Banking App Simulator

Software Engineering

ITSC-3155

Final Project Report

May 6th, 2020

Amanda Poteate

Levi Carpenter

**Table of Contents**

**1. INTRODUCTION 1**

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 Context Diagram and Data Flow Diagram 1

1.5 References 1

1.6 Overview 2

**2. GENERAL PROJECT DESCRIPTION 2**

2.1 project Perspective 2

2.2 project components 2

2.3 specific goals 2

2.4 overview of programs related to specific goals 3

2.5 Assumptions and Dependencies 3

**3. EFFORT 3**

3.1 EFFORT 3

**4. PROGRAMS DEVELOPED 4**

4.1 User Interfaces *4*

4.2 Comparison

11

**5. DISCUSSIONS AND CONCLUSION 25**

5.1 Challenges and Learning Experiences 25

**A. APPENDICES 26**

A.1 Appendix 1:Use Case 1

A.2 Appendix 2 Use Case 2

A.3 Appendix 3 Use Case 3

A.4 Appendix 4 Use Case 4

A.5 Appendix 5 Context Diagram

A.6 Appendix 6 Data Flow Diagram

# **1. Introduction**

*The introduction to this project …*

## **1.1 Purpose**

*The purpose of this project is to provide a user-friendly banking application to be used by both account holder and bank employee for managing funds.*

## **1.2 Scope**

*The software product in production is a mobile/desktop banking app simulation that is entitled “49er Credit Union,” named after the university developing the application. This app will provide user-friendly functions and display to account holders, as well as include employee functions. It does not involve some common bank app functions, such as money transfer to other individuals, for simplicity of implementation. One objective of developing this product is to determine what aspects of an application most impact user transparency and how to make crucial technological applications easier to access for all groups of people. A banking app that eliminates confusion would not only help users feel safer and more secure in handling their money, but would also open up remote banking services and internal employee positions for the elderly and those who aren’t as adept with technology.*

## **1.3 Definitions, Acronyms, and Abbreviations**

*Scrum: Scrum in general is a framework that enables effective, thorough development of quality products while addressing complex issues [4]. Scrum meetings refer to short team meetings that address what has been completed, satisfaction of current progress in relation to the timeline, and plan for future actions.*

*Covid-19: this refers to the global coronavirus pandemic starting in late 2019 and advancing through 2020 to current times, which affected many individuals’ ability to stay home (high-risk persons in particular)*

*MySQL: an open-source management system for databases*

*Dependencies: a task relationship in which one is required to complete another*

## 1.4 Context and Data Flow Diagrams

## *Context and Data Flow diagrams are included in appendices 5 and 6 for reference [7].*

## **1.**5 **References**

*[1] “Scrum Meetings,” L. Carpenter and A. Poteate. Attached file.*

*[2] “State Employees Credit Union,” mobile app resource. Website accessible at* [*https://www.ncsecu.org/*](https://www.ncsecu.org/)

*[3] “BB&T,” mobile app resource. Website accessible at* [*https://www.bbt.com/*](https://www.bbt.com/)

*[4] “What is Scrum?”* [*https://www.scrum.org/resources/what-is-scrum*](https://www.scrum.org/resources/what-is-scrum)

*[5] “Python Documentation,” online reference for researching Python language and features. Website accessible at* [*https://docs.python.org/3/contents.html*](https://docs.python.org/3/contents.html)

*[6] “MySQL,” online reference for database usage. Website accessible at* [*https://www.mysql.com/*](https://www.mysql.com/)

*[7] “Context and Data Flow Diagrams”, Levi Carpenter. Appendices 5 and 6 below.*

## **1.**6 **Overview**

*This program contains implementing an application that simulates typical banking functions. It is designed to help study how users interact with necessary applications like so and how to make applications more user-friendly.*

## ….

# **2. General Description**

*This is an overall description of the project…*

## 2.1 Project Perspective

*This project was tasked over the course of 6 weeks, divided into sections of completion. It began with a proposal and first demo, which included just the basic functions of a bank app (sign up, log in, and view balance). Second and Third stages involved implementing both user and administrative functions to the app, and compiling our development experience into a presentation and report, respectively.*

## 2.2 Project Components

*User functions within the application consist of viewing account balances, transferring money, accessing transaction history and logging out. Administrator functions consist of creating user accounts as well as withdraw and deposit funds capabilities.*

## 2.3 Specific Goals

*Our specific goals consisted not only of milestones but also of general development objectives. Our three milestones started with a prototype submission and continued forth to a final demo submission and a presentation. In terms of product, we hope to present a banking application that translates to easier technological use and feelings of safety to all age groups and life stages. Our aim of such an objective is to contribute to developing knowledge on user-friendliness within applications that must cater to a wide range of audiences. These goals are crucial to helping alleviate the stress that comes from a worldwide crisis such as the current Covid-19 pandemic. We spent time each week in meetings to evaluate progress on these goals.*

## 2.4 Overview of programs related to specific goals

*Each week, we held 3 meetings that lasted from 15 minutes to over an hour to update the team on progress and plans moving forward, detailed in the project Scrum sheet attached [1]. User friendly specifications and necessary app tasks were determined from preexisting banking applications, such as State Employees Credit Union [2] and BB&T [3].*

## 2.5 Assumptions and Dependencies

*There exists an audience of those who would benefit from an easy-to-use banking application, whether that be from lack of experience with technology or for simplicity of usage in general. It is also assumed that as a result of certain drastic crises, users and administrators should have access to an application handling their finances that is both safe and usable remotely. Finally, there exists assumptions that milestone deadlines are upheld at the scheduled times and that all team members are willing and able to contribute and do so effectively. In terms of dependencies, sections of development required completion before others could be begun. Sign up and Login functions, as well as proper connection to the MySQL [6] database were necessary to move onto the rest of the app tasks. Account creation on the administrative side was crucial for transferring funds capabilities.*

## **….**

# **3. Effort**

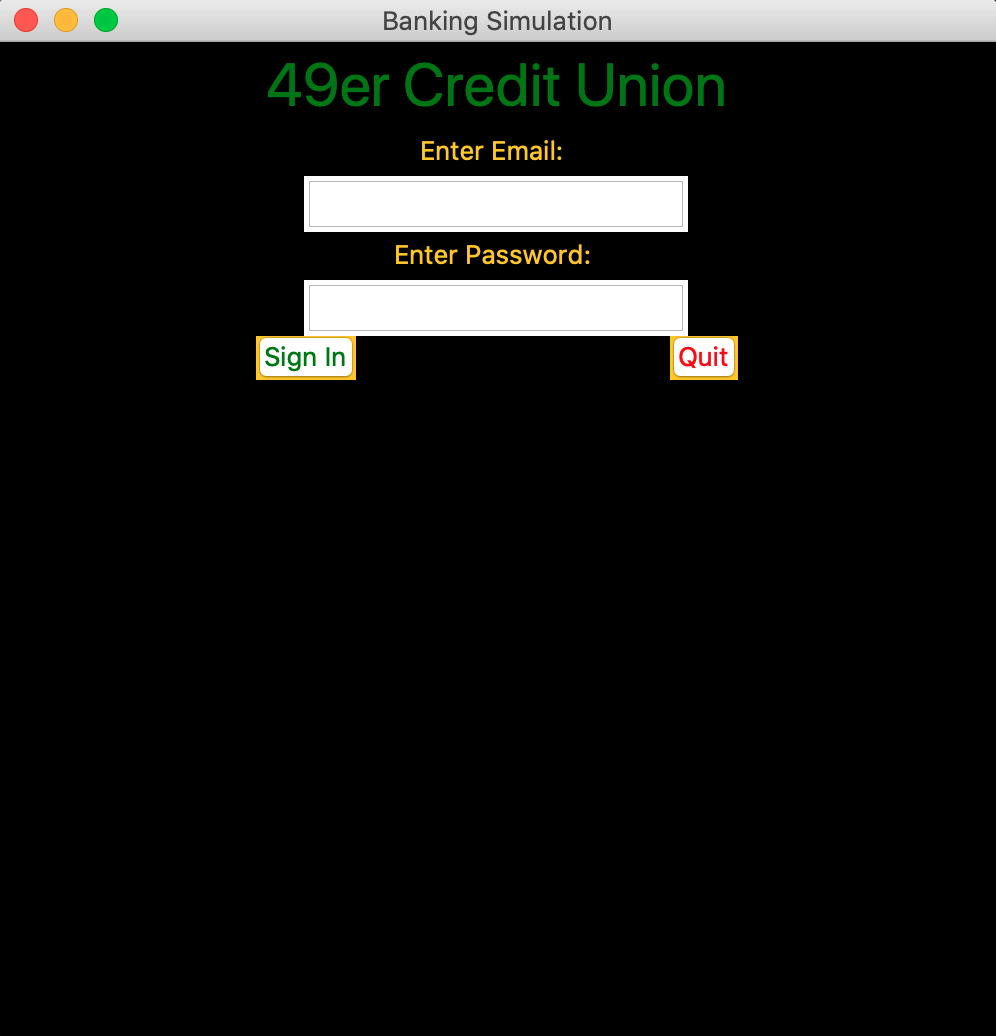
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Estimated Time of Research | Actual Time of Research | Estimated Coding Effort | Actual Coding Effort |
| Proposal | One week | 4 days | low | low |
| Login/Logout | one week | 9 days | low | medium |
| Display Balance | 2 days | 1 day | low | low |
| Deposit/Withdraw | 1 day | 4 days | medium | medium |
| Create account | 1 day | 1 day | medium | high |
| Transfer | 1 day | 1 day | medium | medium |
| Transaction History | 3 days | 2 days | high | medium |
| Test cases | one week overall | one week | medium | low |
| Logout | 1 day | 1 day | low | low |
| Presentation video | 4 days | 2 days | low | low |

