studio that makes software development convenient. Furthermore, in UWP, GUI (Graphic User Interface) development is done using XAML. XAML is quite easy to use and we can build quite intuitive user interface by using it. Applications that are developed in UWP run almost on all devices that are running on windows 10. But it is not cross platform which means these applications don't run on Linux and on Mac OS. Furthermore, NFC reader that I am using in development (ACR1251U USB NFC Reader II) does not have a driver for Windows IoT. Moreover, as the USB NFC reader is not a proximity device, I couldn't find any library for reading NFC tags from the card.

After facing problems with windows IoT and UPW, I installed Rasbian-Jessie on Raspberry Pi and started app development with Python. As python has Tkinter, GUI interface library built in, I first tried to do application development on it. But later I found out that Tkinter doesn't support multithreading and this library is quite old and it is difficult to build intuitive user interfaces with it. While Python and Tkinter were used in a proof of concept software project for experimenting with NFC tags and readers, it seemed not to be good approach to follow for a more complex application.

Python has several GUI libraries that are used in desktop application development. I tried PyQt4 for GUI development but it has also the same problem that it doesn't support multithreading.

Moreover, as we are using the Touch screen, one of the main requirement is, to show keyboard on the screen only when we tap on input field and it should be closed as we

submit the form that Raspberry Pi doesn't have one built in on all Operating Systems. But there are some ready keyboard applications such as matchbox keyboard for Rasbian OS. I did research about how to activate and close the matchbox keyboard in code but couldn't find any useful information. As the result of this problems and requirements I had to do more research on different technologies that meet all the requirements and would make it convenient to develop desktop applications.

As JavaScript is a popular UI development tool, I came across the Electron framework that enables to write desktop applications using web technologies: HTML, CSS and JavaScript. Electron framework uses Chromium and Node.js.

Electron's advantages:

- Electron uses node.js. So you can import many modules.
- Distribution, with Electron you package and distribute the app yourself. An Electron App is packaged with its full environment
- Electron doesn't require Chrome installed.
- Electron has reasonable developer tools for testing and debugging.
- Electron is an open source platform
- Electron documentation is much better compared to other frameworks documentations even though it's a much younger platform.
- Adoption: There are quite a lot of big and successful apps built on top of Electron such as Visual Studio Code, GitHub client, Slack.

In addition, there exist several node.js modules that work well with Electron for reading NFC tag of the card using USB NFC reader.

5.1.2 Choice of technology for Web Portal

Nowadays, most websites are not built with multiple pages; instead, building web site as a single page web application has become mainstream. There are some popular JavaScript frameworks to build single page web applications. For example: Backbone, React, Angular.js etc. I decided to develop the Web Portal using Angular.js, as I am familiar with the framework and it is mostly used in building single page web applications. Angular.js is an open source framework and it provides a rich variety of resources to build a single page web application based on our requirements.

Pros and cons of Angular.js:

Pros:

1. Development is fast and simple as you get familiar with it
2. You write less code for same result as with other libraries
3. Testing is easy
4. It is good for developing highly interactive client side code
5. It supports two-way data binding
6. It manipulates DOM elements
7. Server performance is faster
8. It implements Model View Controller architecture
9. It has dependency injection system
10. It extends HTML

Cons:

1. Steep learning curve
2. It is difficult to debug scopes
3. JavaScript support on the browser is mandatory

5.1.3 Choice of technology for the Backend Application development

There are a lot of programming languages and technologies to build Restful backend applications and all of them have their advantages and disadvantages. However, choosing one is mostly up to system designers and developers. As a software designer and developer I chose Spring Boot to develop the eKohvik Backend Application.

Spring Boot is new generation of Spring framework. As we know spring is based on Java programming language. It is a popular framework for building large applications.

Spring has a lot of advantages:

1. Predefined templates
2. Loose coupling
3. Easy to test
4. Lightweight
5. Fast development
6. Powerful abstraction
7. Declarative support

However it has also some drawbacks that make development more difficult. As a main disadvantage of Spring framework we can say that configuration takes lots of time and knowledge. Configuration can be accomplished using 3 different ways:

1. Using xml configuration file
2. Using Annotations
3. Using Java configuration

And here Spring Boot comes for help. In spring boot we have all the advantages of Spring Framework and it provides default configuration for the project. We can generate Spring Boot application with online generator (https://start.spring.io/) or by using Spring Tool Suite IDE (https://spring.io/tools/sts)

These are features of Spring Boot coning from Spring Boot documentation:

- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Underflow directly (no need to deploy WAR files)
- Provide opinionated 'starter' POMs to simplify Maven configuration
- Automatically configure Spring whenever possible
- Provide production-ready features such as metrics, health checks and externalized configuration
- Absolutely no code generation and no requirements for XML configuration

5.2 Software Development

5.2.1 eKohvik Application software development

In order to start development we have to prepare software development environment. First thing is to install all required software and drivers for hardware.

