

Human-Computer Interaction

CS 420/620

Project Part 3: Design

Team 01: PyBank

Brianna Blain-Castelli and Nikkolas Irwin

November 15, 2019

Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

Course Instructor: Dr. Sergiu Dascalu

Contents

1	Abstract	1
2	High Level Design	2
2.1	System-Level Structural Diagram	2
2.2	System-Level Behavioral Diagram	4
3	Static Interface Design	5
3.1	Sign-in Window	5
3.2	Sign-up Window Part II	6
3.3	Account Overview Window	7
3.4	Checking Account Window	8
3.5	Online Transfer Funds Window	9
3.6	Visualization Window	10
4	Alternative Designs	11
4.1	Alternative Sign-in Window	11
4.2	Alternative Visualization Window	12
5	Annotated References and Resources	13
5.1	The Design of Everyday Things	13
5.2	Designing the User Interface	13
5.3	Balsamiq Mockups 3	13
5.4	Lucid Chart	14
5.5	Qt	14
5.6	ZetCode	14
5.7	Material Design	14
6	Contributions of Team Members	15
6.1	Brianna Blain-Castelli	15
6.2	Nikkolas Irwin	15
	References	17

List of Figures

1	An architectural diagram describing the major software components present in PyBank.	3
2	An activity diagram describing the login and account creation processes in PyBank.	4
3	The Sign-in window for PyBank.	5
4	The Sign-up window part II for PyBank.	6
5	The account overview window shows basic information relating to all accounts owned by the user.	7
6	A detailed view of a specific account's checking account details.	8
7	A detailed view of a specific account performing an online transfer.	9
8	A detailed view of a specific account's visualizations.	10
9	A medium-fidelity prototype alternative for the sign-in window.	11
10	A medium-fidelity prototype alternative for the checking account visualizations window.	12

1 Abstract

PyBank is an interactive Python GUI application that simulates a desktop client for personal banking. The application will contain a number of common banking processes such as creating a new banking account, checking account information, depositing funds, withdrawing funds, and some analytical analysis related to an accounts spending over time. However, where most banking applications provide information only through text, this application will additionally provide a graphical method through which users may view their data. The design of this application utilizes the principles of human-computer interaction such as the Fundamental Principles of Interaction [1] and the Eight Golden Rules of Interface Design [2] so that the application provides an enjoyable experience to its end-users.

2 High Level Design

Included below in the following sub-sections (Section 2.1 and Section 2.2) are an architectural diagram and activity diagram which describe some high-level design features of PyBank.

2.1 System-Level Structural Diagram

The high-level components of PyBank can be listed broadly as the following:

- PyBank PyQt5 GUI
- Navigation Menu Component
- Login Component
- Sign-up Component
- Account Overview Component
- Checking Account Component
 - Show Graphs Component
 - Deposit Component
 - Withdraw Component
- Savings Account Component
 - Show Graphs Component
 - Deposit Component
 - Withdraw Component
- Credit Card Account Component
 - Show Graphs Component
 - Make A Payment Component
 - Request Credit Line Increase Component
- CSV File I/O Component

The relationship between the PyBank PyQt5 GUI application and its related components are further depicted in the architectural diagram shown below in Figure 1. This figure shows the components that are enclosed within the PyBank PyQt5 GUI as well as the CSV File I/O component which handles transaction processing for each PyBank customer. Unidirectional arrows convey an association between one component and another. For example, the Navigation Menu component is associated with six other components in the application. Bidirectional arrows convey both association and communication between two components in the application. Finally, dotted arrows are drawn to make it easier to visualize arrows that must cross the boundary of another arrow.