## ….

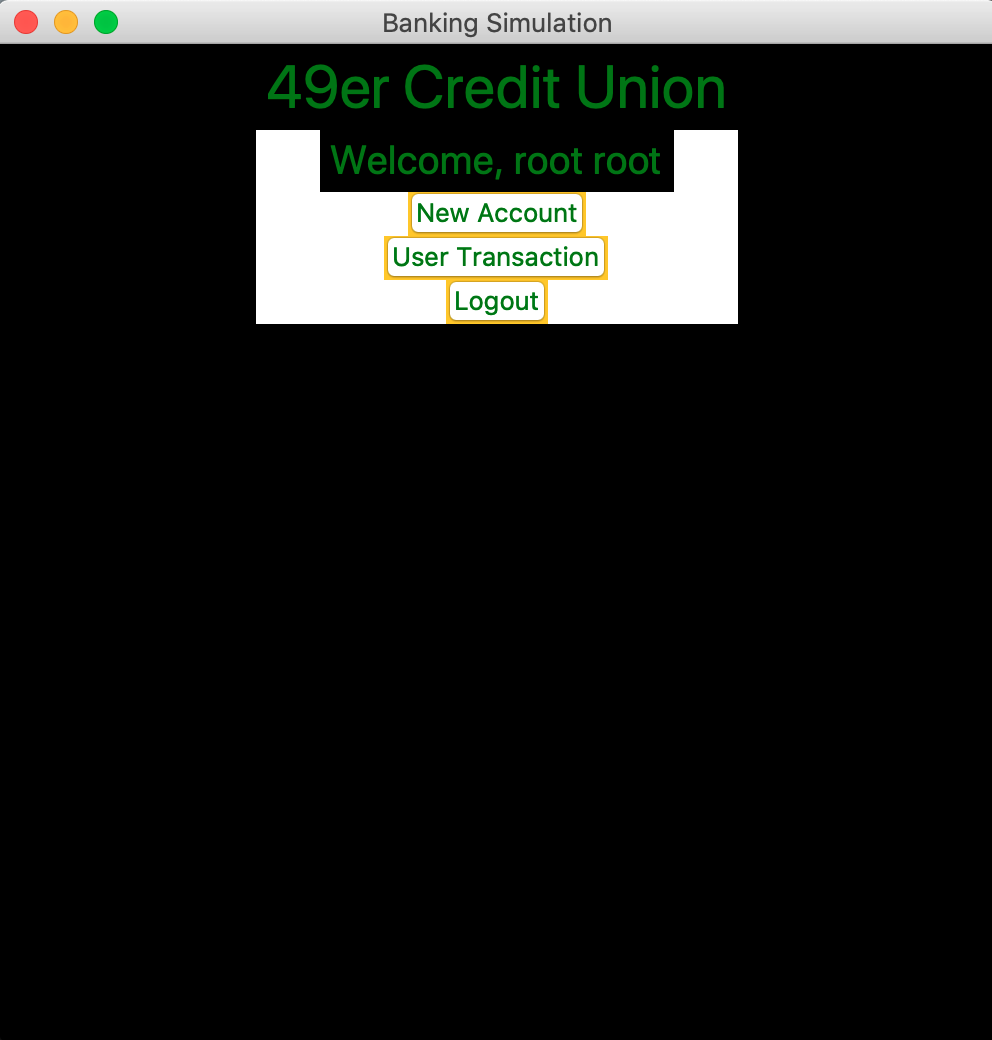
# **4. Programs Developed**

**4.1 User Interfaces:**

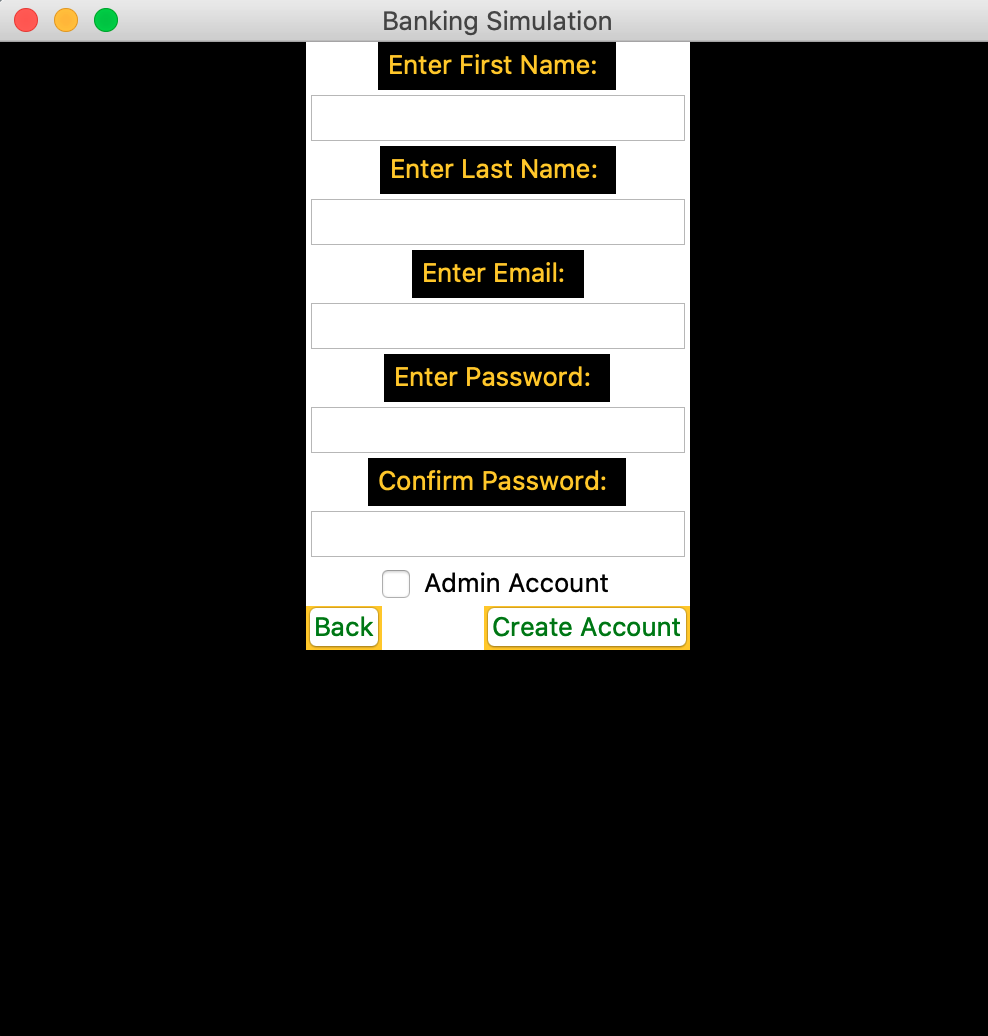
Sign in Window

**

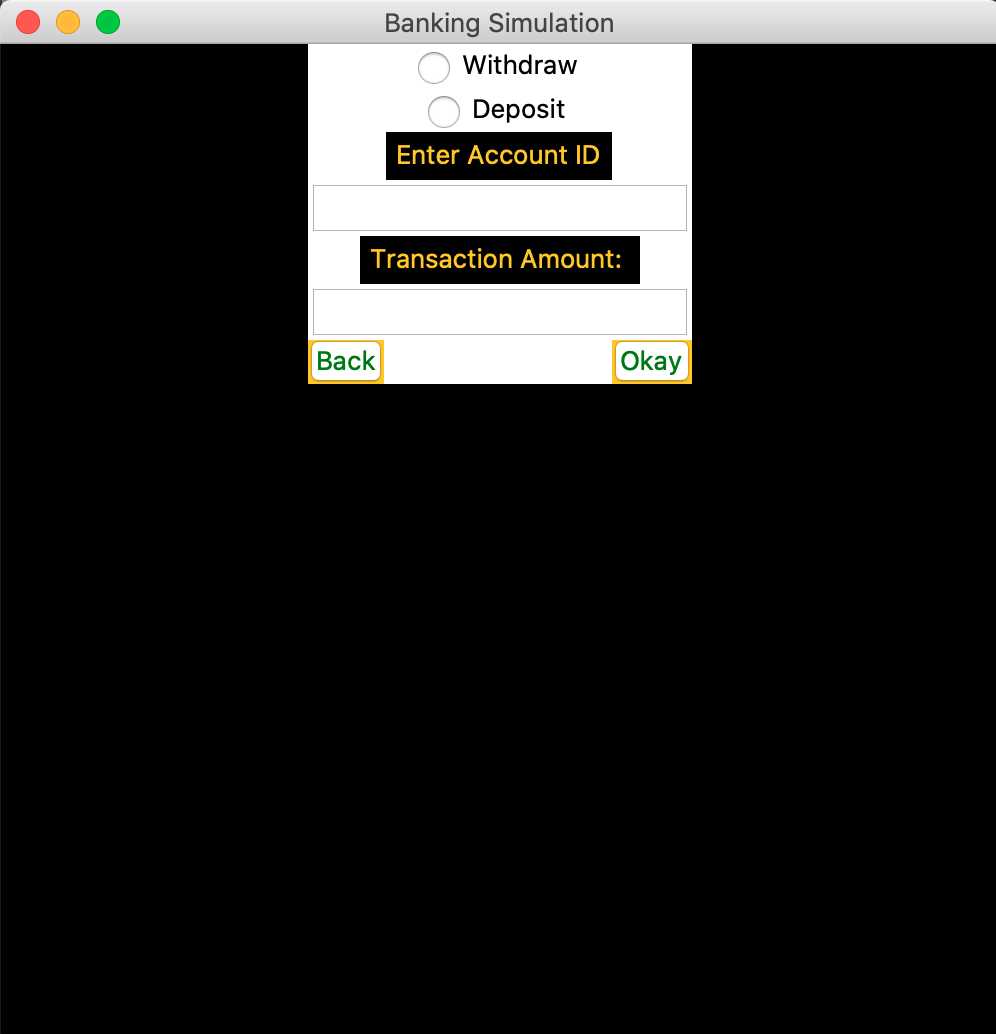
Admin Home Window



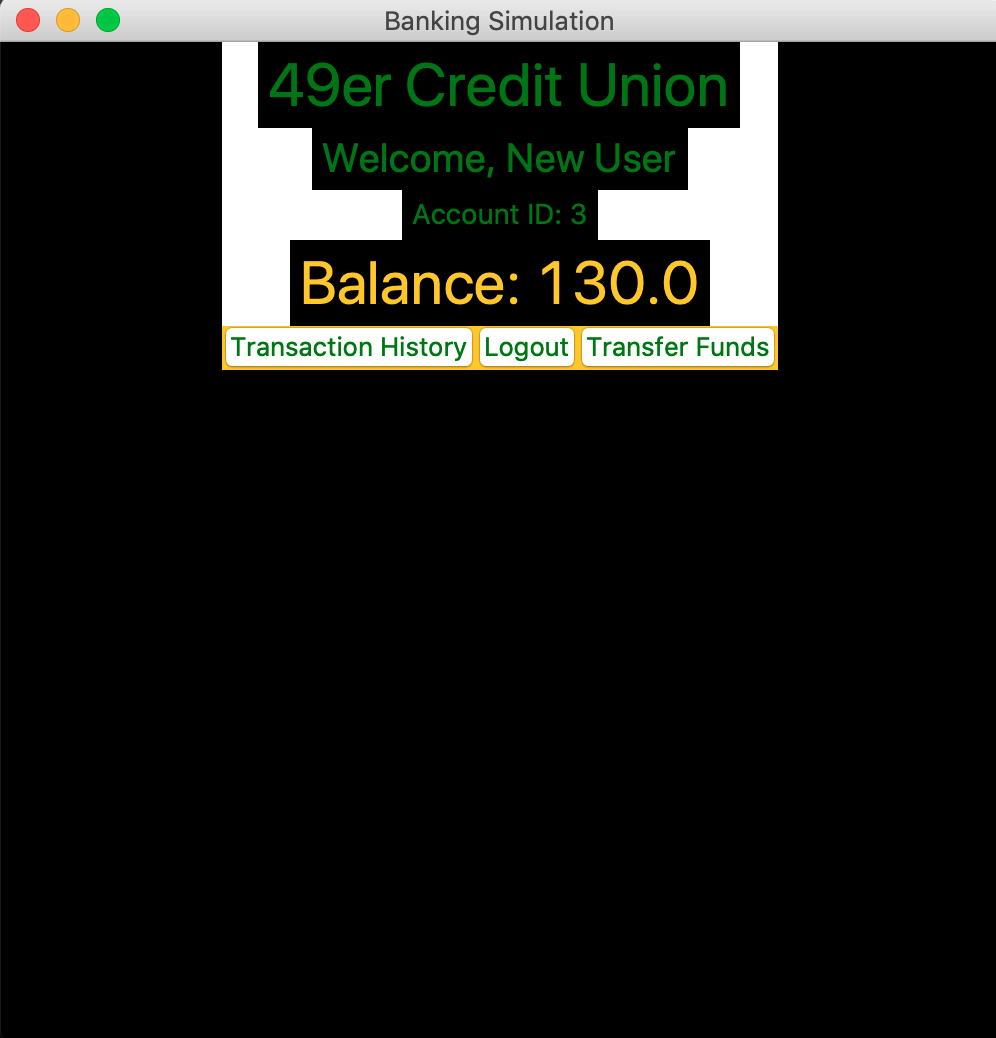
Create Account Window



User Transaction Window



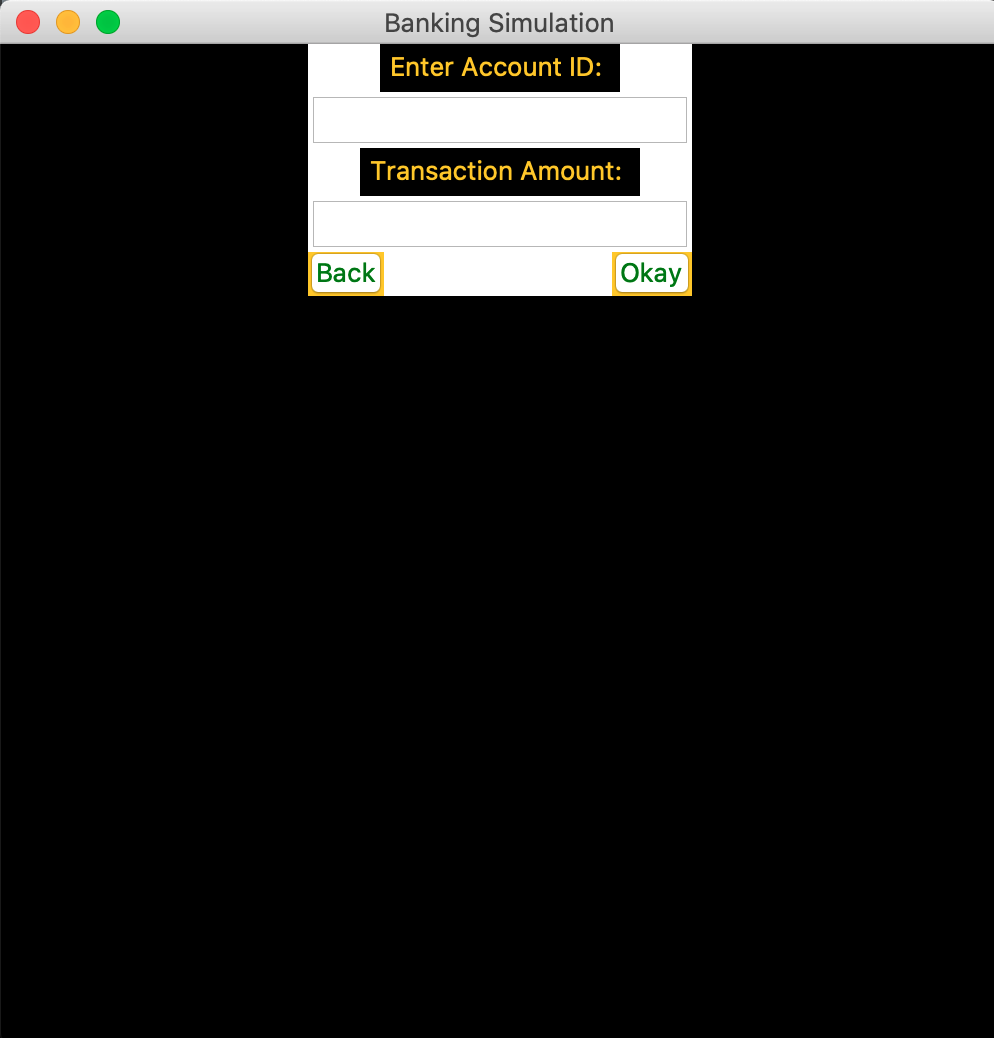
User Home Window



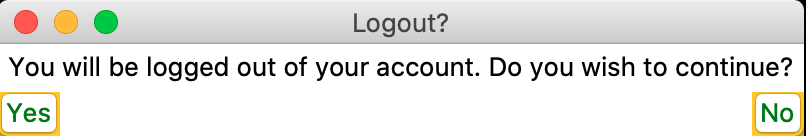
Transaction History Window



Transfer Funds Window



Logout Confirmation Window



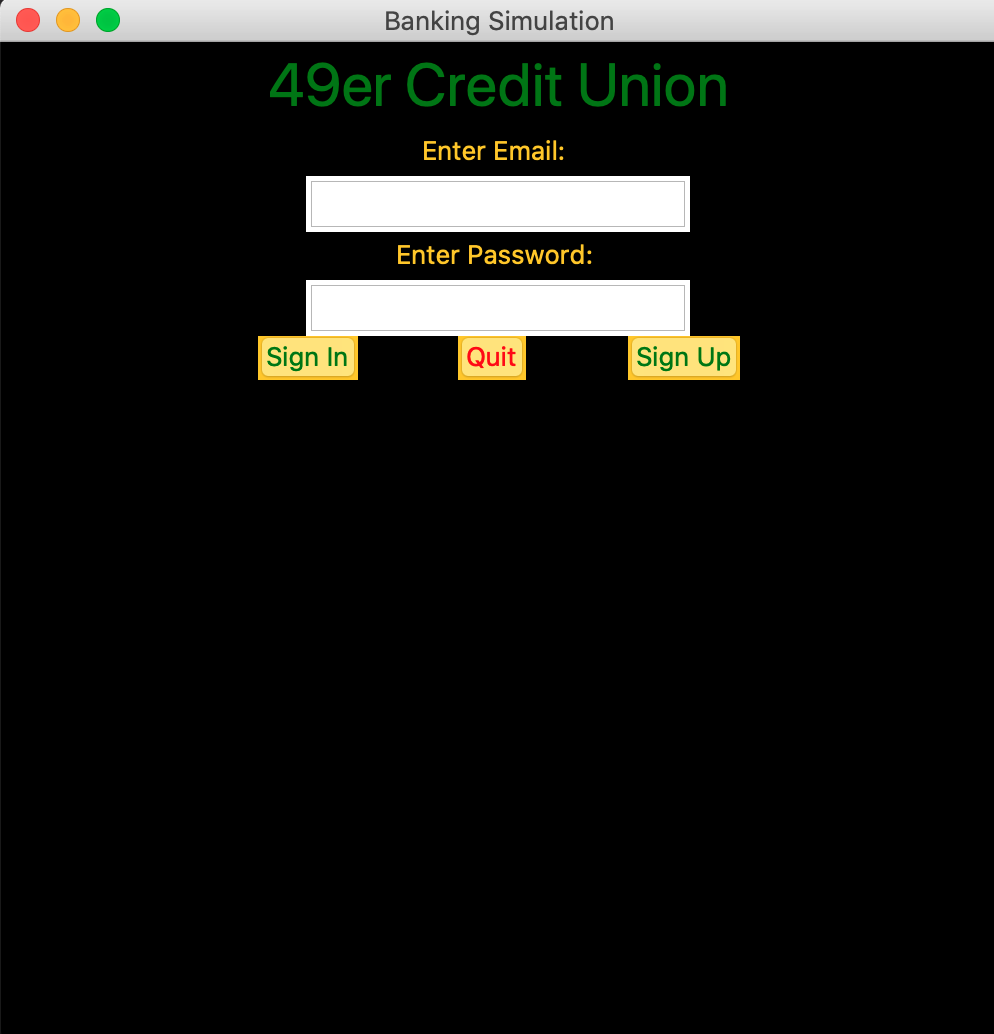
Error Window



