**Архитектурно описание**

Тема:

**Банков софтуер**

Група:76

Проект №: 5

Автори:

Анелия Бориславова Петрова, 471219071

Димитър Димитров Узунов, 471219003

Димитър Тодоров Тодоров, 471219053

Стилян Бориславов Саравски, 471219020

Хайрабед Хартюн Агопов, 471219069

26.10.2021г.

Contents

[Въведение 2](#_Toc86412000)

[Предназначение 3](#_Toc86412001)

[Обхват 3](#_Toc86412002)

[Актьори 3](#_Toc86412003)

[Архитектурен обзор 3](#_Toc86412004)

[Use-Case изглед 4](#_Toc86412005)

[Логически изглед 4](#_Toc86412006)

[Изглед на процесите 5](#_Toc86412007)

[Нефункционални изисквания 6](#_Toc86412008)

# Въведение

С напредването на технологиите, широкият пазар има нужда да модернизира своите услуги. За да бъде по-надеждна една банкова институция ,тя трябва да предостави на своите клиенти улеснен достъп до паричните им средства и снетки. Това налага въвеждането на модернизирана онлайн система за управление на личните финанси в мрежата. Софтуерът ще трябва да има лесен за използване интерфейс, който предоставя на потребителя пълният набор от услуги, които банката предлага.

# Предназначение

## Обхват

Целта на този документ е да предостави изчерпателен архитектурен преглед на системата за Банкова институция. Този документ описва как функционалният анализ и случаите на използване се превеждат и структурират в архитектурата от екипа за разработка.

Този документ представя техническата архитектура на софтуера. В този документ ние се фокусираме върху изборите, направени за него.

## Актьори

Този документ е предназначен да се чете от:

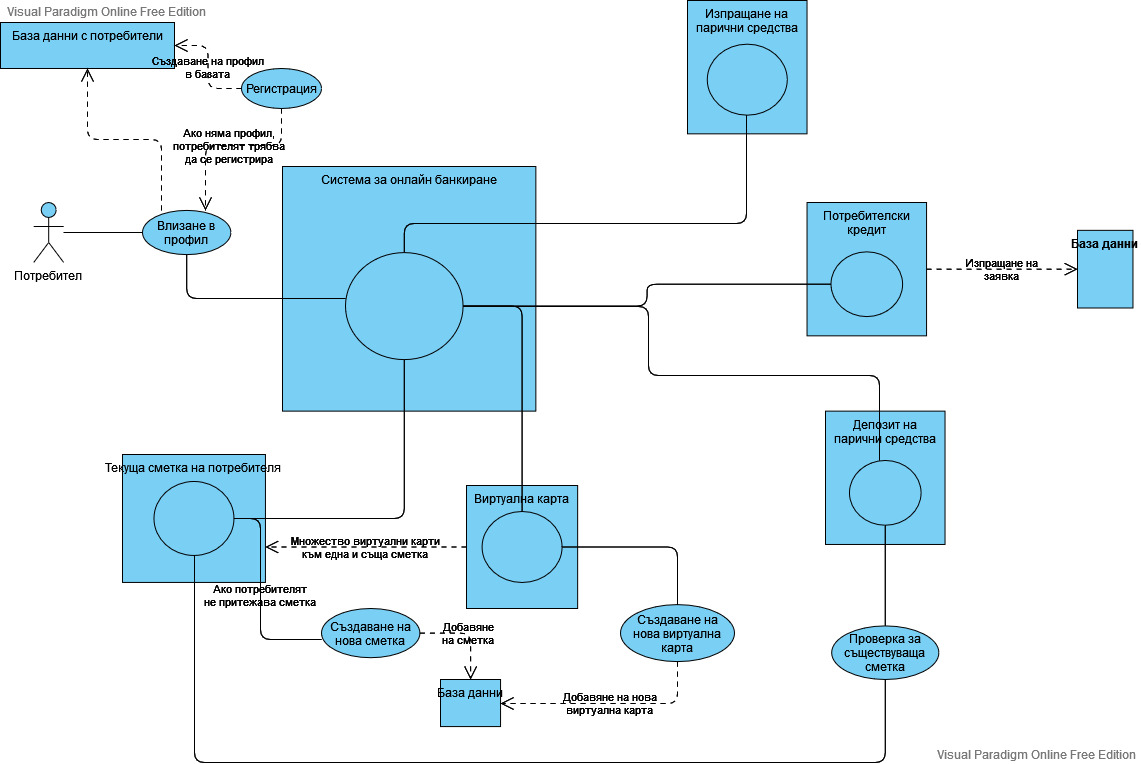
* Екипът, разработващ проекта
* Екип, оценяващ проекта (преподавателски екип)
* Клиенти на банката