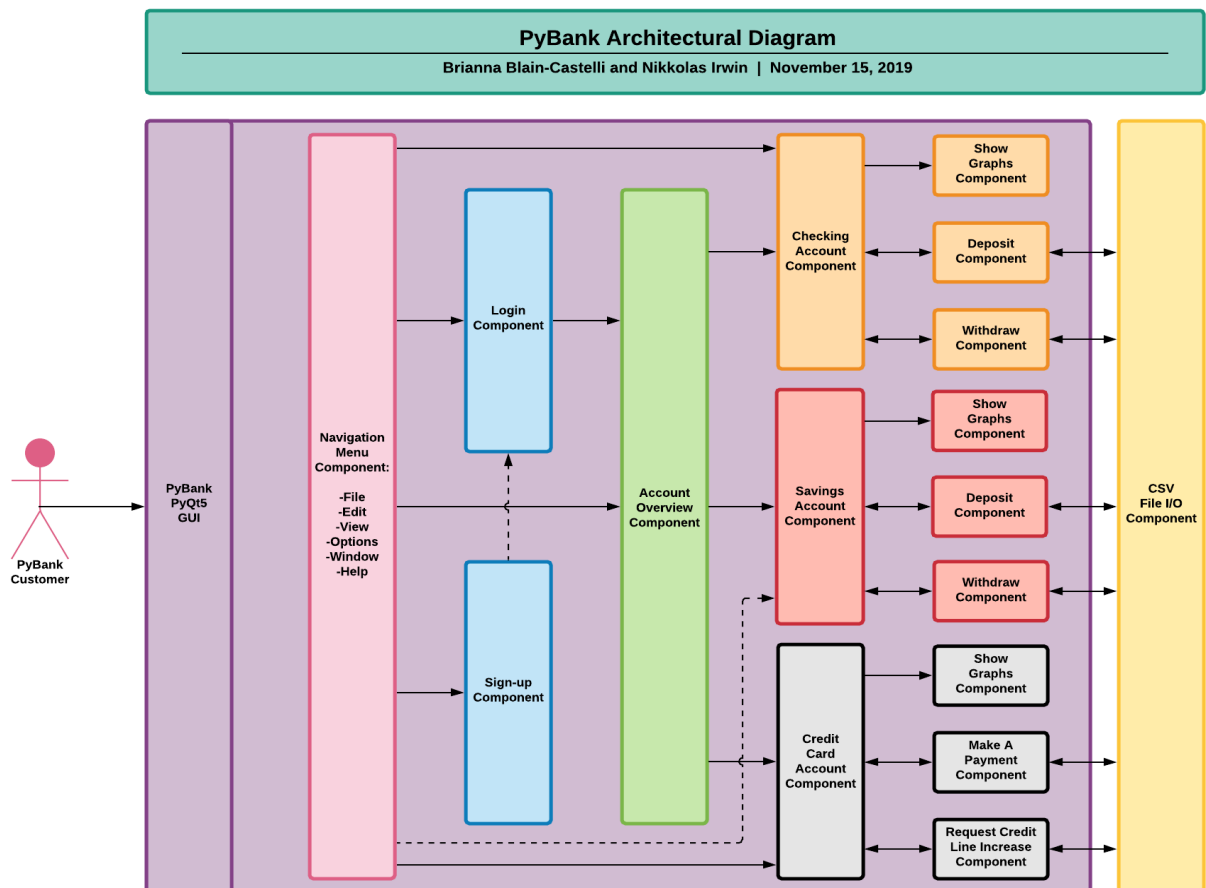


Figure 1: An architectural diagram describing the major software components present in PyBank.

2.2 System-Level Behavioral Diagram

The activity diagram shown below in Figure 2 depicts the sequence of actions that a PyBank customer would take to either login to their existing account or sign-up for a new account and then log into that account. The activity begins with a PyBank customer executing the desktop client and completes once the PyBank customer has logged into their account and is presented with the *Account Overview* window.