**4.2 Comparison:**

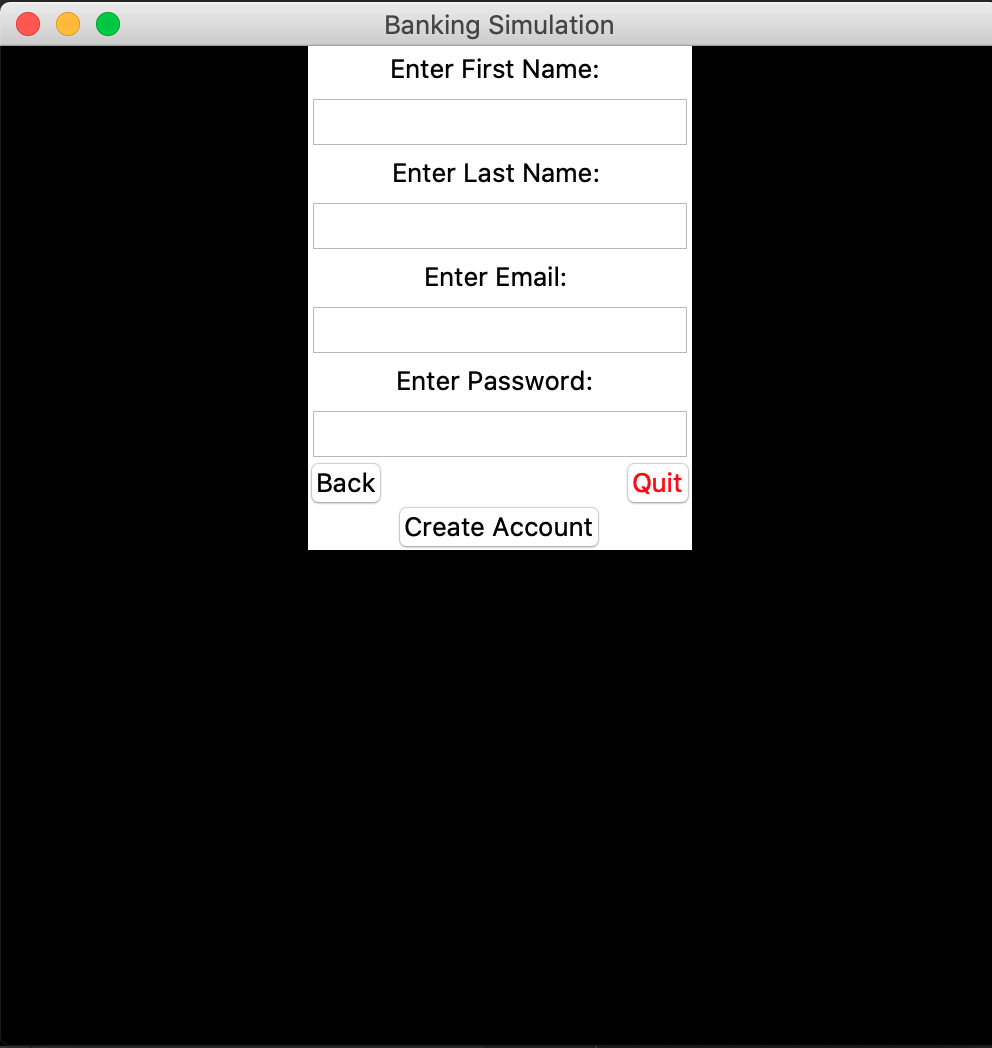
Sign In window

The signup button has been removed and the quit button has been moved to the right in the enhanced version as opposed to the older version.



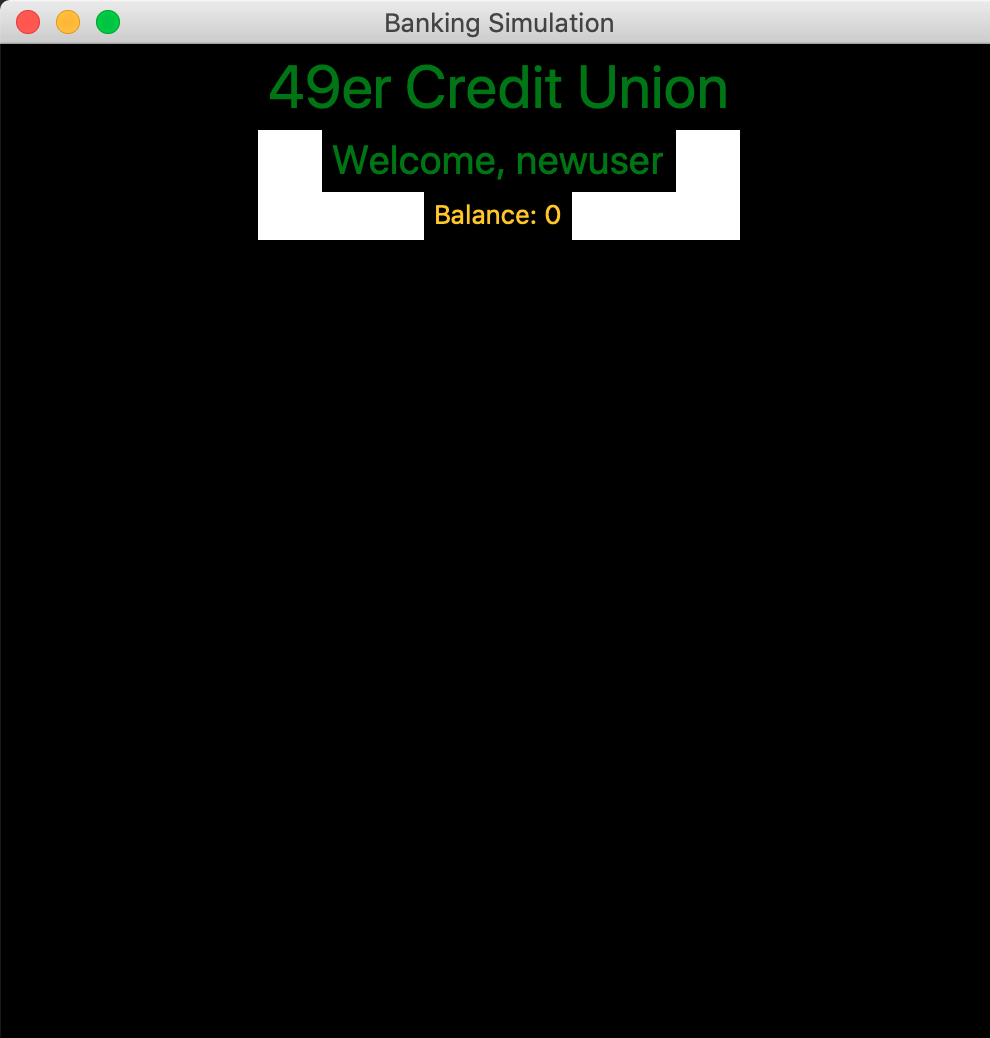
Sign Up Window

Compared to the older version, the enhanced version has added a confirm password entry and admin checkbox. The create account button has been removed and the quit button has been replaced by an okay button. Additionally, the color theme has been added to the enhanced version.

