

**SEN2212**

**Data Structure and Algorithms II**

**Project Report**

Group No: 21

Project Title: Ariel Banking System

Lab Section No: Student ID: Student Full Name:

902 1904332 Emir Ata Yalçın

904 1806800 Arda Taşdemir

1. Introduction
   1. Purpose/Project Proposal

*Our aim in this project is to provide a look that will appeal to all kinds of users of all ages and to provide users with ease of use. In our application, you have an account number defined for everyone and a secure box or credit card information according to the account type you choose.*

* 1. Software Language/ Project Environment

*We used java for the program in general. While using css for GUI, our choice was MySQL for the database included in the program.*

* 1. Data Structures

*In our program, we used the hash map data structure to provide access to the accounts and the methods in them. In our map, our Key value is the accountNumber variable specific to all accounts taken from our Object, while Value is the Account object. The reason we use hash maps is that we didn't need a specific order while keeping our objects in the list, and we decided that this is the best data structure that we can easily call the object we want. We used the linked list data structure in our list, where account transactions are recorded, because we wanted to store every transaction we made, and we thought that the most correct choice would be the linked list, since we did not*

*know how many transactions to be made.*

* 1. Work Partitioning

*A list showing all business events to which the work.*

|  |  |  |  |
| --- | --- | --- | --- |
| *Name* | *Role* | *Date* | *Description* |
| *Emir Ata Yalçın*  *1904332* | *Back-end developer* | *31.03.2021-10.05.2021* | *I implemented the data structures and all windows of GUI.* |
| *Arda Taşdemir*  *1806800* | *Interface designer* | *31.03.2021-10.05.2021* | *I designed the GUI of the project using Java extensions.* |

