# BankApp

[BankApp](#_Toc88155780)

[1. Introduction](#_Toc88155781)

[1.1. Description](#_Toc88155782)

[1.2. Basic requirements](#_Toc88155783)

[1.3. User stories](#_Toc88155784)

[1.4. Technologies and tools](#_Toc88155785)

[2. Model](#_Toc88155786)

[3. Project plan](#_Toc88155787)

[4. Problems](#_Toc88155788)

## Introduction

### Description

This project aims to create a simple BankApp in Java.

The bank should be able to hold several customers with several accounts each.

### Basic requirements

* Only debit accounts
* Classes
  + Account
  + Customer
  + Bank
* Bank
  + Print list with all customers (SSN, first and last name) to textfile
  + Add customer with unique SSN
  + Change a customers name but not SSN
  + Delete existing customer and close accounts
  + Create account
    - Only existing customers
    - Unique account number, starting from 1001, 1002 etc
  + Close existing account, print saldo and remove account
* Customer
  + Select and show information about a customer
    - Name
    - SSN
    - Account numbers
      * Type of account
      * Saldo
  + Deposit money to an account
  + Withdraw money from an account if amount is covered

### User stories

0-99 Prio 1

100-199 Prio 2

200-299 Prio 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Story | As a <user> | I want to <perform a task> | so that I can <achieve a goal> | Acceptance criteria | Status |
| 0 | Teacher | see all options in the UI | select functions in my gradebook | * see all options in the UI, at least:   + list students     - select student     - read grade     - write grade     - add student     - remove student     - <home>   + read average grade   + read highest grade   + read lowest grade | x |
| 1 |  |  |  |  | x |
| 2 |  |  |  |  | x |
| 3 |  |  |  |  | x |
|  |  |  |  |  |  |
| 100 | Teacher | be able to add a new student | add grades to new students | * input fields for student details * submit at least name and SSN * information stored in selected textfile * add/remove in the RecordCatalogue | x |
| 101 |  |  |  |  |  |
|  |  |  |  |  |  |
| 200 | Principal | see all teachers averages | see deviations |  |  |
| 201 |  |  |  |  |  |
| 202 |  |  |  |  |  |
| 203 |  |  |  |  |  |
|  |  |  |  |  |  |

### Technologies and tools

IDE - IntelliJ

Programming language - Java

Version control system - Git

Source code repository service - Github

Documentation - GoogleDocs, drawIO

SCRUM board - ???

## Model

* 1. Classes
     1. Main

Handles the UI and runs the tasks

* + 1. Account

Describes the attributes of an account.

Holds accountnumber, saldo and account type.

It must be possible to

* + - * Withdraw money
      * Deposit money
      * Get Account Number
      * Get Information
        + Account Number
        + Saldo
        + AccountType.
    1. Customer

Name of customer.

SSN of customer.

A list of customer accounts.

It must be possible to

* + - * Change customer name
      * Get customer information
        + SSN
        + First and last name
        + Account

Account number

Account type

Saldo

* + - * Handle customer accounts
        + Create
        + Delete
        + more?

* + 1. Bank

List with customers.

Methods for handling customers and accounts.

Create necessary private methods.

Public methods must be at least

* **public List<String> GetCustomers()**
  + Returns all customers(SSN and full names) and accounts
* **public boolean AddCustomer(String name, long pNr)**
  + Creates customer
  + Only created if SSN does not exist
  + Returns true if customer created, else false
* **public List<String> GetCustomer(long pNr)**
  + Returns customer information

1. Full name + SSN
2. Account information following

* **public boolean ChangeCustomerName(String name, long pNr)**
  + Change customer name
  + Return true if changed, otherwise false
* **public List<String> RemoveCustomer(long pNr)**
  + Remove customer and accounts
  + Return info
    - Accounts information
    - Withdrawal amount
    - Interest amount
* **public int AddAccount(long pNr)**
  + Create account to customer with this SSN
  + Return generated accountnumber
  + Return -1 if no account created
* **public String GetAccount(long pNr, int accountId)**
  + Returns account information
    - accountId
    - Customer SSN
    - Account number
    - Saldo
    - Account type
    - Interest rate
* **public boolean Deposit(long pNr, int accountId, int amount)**
  + Deposit amount to accountId that belongs to customer SSN
  + Returns true if OK, else false
* **public boolean Withdraw(long pNr, int accountId, int amount)**
  + Withdraw money on account accountId that belongs to customer SSN
  + Return true if OK, else false
* **public String CloseAccount(long pNr, int accountId)**
  + Close account accountId that belongs to customer SSN
  + Return account number and withdrawn amount
    1. Filehandler

Methods for handling the files

* 1. UML

## Project plan

Milestones

* 1. Startup

Deadline 2021-11-18

* Read requirements
* Create documents
* Create project plan
* Setup environment
  1. Design

Deadline 2021-11-19

* Implement class models and interfaces
* Inheritance? Discuss design with teacher
  1. Implementation

Deadline 2021-11-

* Implement user stories
  1. Release

Deadline 2021-11-

* First release includes prio 1 user stories
* Publish on GitHub
* Presentation
* Handing in

## Problems

* Prestudy
  + Inheritance used?
    - Eg base classes could be
      * Account
      * CreditCard
* Planning
  + How to setup and work with SCRUM?
* Implementing