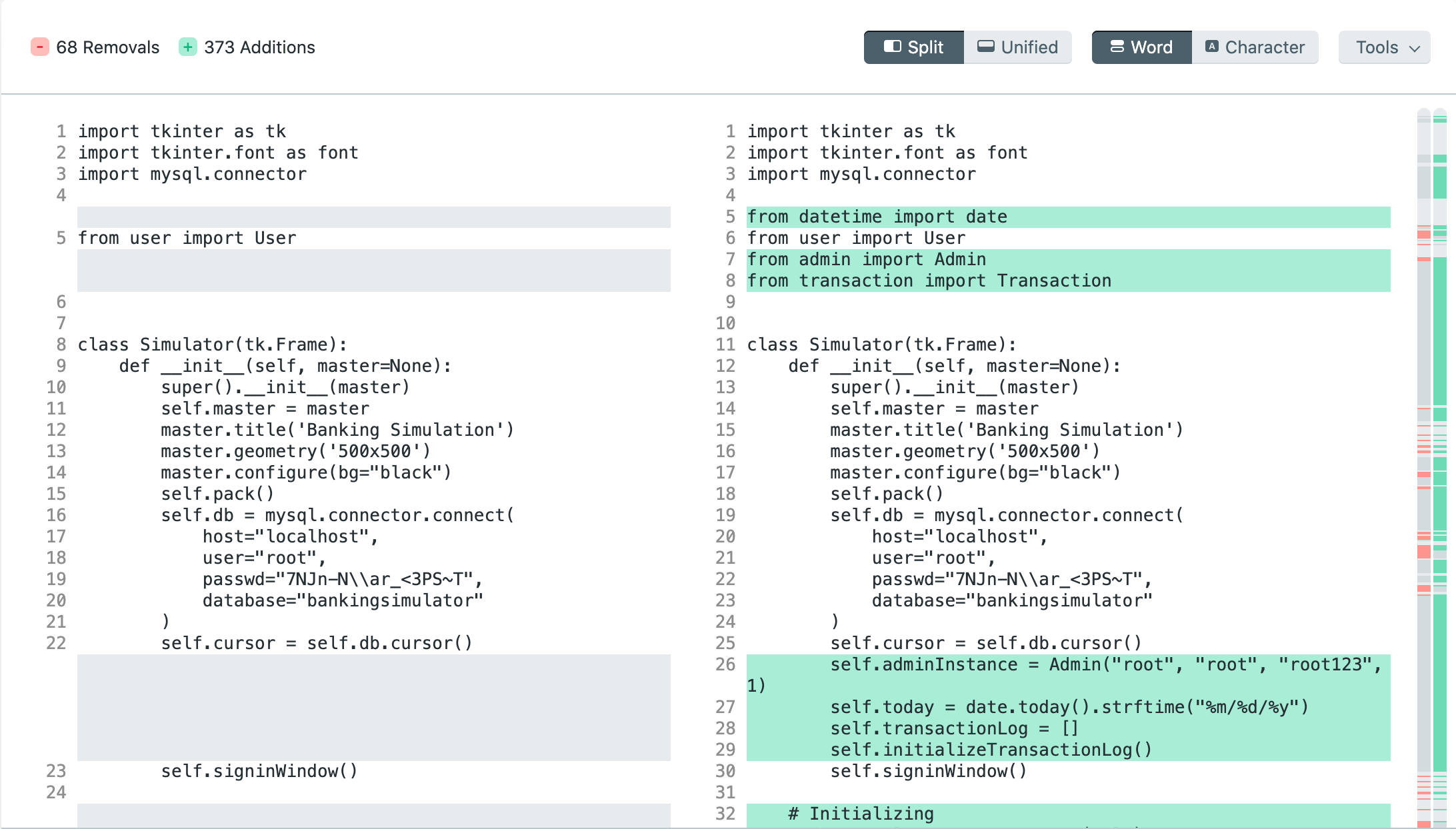


User Home Window

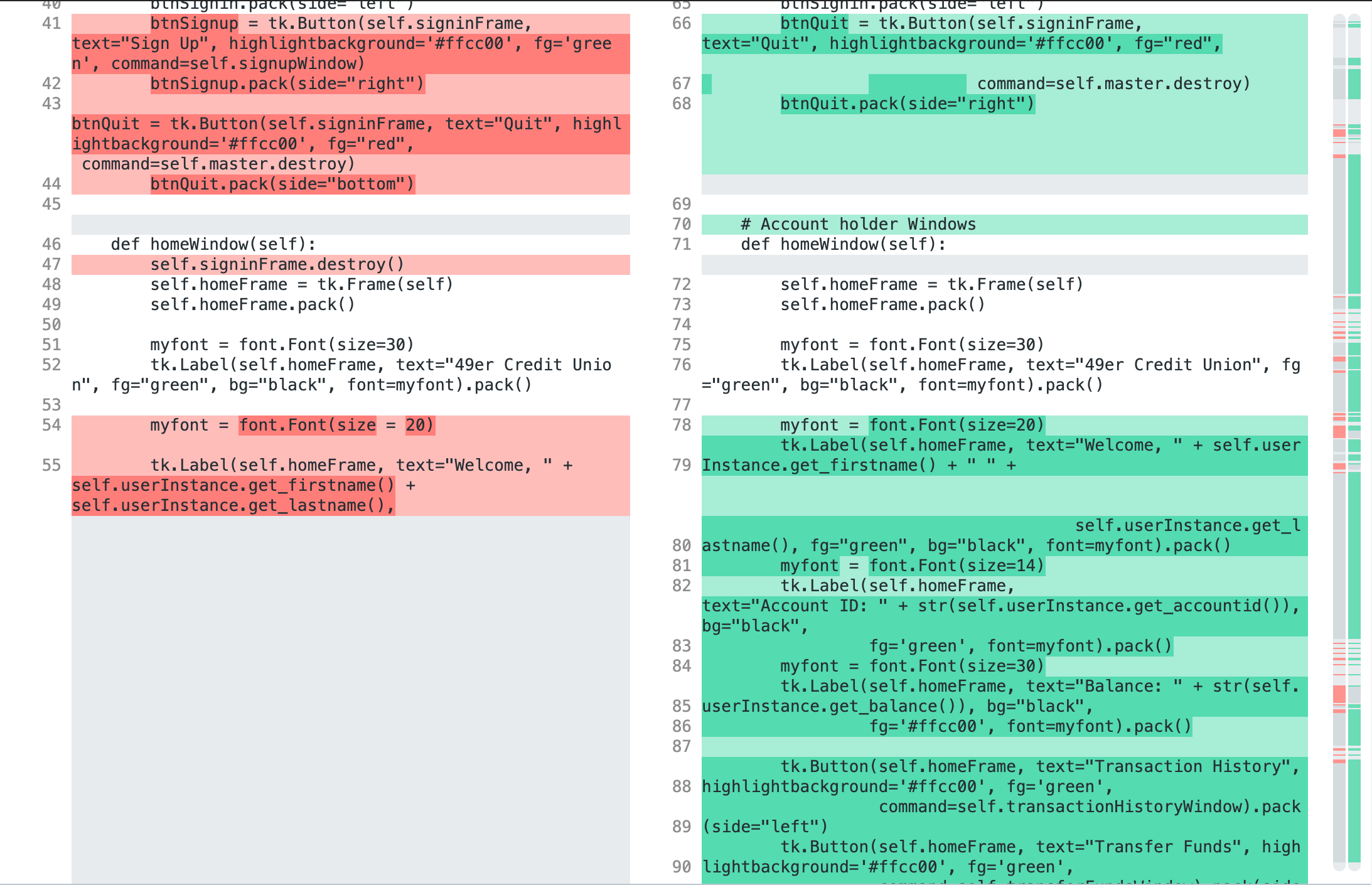
Compared to the old version, the enhanced version that the account ID on display as well as options to navigate to the transaction history window and the transfer funds window and a logout button.

