Mobile banking Application

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <17/03/2021> | <1.0> | Supplementary specification | Angela-Paula Modrîngă |
| <23.02.2021> | <1.1> | Supplementary specification | Angela-Paula Modrîngă |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 4

2.4 Testability 4

2.5 Usability 4

3. Design Constraints 4

Supplementary Specification

# Introduction

In this document, the specifications of the Mobile Banking application will be presented. Those include attributes like availability, performance, security, testability and usability, the non-functional requirements of the application and the design constraints.

# Non-functional Requirements

## Availability

The system will be available for free, and each person with a bank account will be able to use it on their smartphone/tablet after creating an account. The application will be responsive.

## Performance

## Security

Each time the user wants to enter the application, a log-in is required. The user needs to introduce his client code and the password.

## Testability

## Usability

Useful for people that want to manage their financial resources and don’t have, or don’t want to spend, time to go to the bank office.

# Design Constraints

The project will be a mobile application, having a friendly user interface. The source code will be written in Android Studio using Java. The application will be connected to a SQLite Database, that will store all the needed i We will keep track of the development phases using Github.

* Constraint Layout will be used to ensure responsiveness.
* Singleton Design Pattern will be used to provide high level security, because the application needs to make sure that an account is accessed by only one customer at a particular duration.
* Façade Pattern will be used to hide the complex database access interface behind a few easy to understand and maintainable interface