1. Architectural Representation
   1. Use Case Diagram

Diagram

Description automatically generated

* 1. Class Diagram

*Text

Description automatically generated with medium confidence*

1. Application

*Login Screen*

*Graphical user interface

Description automatically generated*

*Register Screen*

*Graphical user interface, application

Description automatically generated*

*Main Screen(Checking Account)*

*A picture containing text

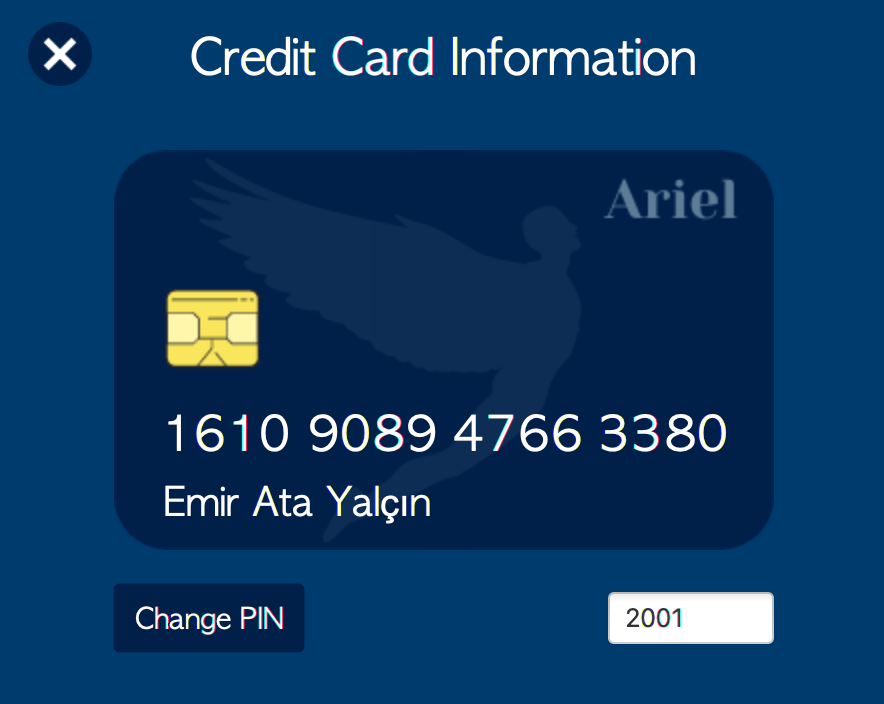
Description automatically generated*

*Main Screen(Saving Account)*

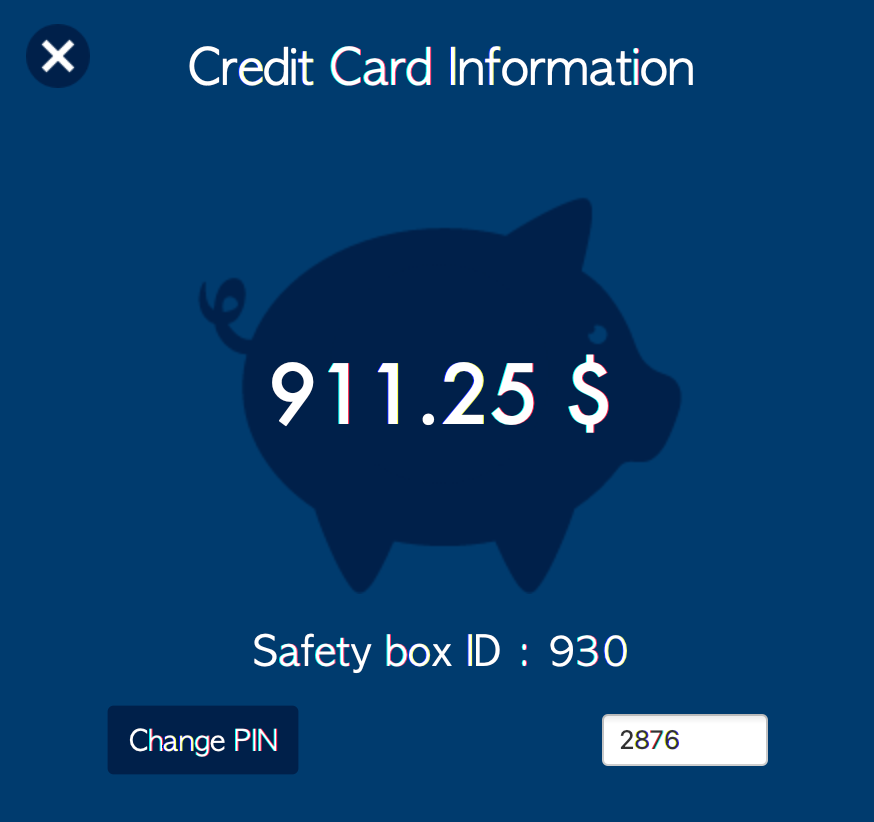