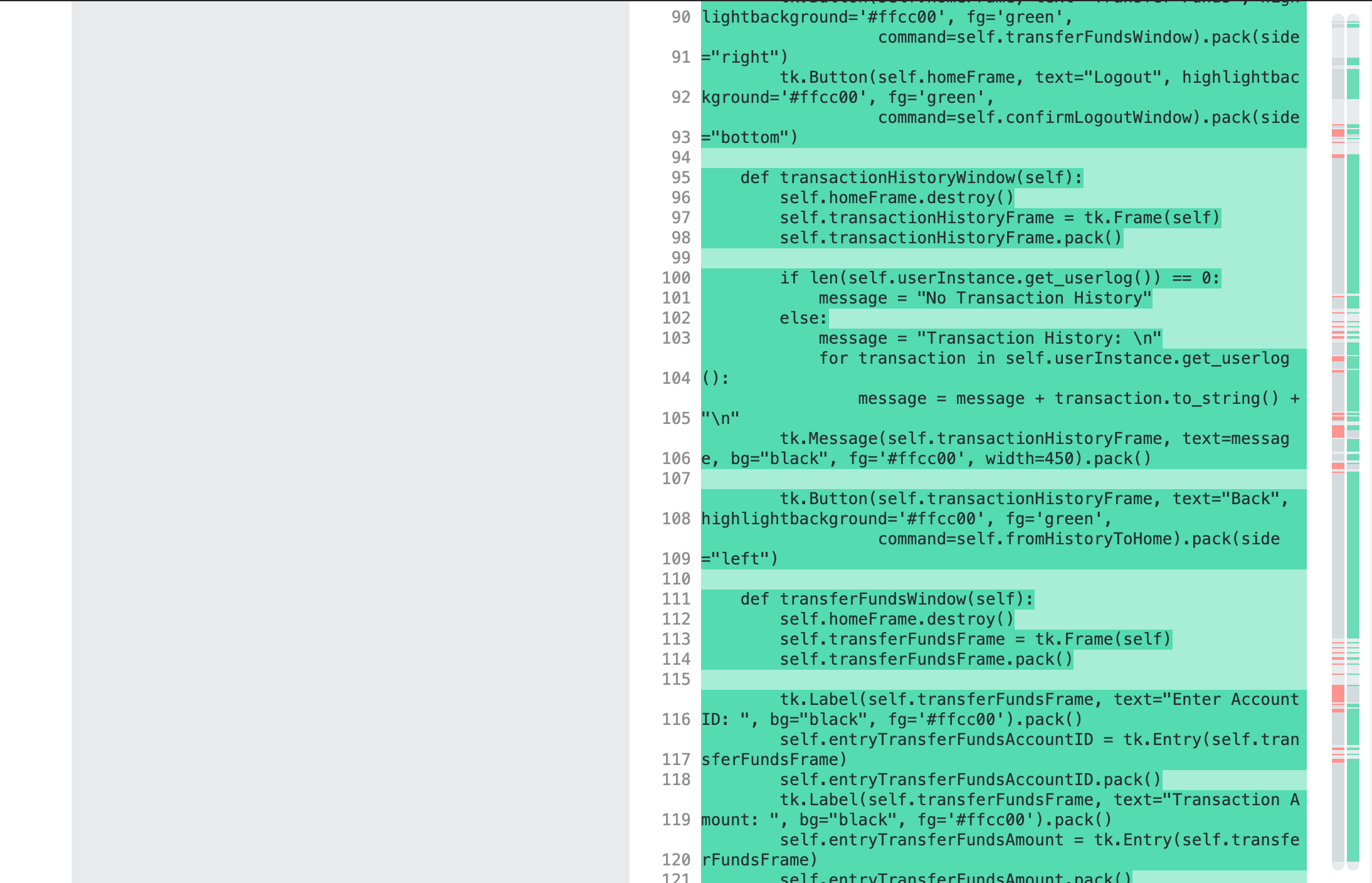


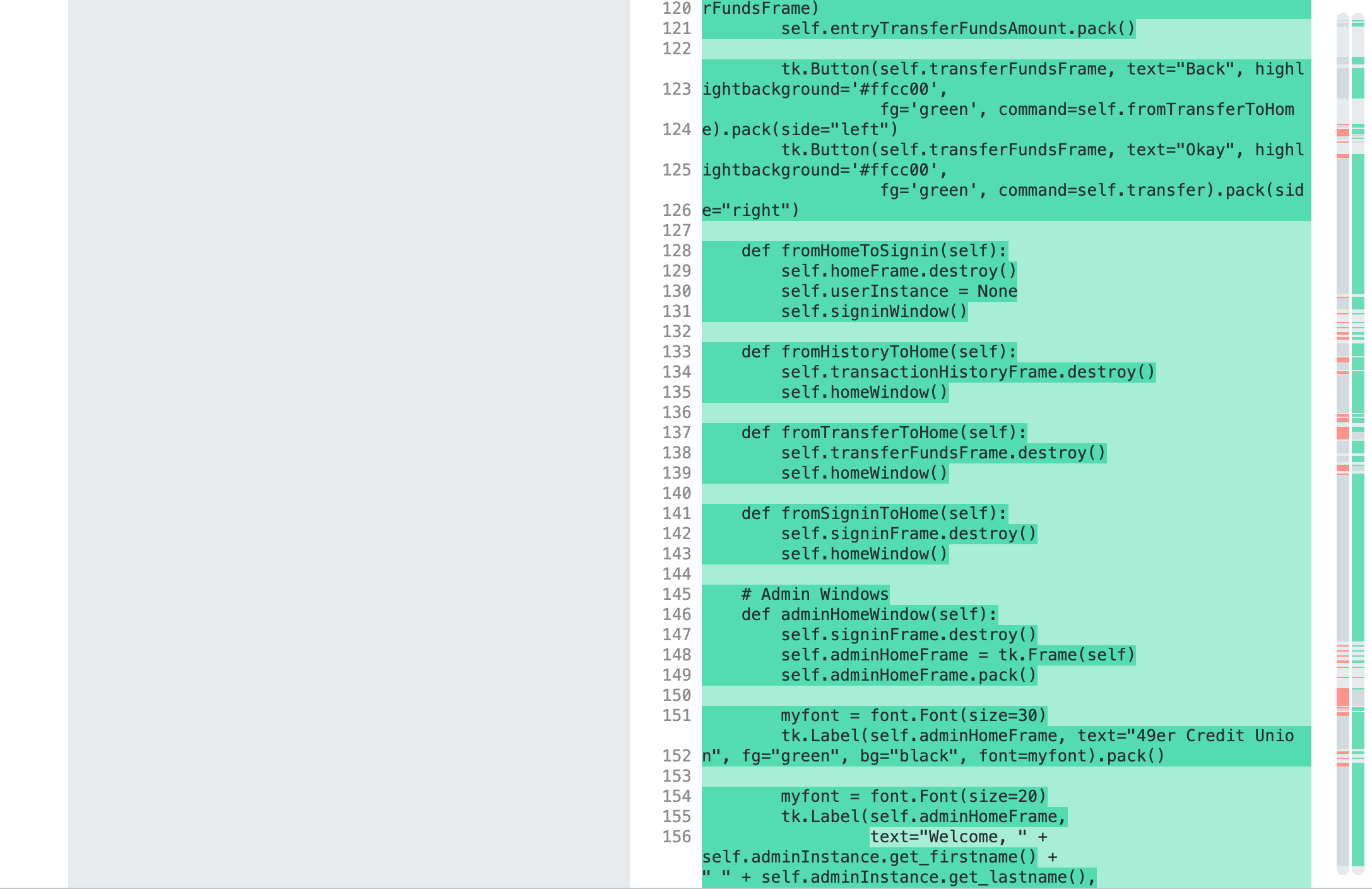
Simulator.py

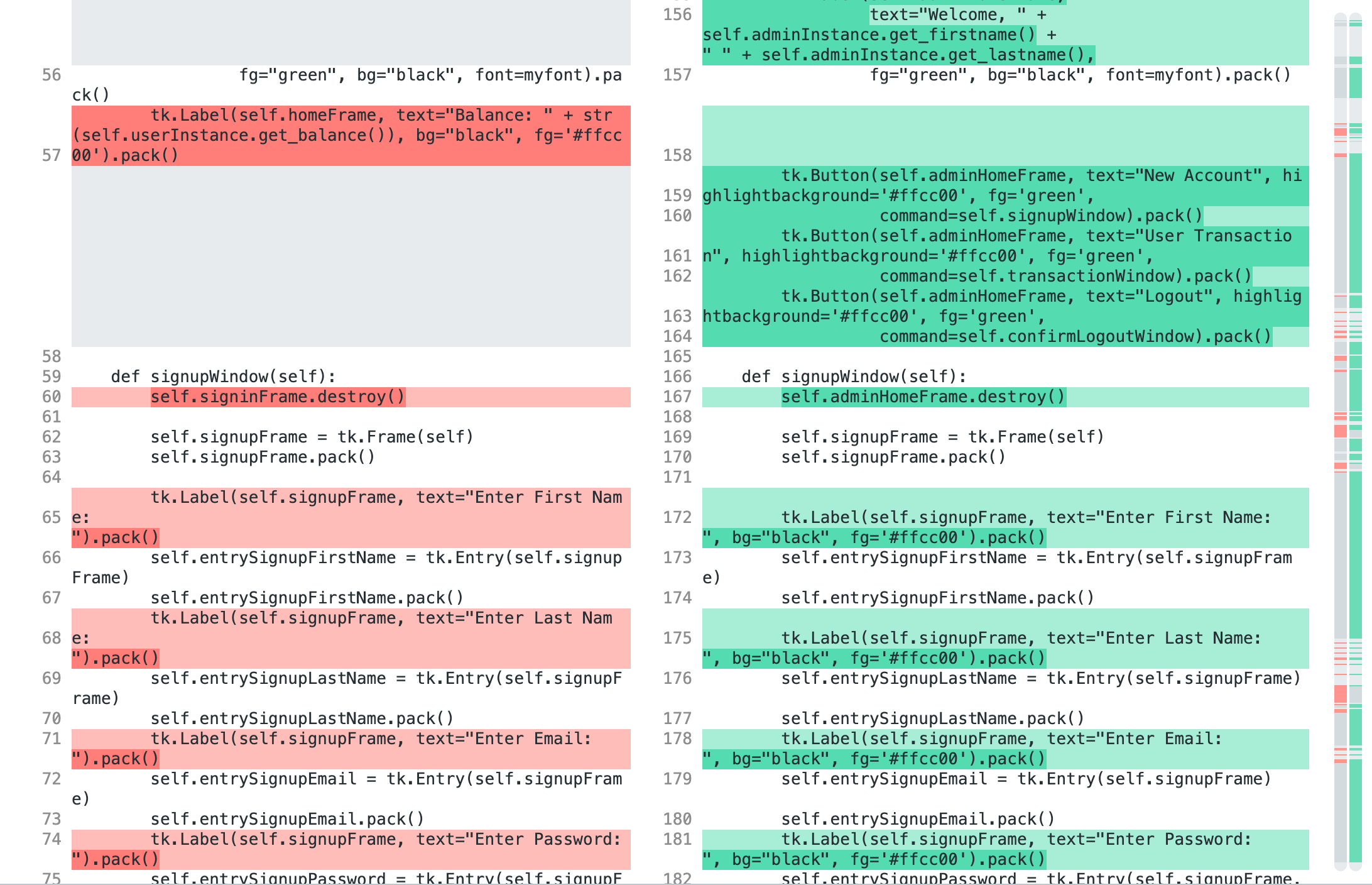


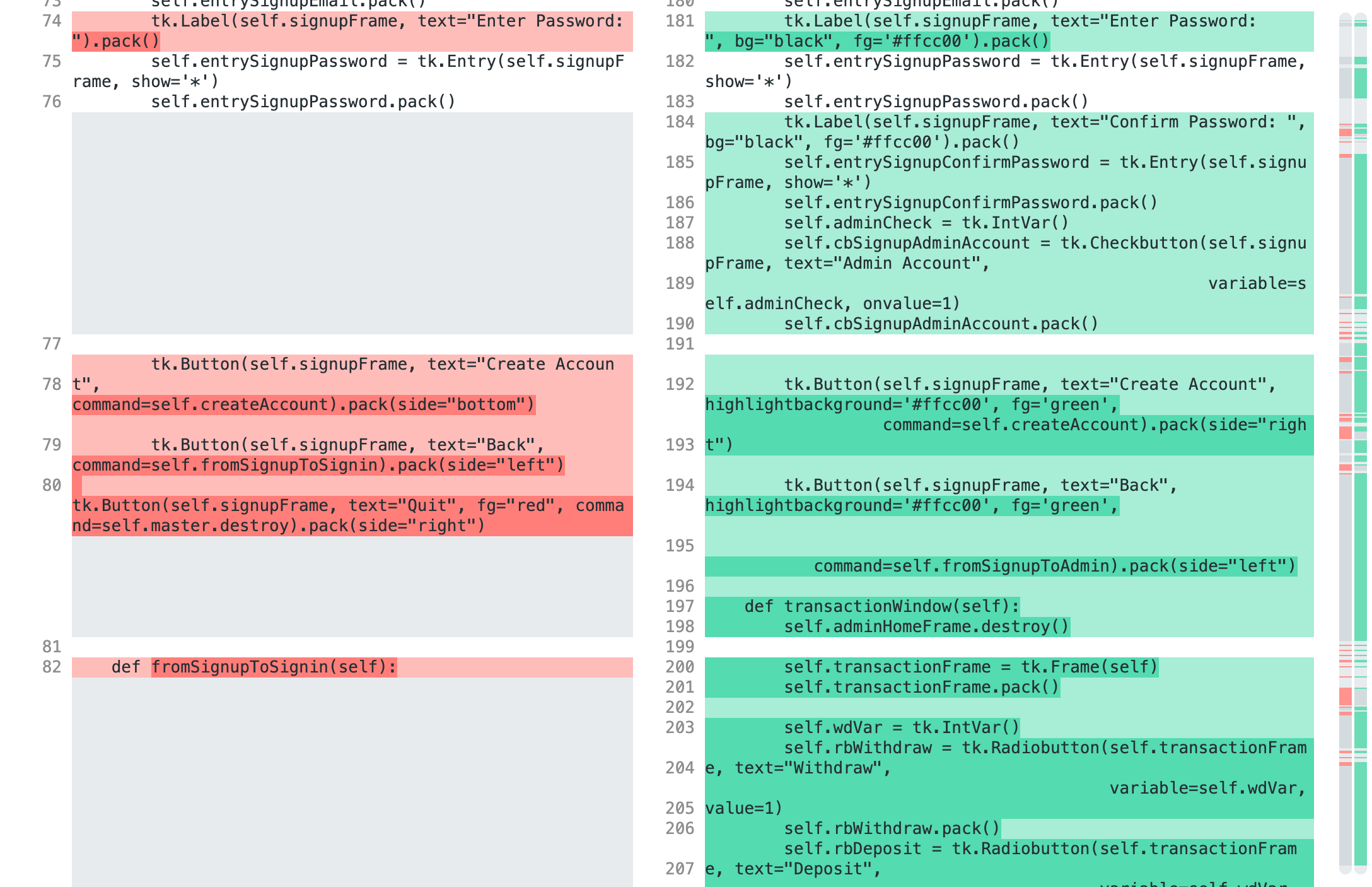


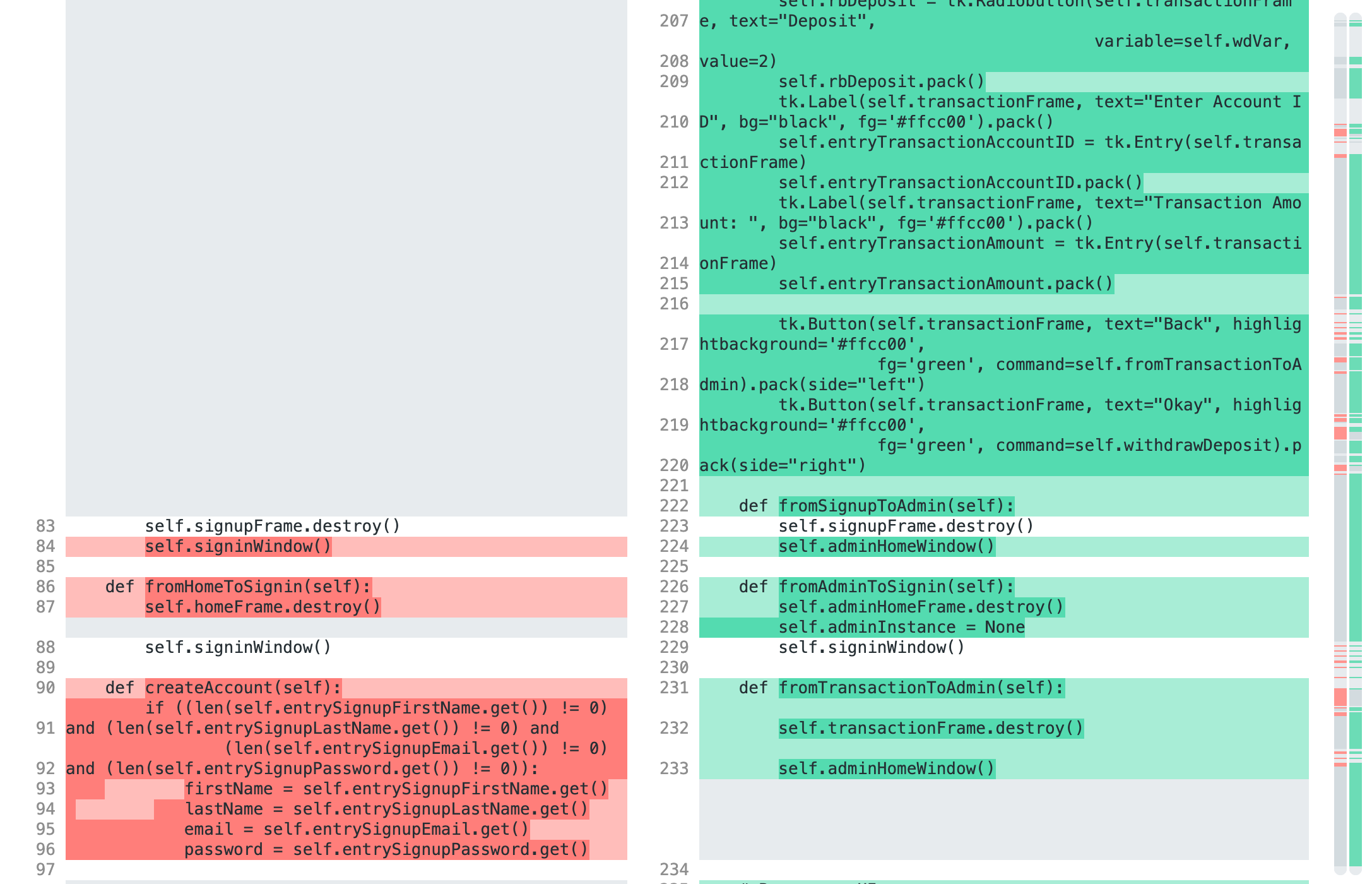




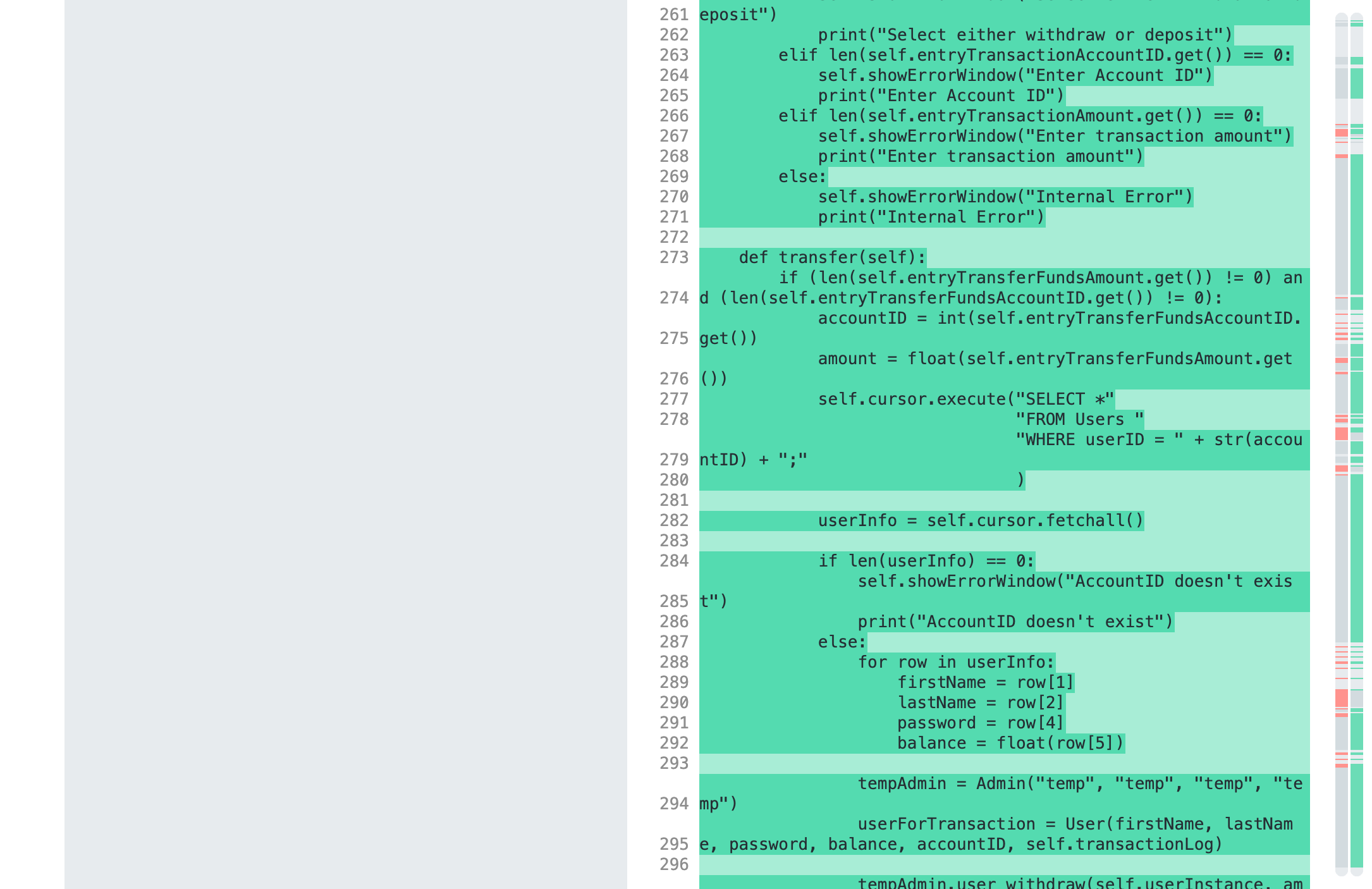


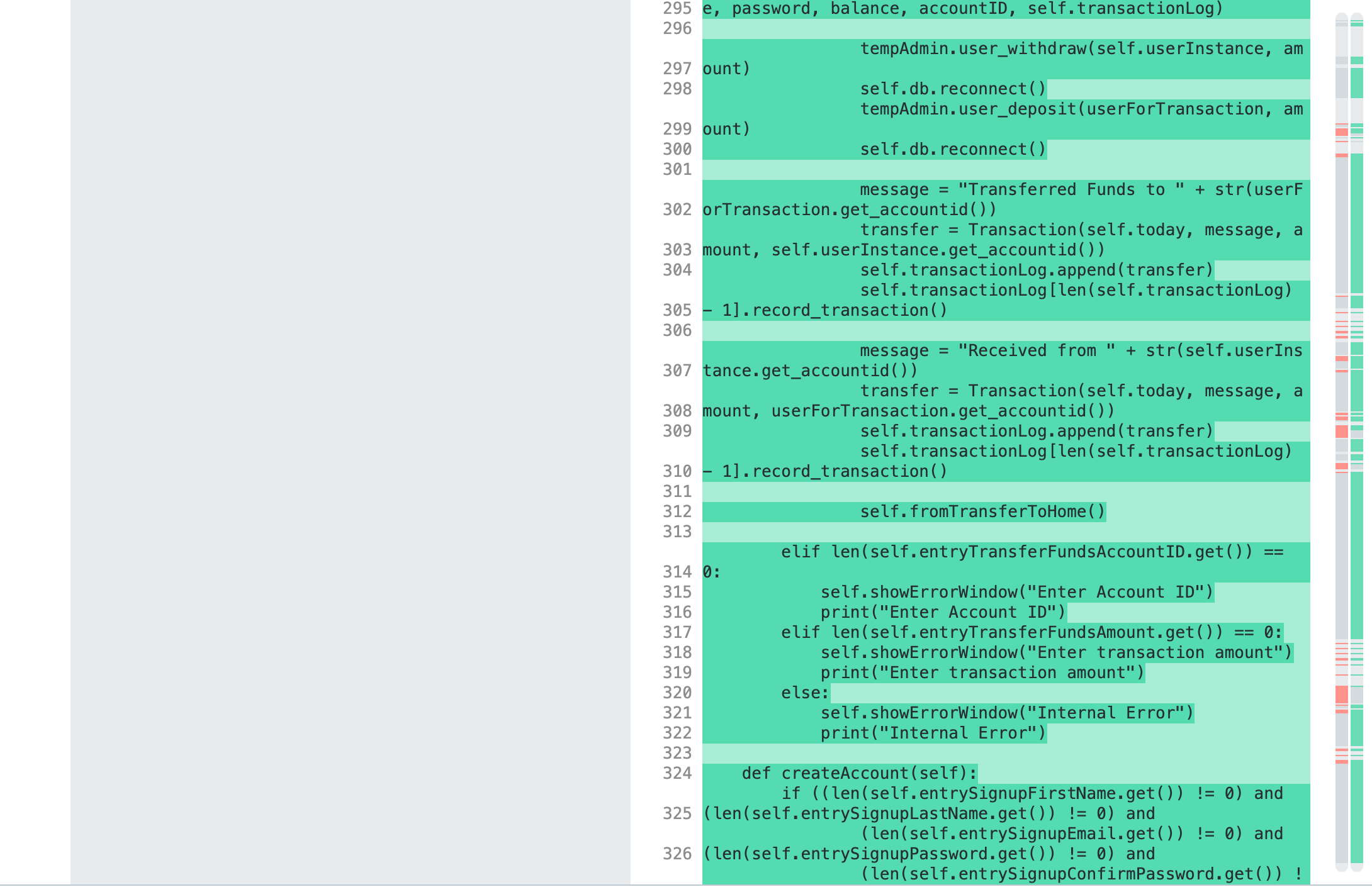