# Архитектурен обзор

Този документ представя необходимите изгледи за представяне на софтуерната архитектура

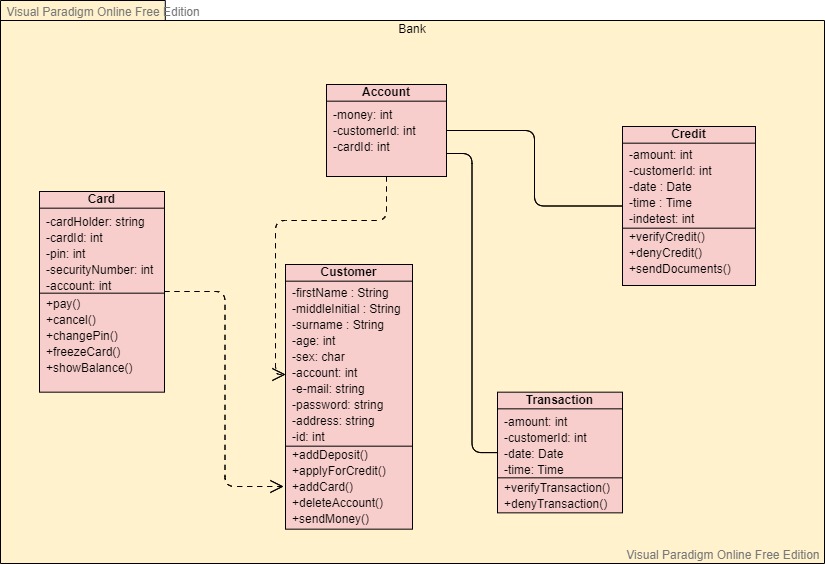
* Use-case изглед – представя функционалните възможност на софтуера (основни сценарии) от гледна точка на ползвателите (актьори).
* Логически изглед – софтуерът се представя като композиция от модули, пакети и класове чрез диаграми, за да се илюстрират връзките между архитектурно зависимите компоненти, в комбинация с текстово описание;
* Изглед на данните – описва перспективата за постоянно съхранение на данни на системата и връзката между тях
* Процесен изглед – представя алгоритмите, реализиращи архитектурно значимите функции.