3.1.1.1 Rasbian-jessie version 2017-01-11 is installed on Raspberry Pi

3.1.1.2 Driver is installed on raspberry pi for ACR1251U USB NFC Reader II

3.1.1.3 Pcsc-lite 1.8.4 is installed on raspberry pi

3.1.1.4 Pcsc-tools is installed on raspberry Pi

3.1.1.5 Node.js version 7.4 is installed on raspberry pi

3.1.1.6 Npm version 4.0 is installed on Pi

3.1.1.7 Electron version 1.5.1 is installed on Pi.

Now everything is ready for development of the eKohvik Application.

5.2.2 The Backend Application development

A new Spring Boot application is generated using Spring Tool Suite IDE and it has following dependencies:

1. spring-boot-starter-data-jpa
2. spring-boot-starter-web
3. spring-boot-devtools
4. spring-boot-starter-test
5. spring-boot-starter-security
6. mysql-connector-java
7. spring-jdbc
8. jjwt
9. spring-boot-maven-plugin

This application contains controller, service, model, repository, dto, service.imp and security packages. Figure 12 describes application layers with packages.

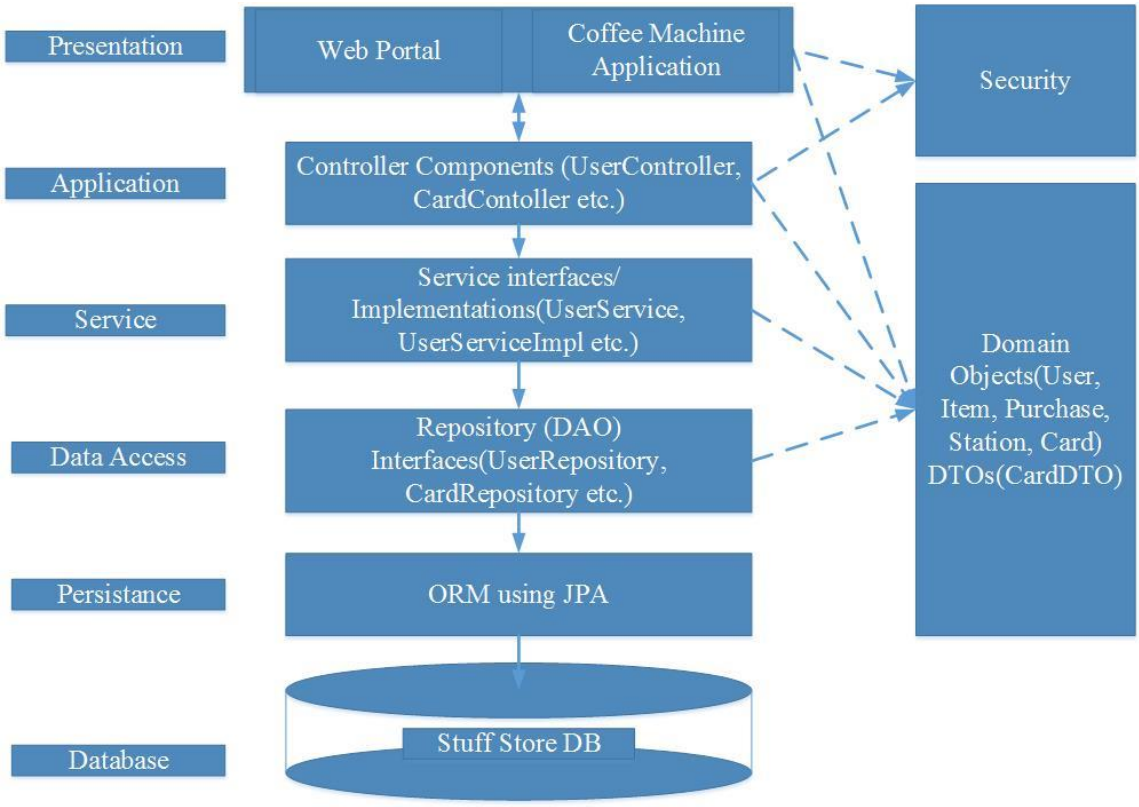

Figure 12 –The eKohvik Application Layered Architecture Diagram

5.2.3 The Web Portal application development

The Web Portal is developed inside the eKohvik Backend Application. It uses Angular.js and Bootstrap. This application is developed in Angular.js with MVC (Model View Controller) structure. It has main model that runs when application starts on the browser.