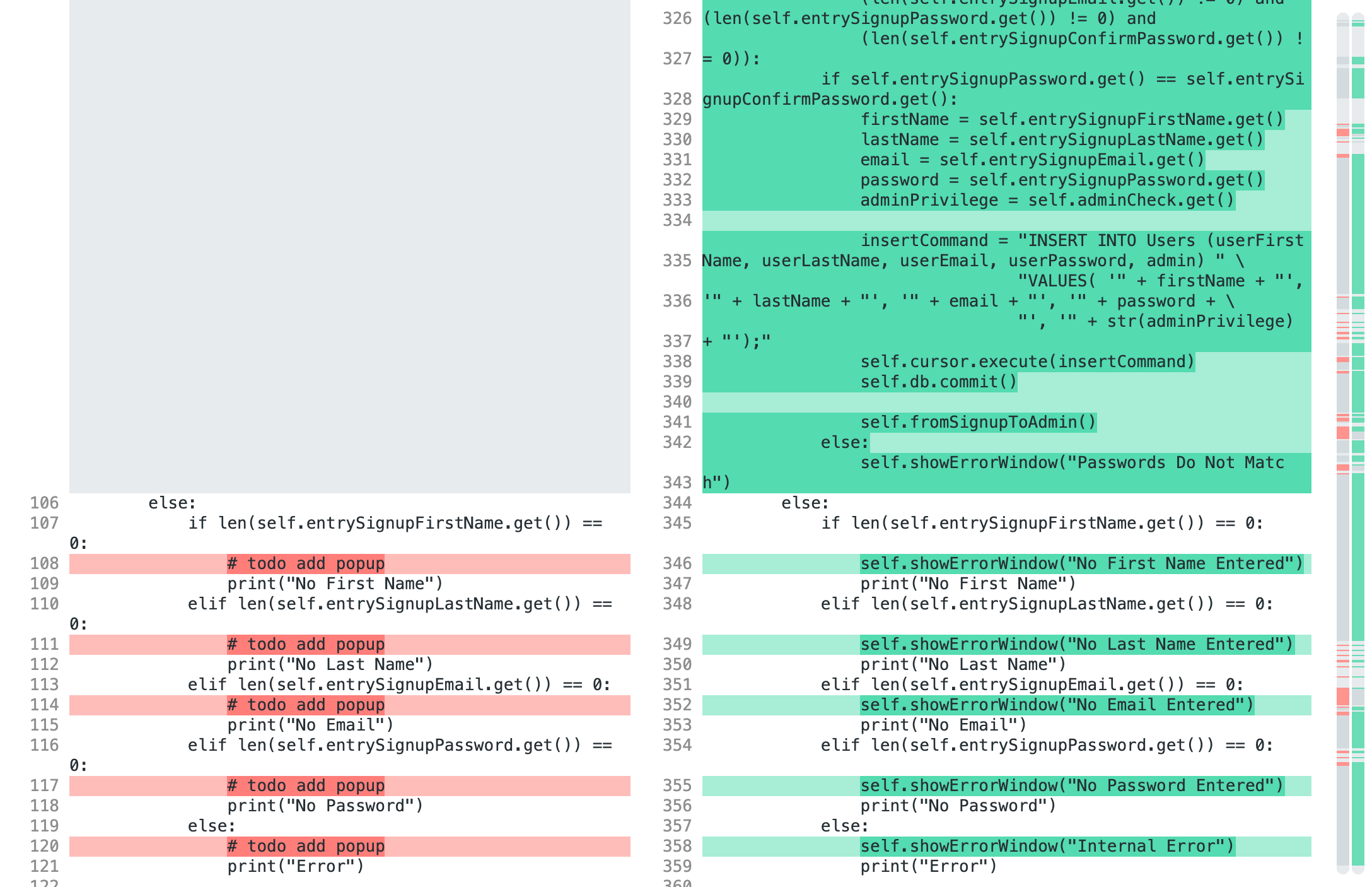




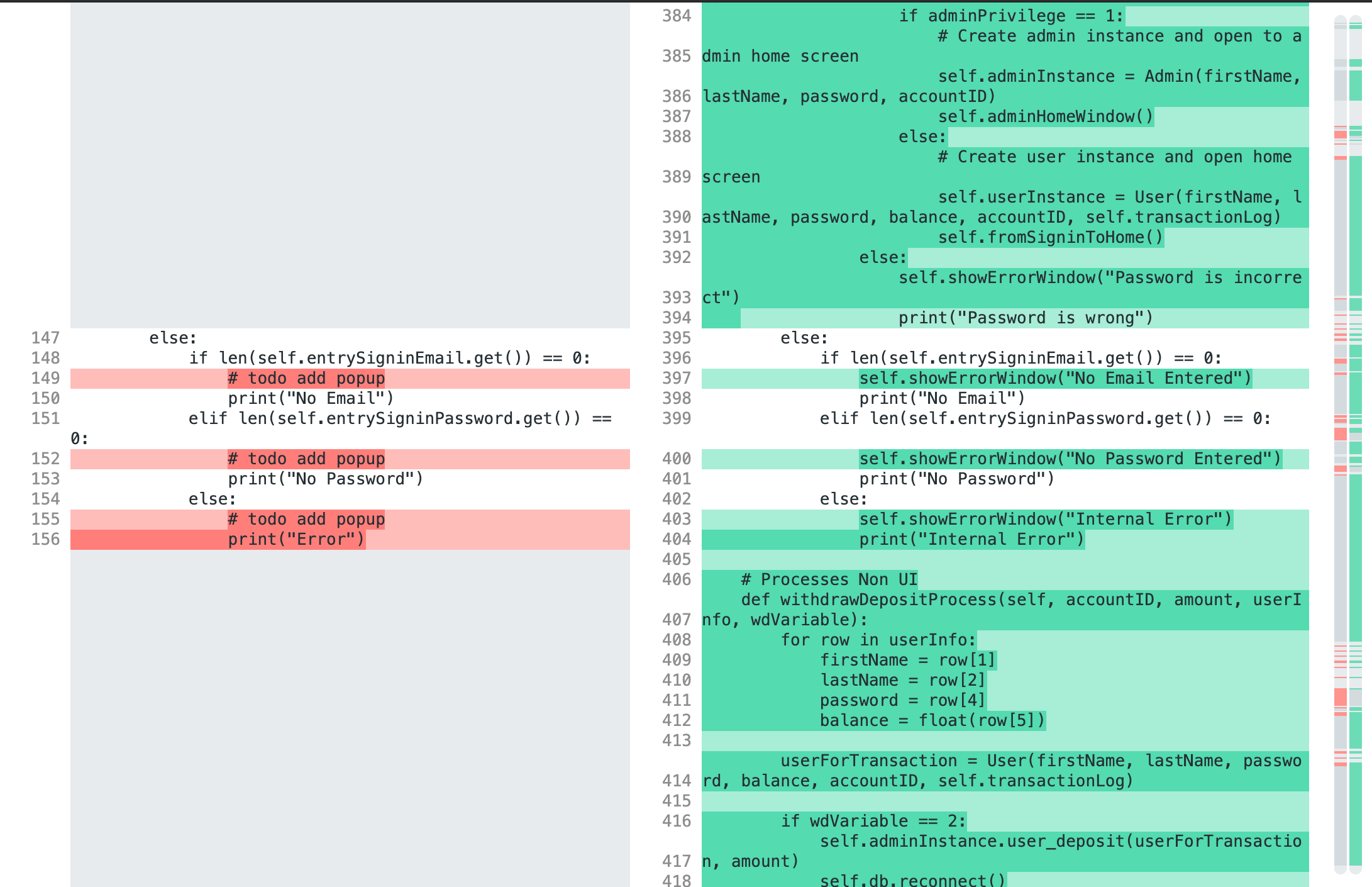


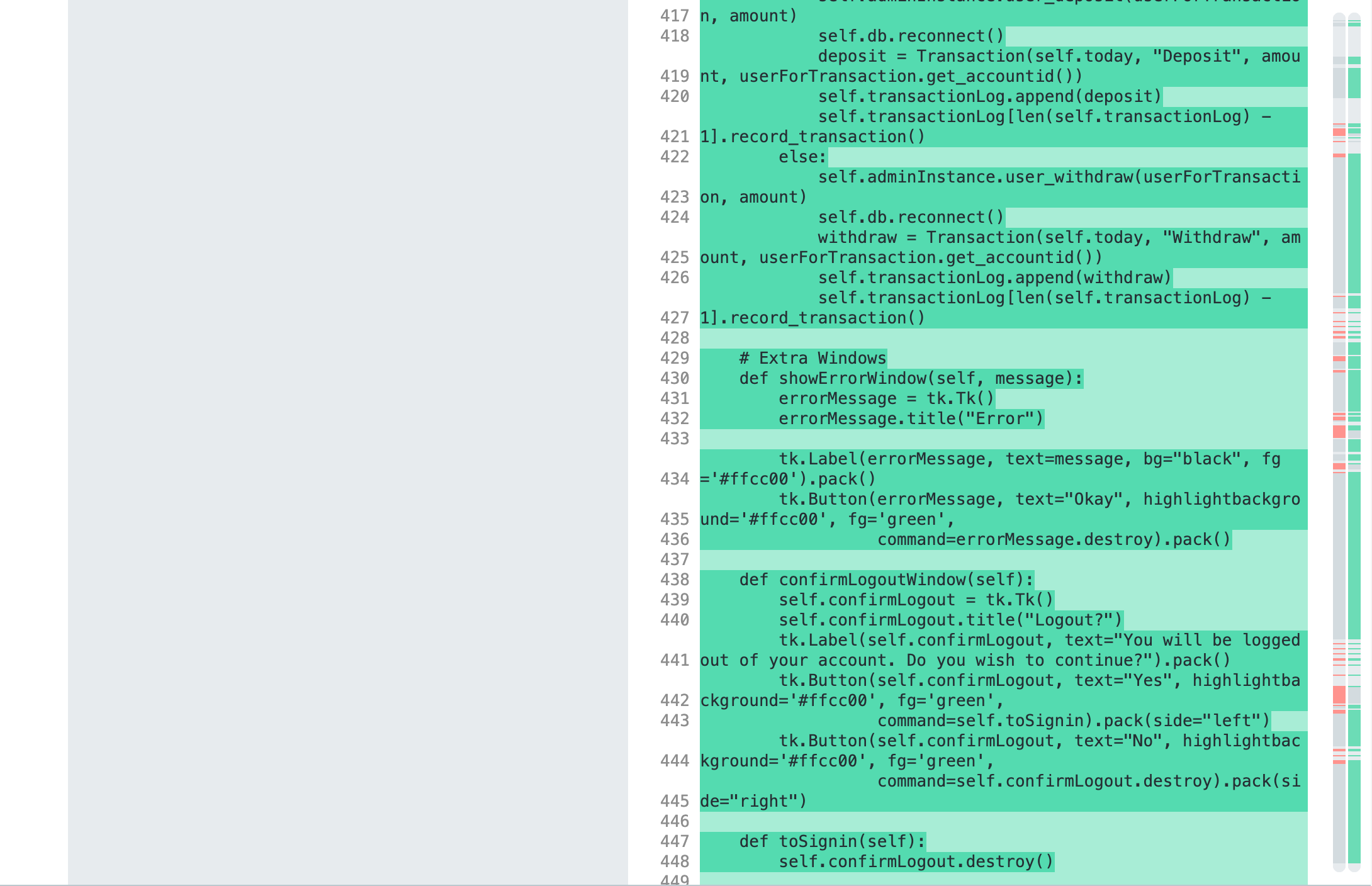


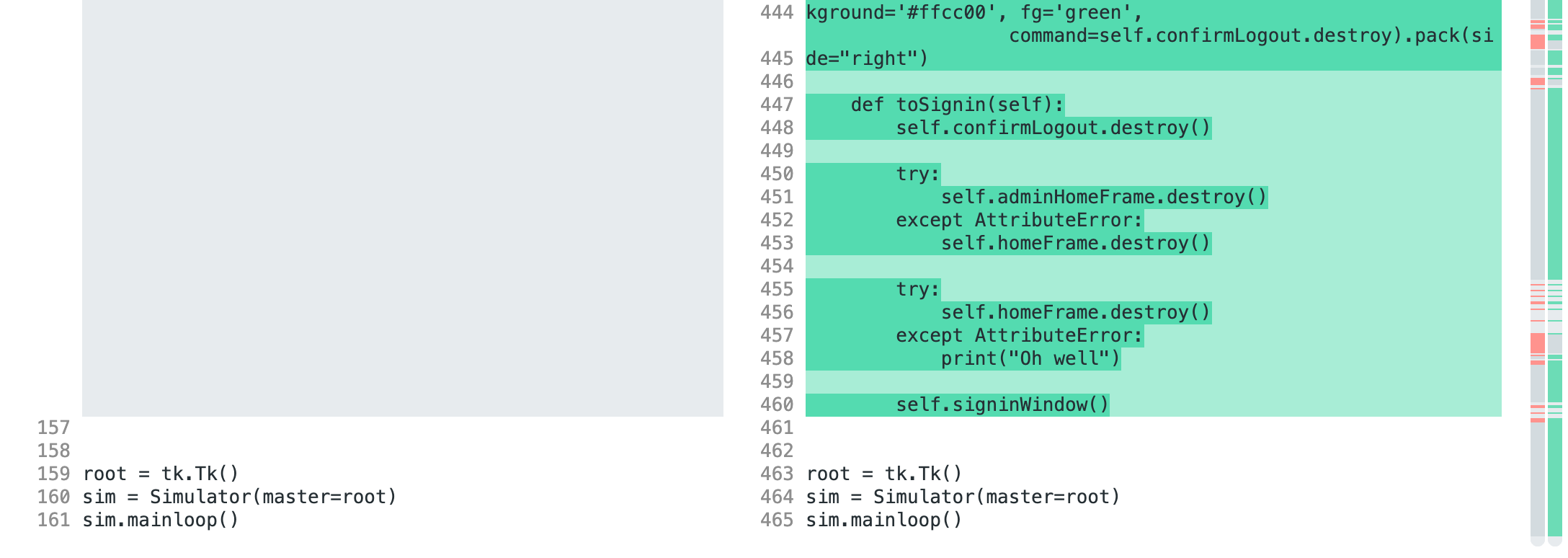




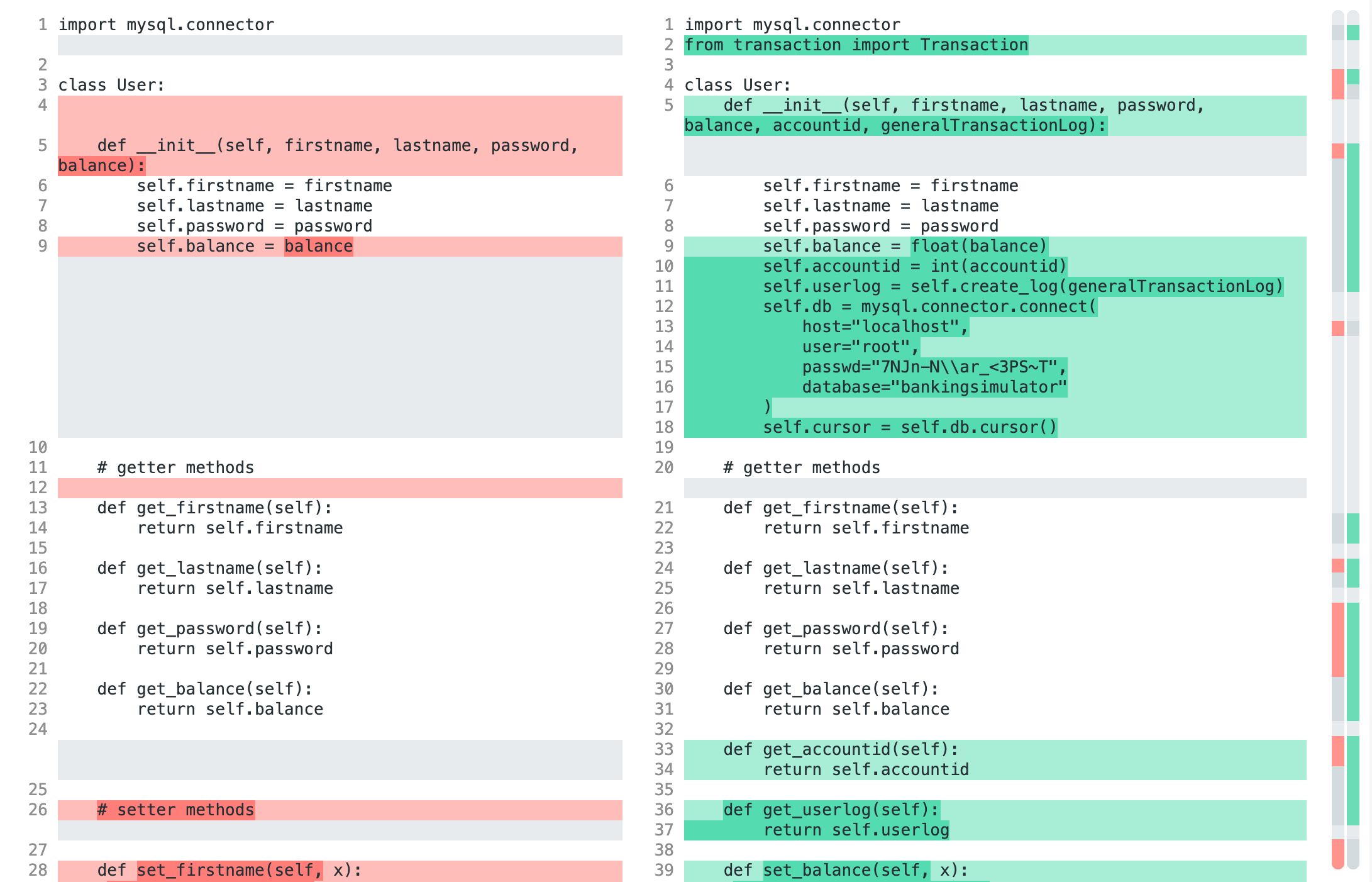


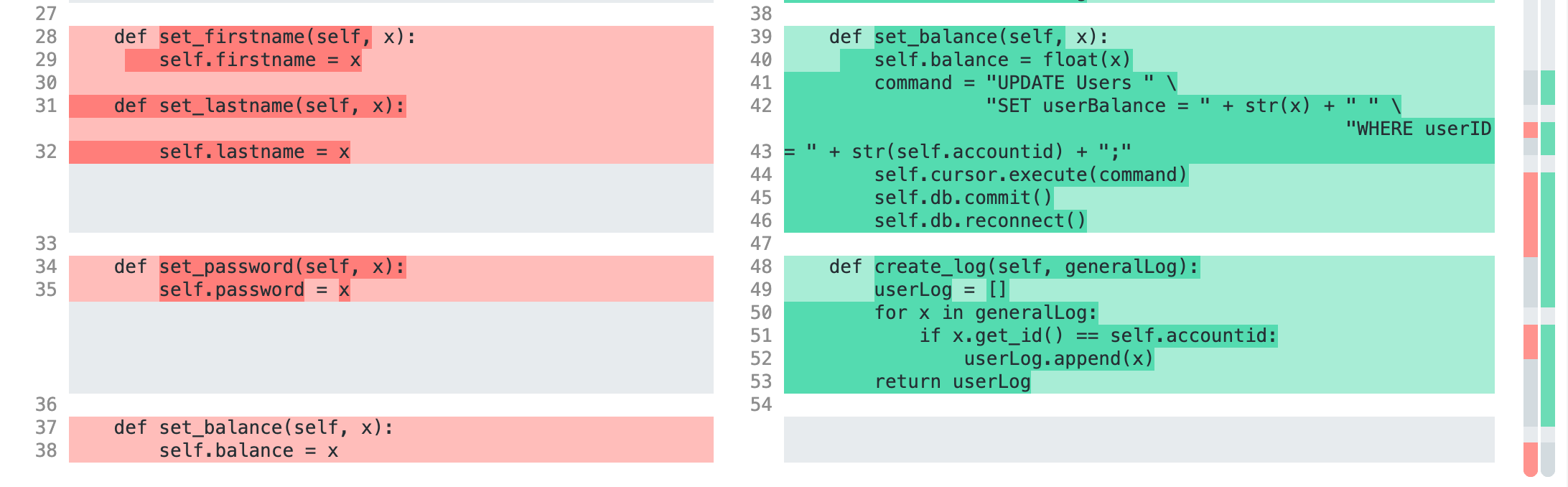




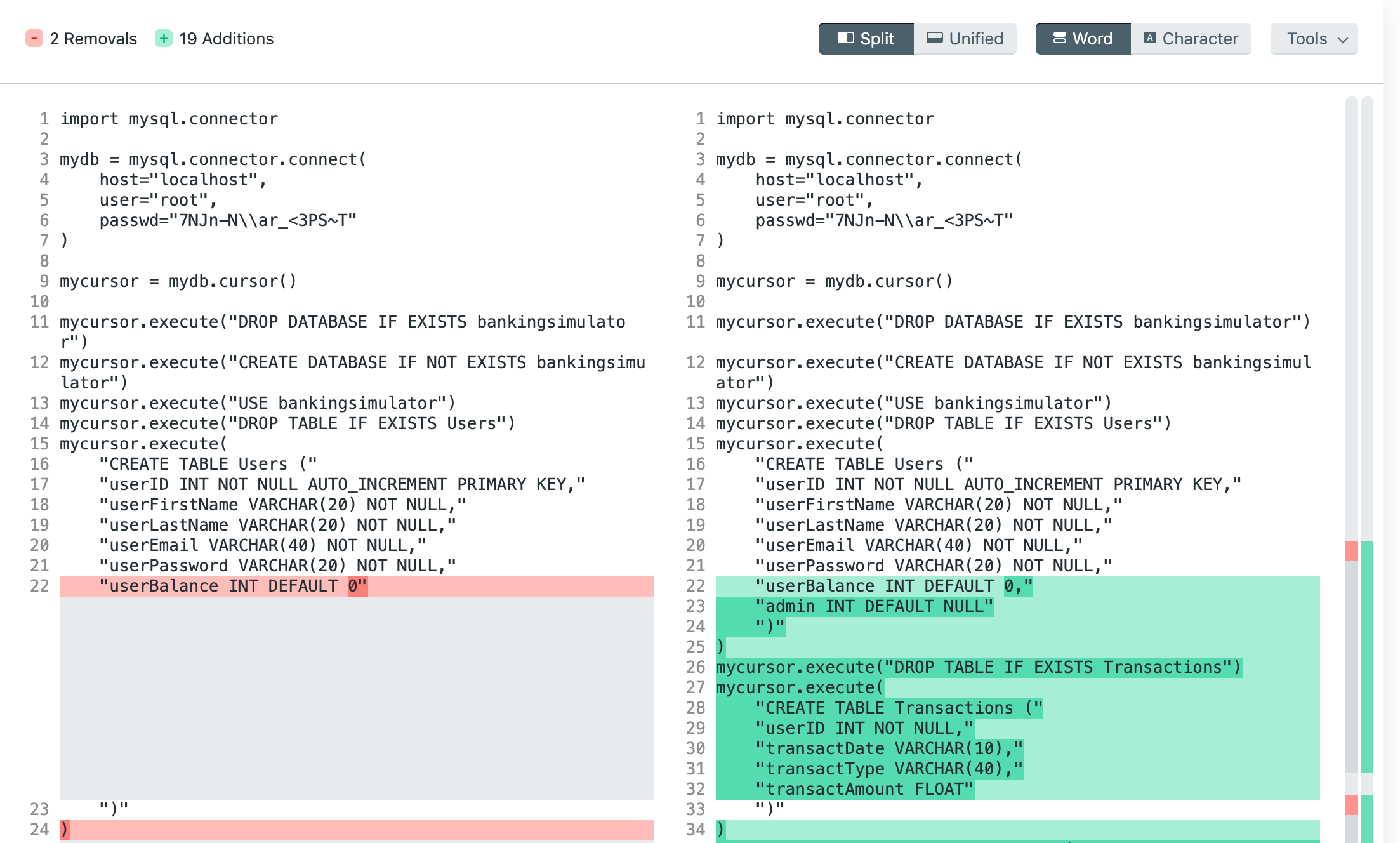