# Use-Case изглед



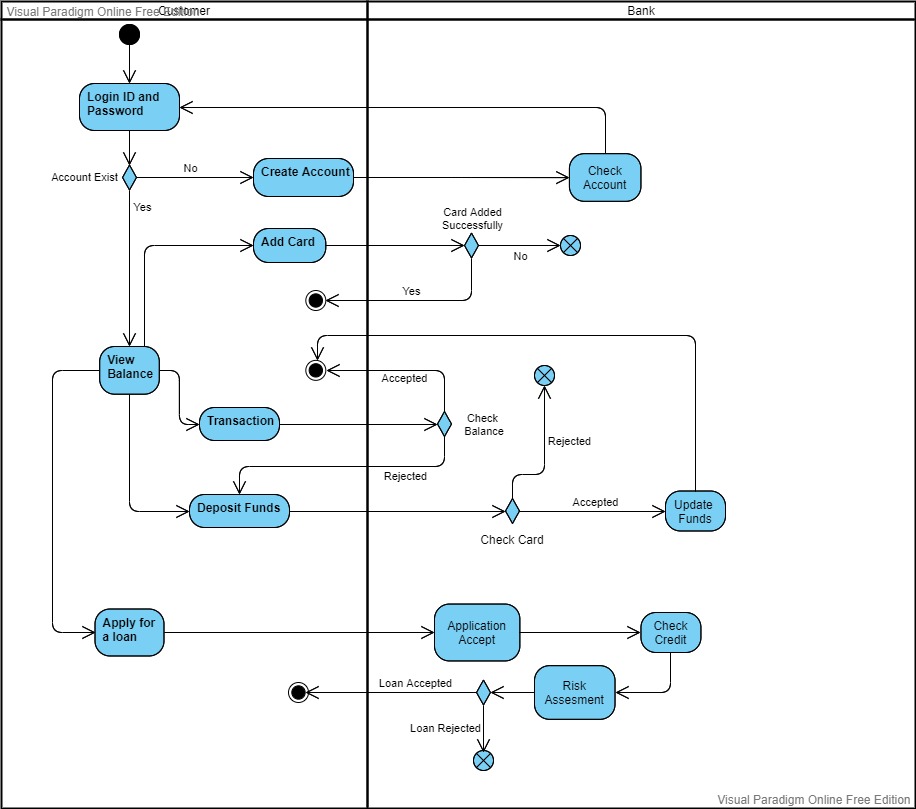
Потребителя влиза в профила си от там се проверява в базата данни с всички потребители. Ако няма профил, трябва да се регистрира и да създаде акаунт. При наличие на профил влиза директно в системата за онлайн банкиране. Има възможност за изпращане на средства. Също така и за потребителски кредит след изпращане на заявка и изчакване до базата данни. Друга функционалност е депозиране на парични средства, от където се проверява дали съществува сметка на потребителя. Има възможност и за виртуална карта, която се създава след проверка с базата данни и дали има съществуваща сметка. В противен случай трябва да се открие сметка.

# Логически изглед



Потребителят създава карта. Създавайки я, се открива сметка на негово име, чрез която може да прави различни транзакции и операции. Може да я използва и за заявка за кредит.

# Изглед на процесите



След въвеждане на данните, проверяваме дали съществува такава сметка. Ако не се създава акаунт и се проверява. Щом имаме акаунт, влизаме и можем да проверим с какви средства разполагаме. От тук имаме достъп до правене на транзакции, депозити и заявление за кредит. В случай на правене на депозит, проверява се картата и при успешно депозиране се обновява баланса на сметката. Ако не, получаваме отказ. При опцията за кредит, подаваме заявка и изчакваме одобрение, от там се проверява кредита и какъв е рискът, от където изходите са два - одобрен или не. Опцията да се добави карта е след влизане в акаунта. След въвеждане на данните, при успешно извършване на стъпките, получаваме карта. В противен случай- не.

# Нефункционални изисквания

Сигурност – данните на клиентите ни са от най-голямо значение за нашият проект. Данните, които ще притежаваме са твърде чувствителни и се нуждаят от сигурна защита, защото в неподходящи ръце биха им създали проблеми.

Достъпност – друго нужно качество на нашият софтуер е той да бъде максимално безгрешен. За тази цел ни е нужно максимално запознат екип, който бързо да открива и отстранява възникнали проблеми при потребителите.

Производителност – банковите уебсайтове и приложения са широко използвани в днешни дни, тоест ни е нужен един добър софтуер, който да издържа на максимално натоварване от едновременно влезли потребители.

Използваемост – дизайнът на един сайт е важен, за да улесни максимално своите клиенти, както и съответно да привлече нови. Идеята е максимално бързо да се осъществи желаната дейност от потребителя.