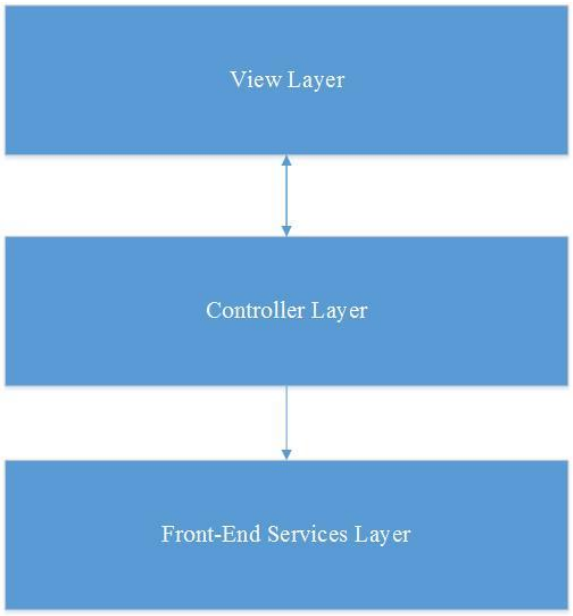

Figure 13 – The Web Portal Application Layered Architecture Diagram

5.2.4 Source code of the projects

The eKohvik Application - https://github.com/Khasanboy/eKohvikApplication
The eKohvik Backend Application -
https://github.com/Khasanboy/eKohvikBackendApplication

6 Summary

Point of sale systems play vital role in retail industry and developing a POS system to a certain environment requires doing more research before starting system development. In our system development it is supposed that customers use the application honestly, and for this reason, we tried to build the application as simple as possible from users' perspective.

Moreover, using RFID technology in the system for authentication was main requirement and developing prototype for usage of RIFD authentication into different services was one of the main goals of this work. Before starting the development software requirements specification for the system was prepared according to the needs and usage environment. However, during the software development some of the requirements changed and became more specific.

Furthermore, using Raspberry Pis with touch screen as a station in the self service POS systems is something new and because of that it required the system development to be started from the beginning.

Final product enables staff members of the Department of Software Science of TTU to buy items in common rooms with their staff ID cards, public transportation cards or any other RFID card that can provide a constant UID. Moreover, users can register themselves into the system using the Raspberry Pis located in common rooms of the faculty and add another card into their accounts if they forget registered ones and wants to use the system with another cards. Besides these users can check their current balance or list of purchases they have made using the stations instead of meeting system administrators.

System facilitates system administrators control whole system easily and retrieve any information if needed.

In the future, this system can be extended with other services and systems easily and another type of authentication or two factor authentication also can be supported.

References

[1] M. Zrivan and Z. Erlich, "Identification and Authentication: Technology and Implementation Issues," *Communications of the Association for Information Systems,* vol. 17, pp. 90-105, 2006.

[2] H.-L. Shieh and Y.-C. Liao, "RFID restaurant POS system," in *Machine Learning and Cybernetics International Conference*, Guilin, China, 2011.

[3] A. Gerald G, "Consumer-Centric Rfid Point of Sale Transaction System and Method". United States of America Patent 20080191878, 14 August 2008.

[4] U. B. Ceipidor, C. M. Medeglia, A. Marino, S. Sposato and A. Moroni, "A protocol for mutual authentication between NFC phones and POS terminals for secure payment transactions," in *2012 9th International ISC Conference on Information Security and Cryptology*, Tabriz, 2012.

[5] M. Dearing, G. Vidugiris, W. A. Linton, J. Linton and J. E. Krueger, "RFID point of sale and delivery method and system". United States of America Patent US 7791479 B2, 7 September 2010.

[6] K. Finkenzeller, RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, Sons Ltd, 2003.

[7] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing ,* vol. 5, no. 1, pp. 25-33, 2006.

[8] S. Ahuja and P. Potti, "An Introduction to RFID Technology," *Communications and Network,* vol. 2, pp. 183-186, 2010.

[9] A. Telba and K. Jamil, "Radio Frequency Identification Design and Simulation," in *World Congress on Engineering 2014*, London, 2014.

[10] "Electron Documentation," Github, 2017. [Online]. Available: https://electron.atom.io/docs/tutorial/about/. [Accessed 25 4 2017].

[11] S. Kinney, "What is Electron," in *Electron In Action*, Manning, 2016.

[12] "Angular.js Project," Google, 2017. [Online]. Available: https://angularjs.org/. [Accessed 20 4 2017].

[13] A. Lerner, ng-book - The Complete Book on AngularJS, Fullstack io, 2013.

[14] F. Gutierrez, Pro Spring Boot, Apress, 2016.

[15] "Spring Boot," Pivotal Software, 2017. [Online]. Available: http://projects.spring.io/spring-boot/. [Accessed 20 4 2017].

[16] "PSCS lite project.," 2015. [Online]. Available: https://pcsclite.alioth.debian.org/pcsclite.html. [Accessed 20 4 2017].

[17] "npm," npm Inc, 2017. [Online]. Available: https://www.npmjs.com/package/nfc-pcsc. [Accessed 10 4 2017].

[18] "Raspberry Pi ($35 computer)," 6 2016. [Online]. Available: http://whatis.techtarget.com/definition/Raspberry-Pi-35-computer. [Accessed 28 4

2017].

[19] “RASPBERRY PI TOUCH DISPLAY,” RASPBERRY PI FOUNDATION, [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-touch-display/. [Accessed 20 4 2017].

[20] “ACR1251U USB NFC Reader II,” Advanced Card Systems Holdings Limited, 2017. [Online]. Available: http://www.acs.com.hk/en/products/218/acr1251u-usb-nfc-reader-ii/. [Accessed 5 4 2017].