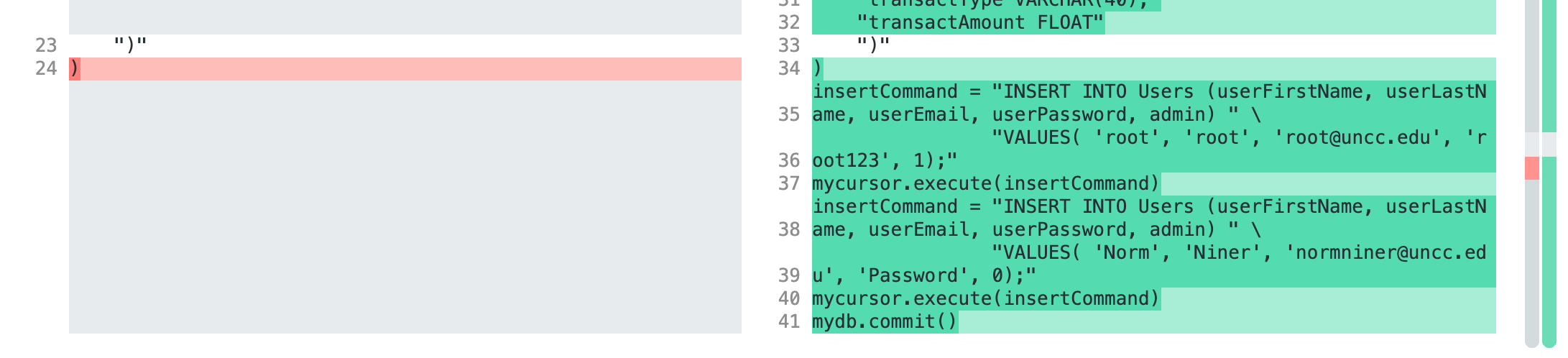
User.py



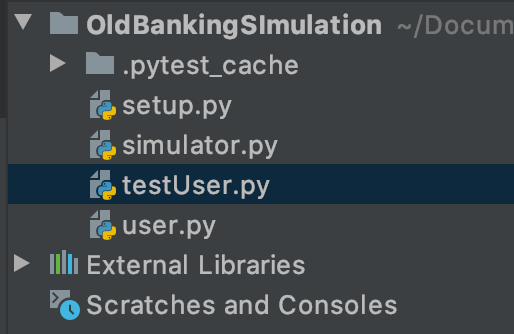
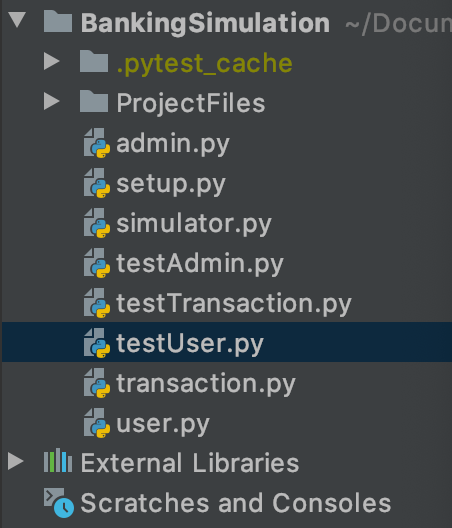


Setup.py





Additionally, the enhanced version has two more files that work with simulator.py. These files are admin.py and transaction.py.