*A screenshot of a computer

Description automatically generated with medium confidence*

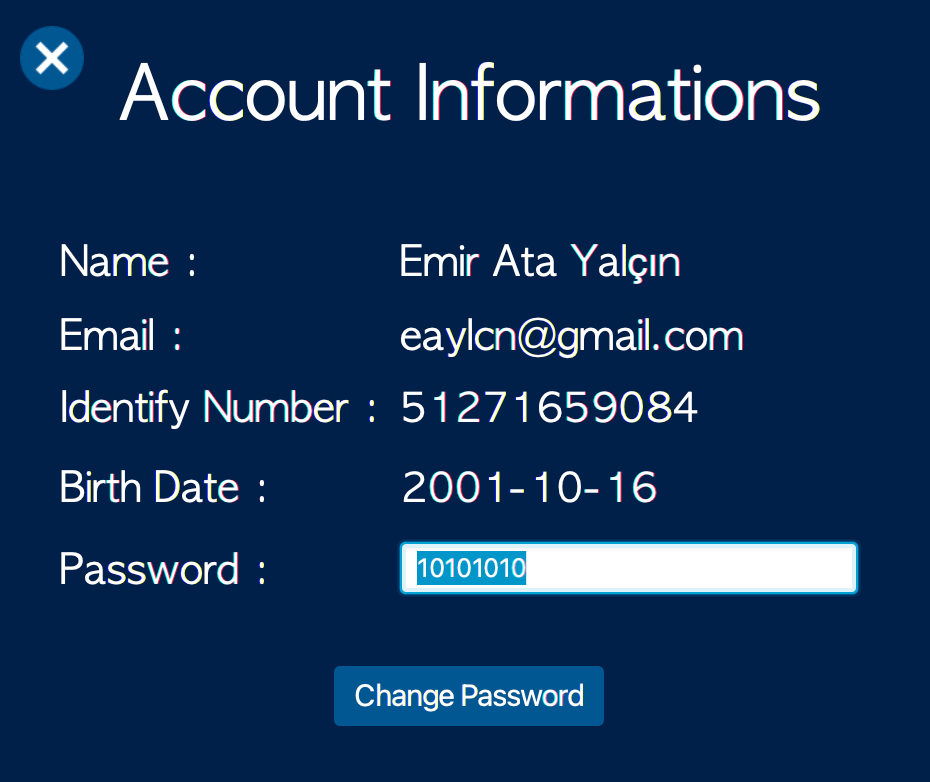
*Credit Card Information Screen*

**

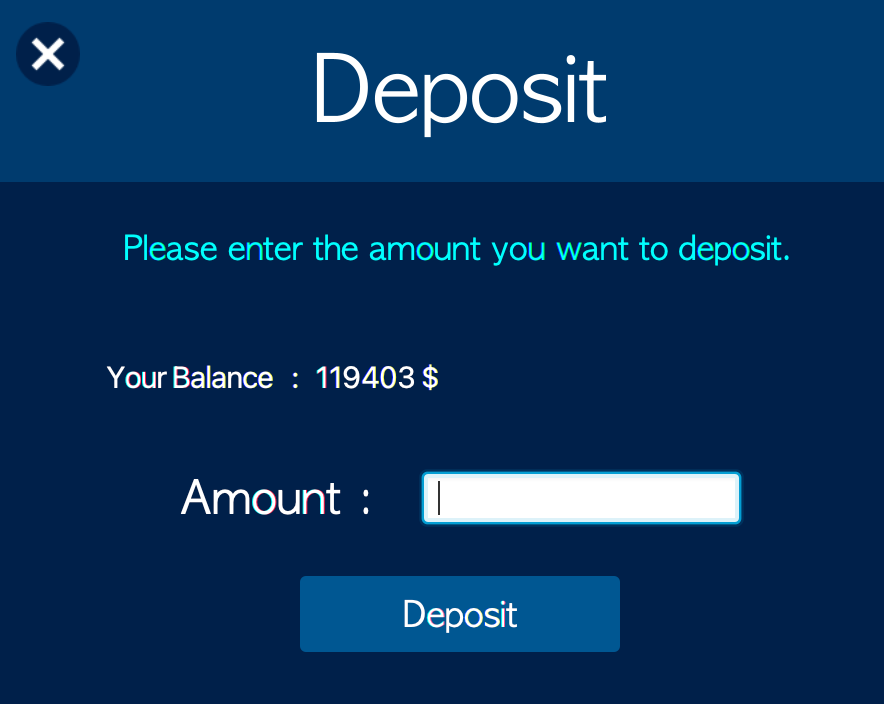
*Money Box Information Screen*

**

*Account Information Screen*

**

*Deposit Screen*

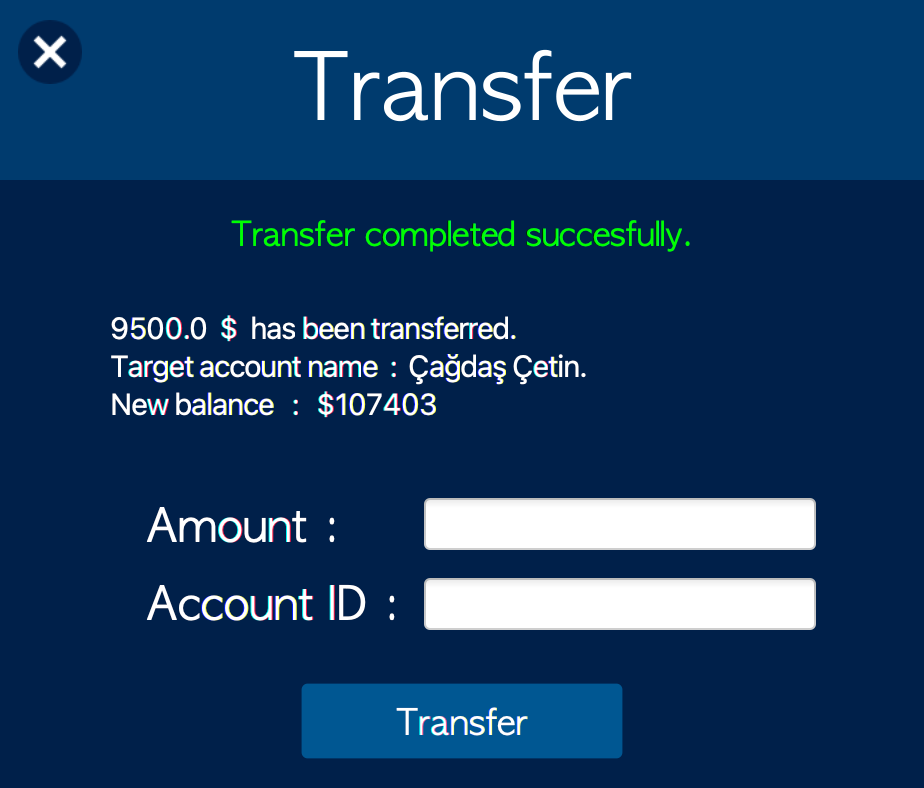
**

*Withdraw Screen*

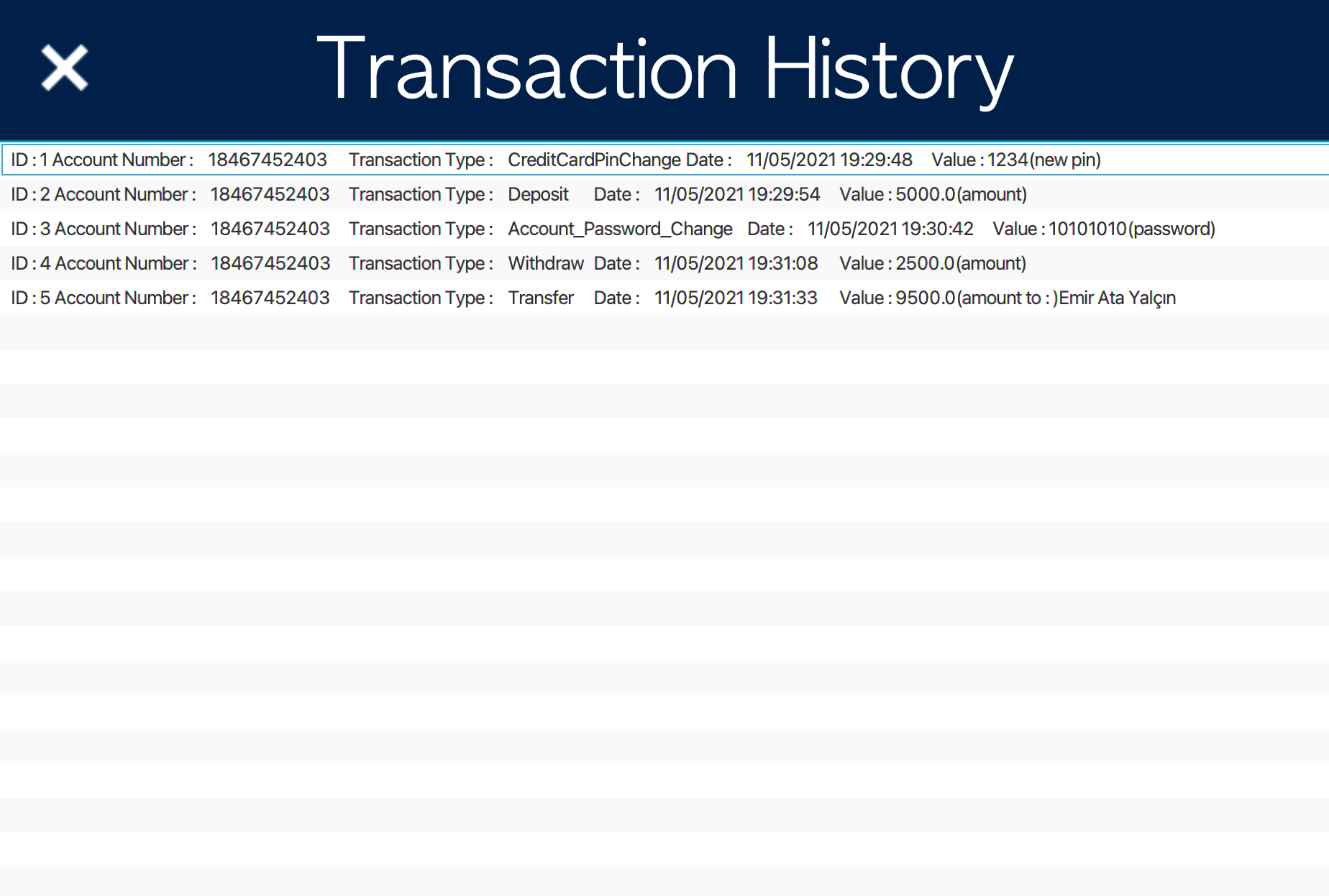
*Graphical user interface, application, website

Description automatically generated*

*Transfer Screen*

**

*Transactions List*

**

1. Conclusion / Summary

*Today, people care about simplicity and ease of use. In our application, we were aiming for ease of use with a strong simple design that appeals to everyone and we think we have achieved this. If we talk about the technical aspects of the work, we thought it would be very correct to use Linked lists, which is the main component of our method that enables it to control its transactions. Hashmap, on the other hand, seemed very logical to use our objects to move them to different windows and make transactions. This is all we will talk about the two most important components of our application, design and technique.*