## ….

# **5. Discussions and Conclusions**

*This project was definitely not without its challenges. In terms of the team, working with less people as the time went on proved to be more difficult than expected, as the workload per person raised each loss (not to mention, we had to reconfigure and reschedule the entire project each time). Python was a brand new (or relatively new) language to the team, so learning how to properly program and use the correct features was an obstacle that took time to conquer [5].*

*Through the roadblocks, the entire project was a learning experience in itself. Not only did we learn to adapt to change within the team and budget our time wisely to complete objectives on schedule, but we also gained personal insight to developing software. The structures within software development - scrum, team meetings, diagrams, github, coding, presenting, etc - allow developers to create a smoothly running team with clear pathways open to change along the way. Altering and improving our product as we built it helped facilitate team communication, brainstorming, and innovation.*

## ….

# A. Appendices

**A.1 Use Case 1**

**{Banking Simulation}**

**Author (s): \_Levi Carpenter, Amanda Poteate\_ Date: \_04/12/20\_**

**Version: \_\_\_1\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Sign Up** | | **USE CASE TYPE** |
| **USE CASE ID:** | **2** | | **Business Requirements: o** |
| **PRIORITY:** | **High** | | System Analysis: o |
| **SOURCE:** | **Admin** | | **System Design: þ** |
| **PRIMARY BUSINESS ACTOR** | **Admin** | | |
| **PRIMARY SYSTEM ACTOR** | **N/A** | | |
| **OTHER PARTICIPATING ACTORS:** | * **N/A** | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * **N/A** | | |
| **DESCRIPTION:** | **Allows the admin to create new user and admin accounts** | | |
| **PRE-CONDITION:** | **User is signed in to admin account** | | |
| **TRIGGER:** | **Create account button press** | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1: Admin clicks create account** | **Step 2: System displays sign up window** | |
|  | **Step 3: Admin enters account info** | **Step 4: System checks password confirmation, creates account, sends admin to home screen** | |
| **ALTERNATE COURSES:** | **Step 4a: System detects Passwords don’t match and prompts user to re-enter password**  **Step 4a1: Go to step 3** | | |
|  |  | | |
| **CONCLUSION:** | **Database of system users** | | |
| **POST-CONDITION:** | **New user or admin account is created** | | |
| **BUSINESS RULES** | * **Only admin can create accounts** | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * **One account can be created at a time** | | |
| **ASSUMPTIONS:** | * **Mysql server is running** * **Setup.py file has been run** * **User can sign in to admin account** | | |
| **OPEN ISSUES:** | **Would it be easier to understand if the user can withdraw and deposit?** | | |
|  |  |  |  |

**A.2 Use Case 2**

**{Banking Simulation}**

**Author (s): \_Levi Carpenter, Amanda Poteate\_ Date: \_04/14/20\_**

**Version: \_\_\_1\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Transfer Funds** | | **USE CASE TYPE** |
| **USE CASE ID:** | **4** | | **Business Requirements: o** |
| **PRIORITY:** | **Medium** | | System Analysis: o |
| **SOURCE:** | **User** | | **System Design: þ** |
| **PRIMARY BUSINESS ACTOR** | **N/A** | | |
| **PRIMARY SYSTEM ACTOR** | **User** | | |
| **OTHER PARTICIPATING ACTORS:** | * **Other User** | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * **N/A** | | |
| **DESCRIPTION:** | **Allows the user to transfer funds into another account** | | |
| **PRE-CONDITION:** | **At least two User accounts are set up**  **User is signed in** | | |
| **TRIGGER:** | **Transfer Funds Button press** | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1: User presses Transfer Funds button** | **Step 2: System displays transfer funds screen** | |
|  | **Step 3: User enters accountID and transaction amount** | **Step 4: System accesses user account, completes transaction, and sends user to home screen** | |
|  |  |  | |
| **ALTERNATE COURSES:** | **Step 4a: System detects an invalid accountID entered and prompts user to enter a valid accountID**  **Step 4a1: Go to step 3** | | |
|  | **Step 4b: System detects transfer exceeds user account balance and prompts user to enter a lower transfer amount**  **Step 4b1: Go to step 3** | | |
|  |  | | |
| **CONCLUSION:** | **Log of transactions** | | |
| **POST-CONDITION:** | **Funds have been transferred from user account to other user account** | | |
| **BUSINESS RULES** | * **User can only tranfer up to their account balance** | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * **One transaction can be completed at a time** | | |
| **ASSUMPTIONS:** | * **Mysql server is running** * **Setup.py file has been run** * **User is logged in** | | |
| **OPEN ISSUES:** | **What is an appropriate home screen layout for Users and Admin?** | | |
|  |  |  |  |

**A.3 Use Case 3**

**{Banking Simulation}**

**Author (s): \_Levi Carpenter, Amanda Poteate\_ Date: \_04/14/20\_**

**Version: \_\_\_1\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Withdraw/Deposit** | | **USE CASE TYPE** |
| **USE CASE ID:** | **3** | | **Business Requirements: o** |
| **PRIORITY:** | **Medium** | | System Analysis: o |
| **SOURCE:** | **Admin** | | **System Design: þ** |
| **PRIMARY BUSINESS ACTOR** | **Admin** | | |
| **PRIMARY SYSTEM ACTOR** | **N/A** | | |
| **OTHER PARTICIPATING ACTORS:** | * **User** | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * **N/A** | | |
| **DESCRIPTION:** | **Allows the admin to withdraw or deposit funds from user accounts** | | |
| **PRE-CONDITION:** | **At least one User and one Admin account is set up**  **Admin is signed in** | | |
| **TRIGGER:** | **Transaction Button press** | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1: Admin presses Transaction button** | **Step 2: System displays transaction screen** | |
|  | **Step 3: Admin enters accountID, transaction amount, and selects transaction type** | **Step 4: System accesses user account, completes transaction, and sends admin to home screen** | |
|  |  |  | |
| **ALTERNATE COURSES:** | **Step 4a: System detects an invalid accountID entered and prompts user to enter a valid accountID**  **Step 4a1: Go to step 3** | | |
|  | **Step 4b: System detects withdraw exceeds account balance and prompts user to enter a lower withdraw amount**  **Step 4b1: Go to step 3** | | |
|  |  | | |
| **CONCLUSION:** | **Log of transactions** | | |
| **POST-CONDITION:** | **Funds have been deposited to or withdrawn from user account** | | |
| **BUSINESS RULES** | * **Admin can only withdraw or deposit with consent from user** | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * **One transaction can be completed at a time** | | |
| **ASSUMPTIONS:** | * **Mysql server is running** * **Setup.py file has been run** * **Admin is logged in** | | |
| **OPEN ISSUES:** | **What is an appropriate home screen layout for Users and Admin?** | | |
|  |  |  |  |

**A.4 Use Case 4**

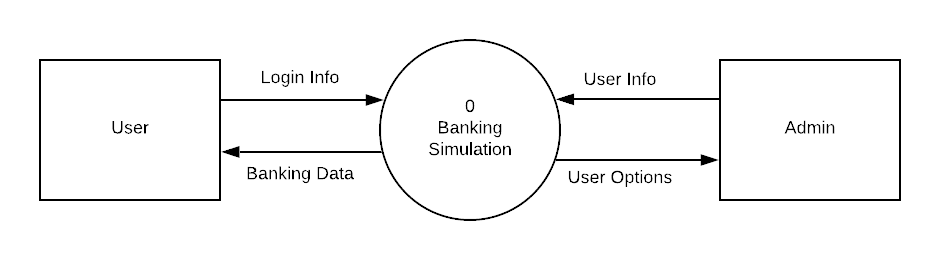
**{Banking Simulation}**

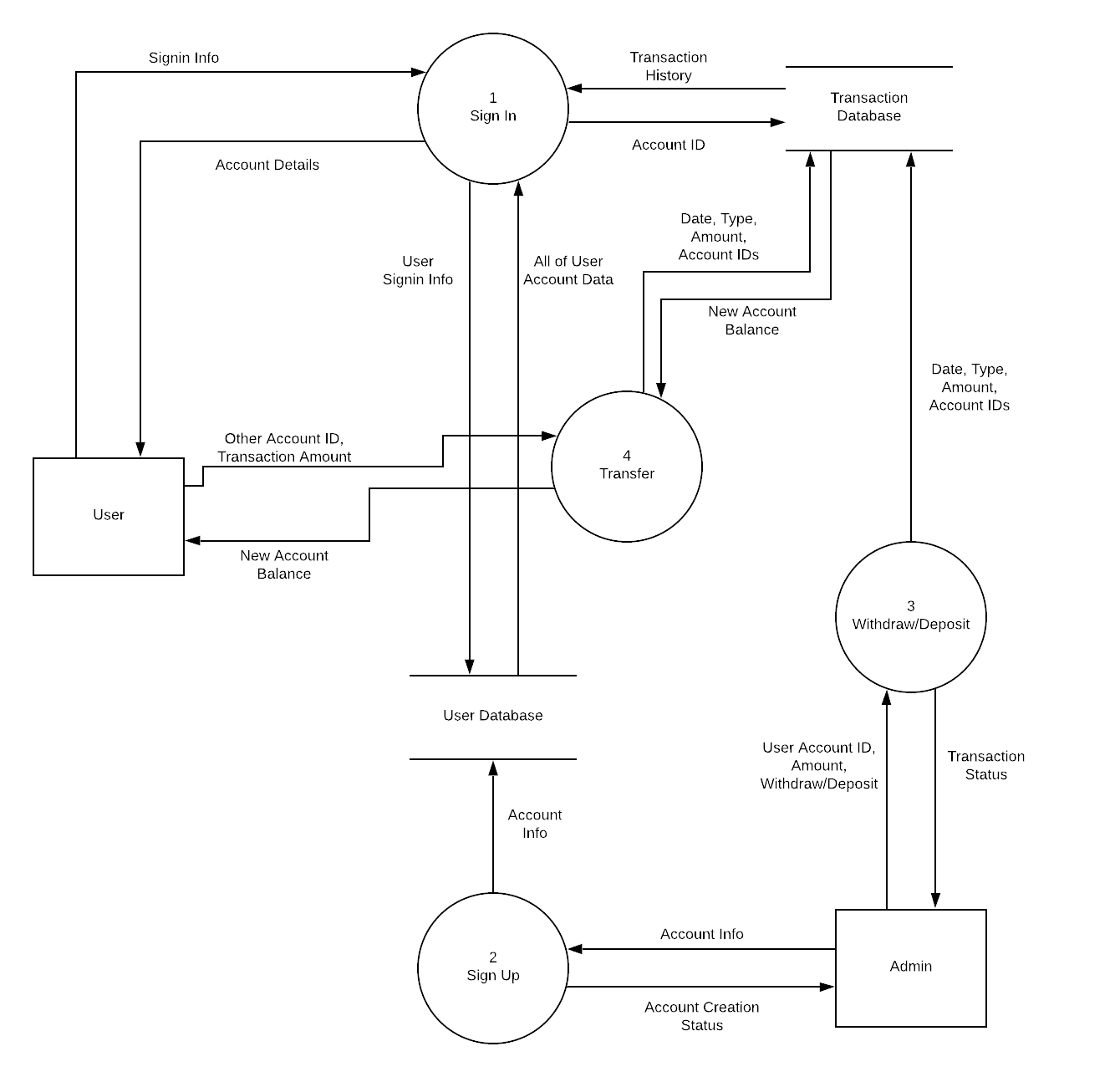
**Author (s): \_Levi Carpenter, Amanda Poteate\_ Date: \_04/10/20\_**

**Version: \_\_\_1\_\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Sign In** | | **USE CASE TYPE** |
| **USE CASE ID:** | **1** | | **Business Requirements: o** |
| **PRIORITY:** | **High** | | System Analysis: o |
| **SOURCE:** | **Admin** | | **System Design: þ** |
| **PRIMARY BUSINESS ACTOR** | **Admin** | | |
| **PRIMARY SYSTEM ACTOR** | **User** | | |
| **OTHER PARTICIPATING ACTORS:** | * **N/A** | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * **N/A** | | |
| **DESCRIPTION:** | **Allows the user or admin to sign in through verifying email and password** | | |
| **PRE-CONDITION:** | **System account is set up** | | |
| **TRIGGER:** | **Sign in Button press** | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1: User runs Application** | **Step 2: System displays Log in Screen** | |
|  | **Step 3: User enters Log in info** | **Step 4: System checks email and password with database and logs user in** | |
|  |  |  | |
| **ALTERNATE COURSES:** | **Step 4a: System displays incorrect Email or Password and prompts user to re-enter info**  **Step 4a1: Go to step 3** | | |
|  |  | | |
| **CONCLUSION:** | **Log of log in activity** | | |
| **POST-CONDITION:** | **User is logged into account** | | |
| **BUSINESS RULES** | * **User is logged into user home screen** * **Admin is logged into admin home screen** | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * **One account can be logged in at a time** | | |
| **ASSUMPTIONS:** | * **Mysql server is running** * **Setup.py file has been run** | | |
| **OPEN ISSUES:** | **What is an appropriate home screen layout for Users and Admin?** | | |
|  |  |  |  |

**A.5 Context Diagram**

**A.6 Data Flow Diagram**

****