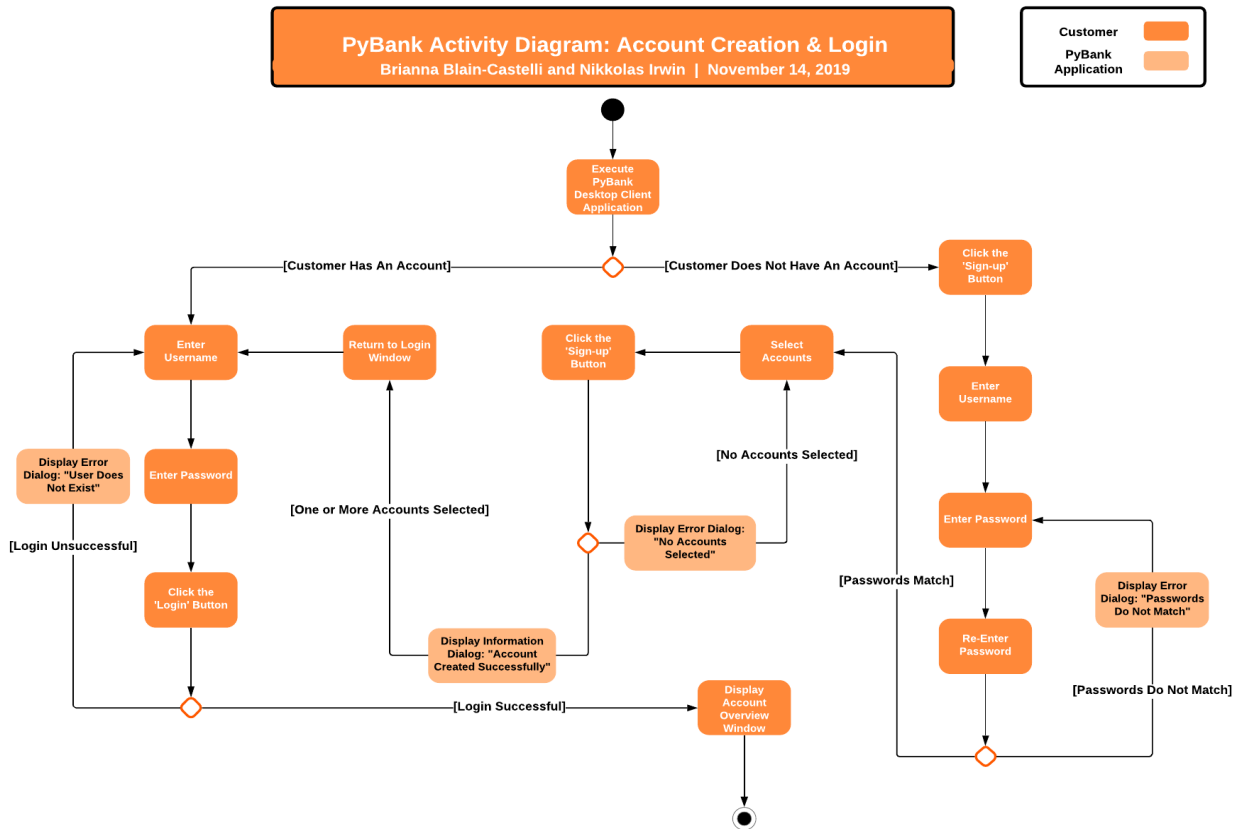


Figure 2: An activity diagram describing the login and account creation processes in PyBank.

3 Static Interface Design

Mock-ups of key views were created using Balsamiq [3] to illustrate the desired user interface associated with these windows prior to finalized implementation.

3.1 Sign-in Window

Figure 3 shows a mock-up of the login window for PyBank. The icon shown is a current placeholder until a finalized logo for the application can be developed. Password will display a masked version of the entered password for user privacy; however, when holding the eye shaped icon down, the password will show for the user. There is a link to sign-up available, button to sign up, and a button to cancel login and exit the application.

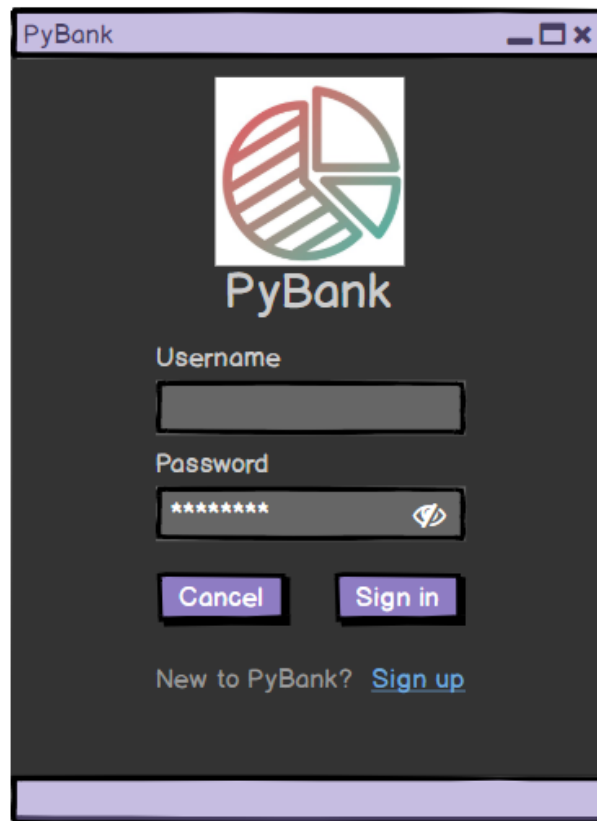
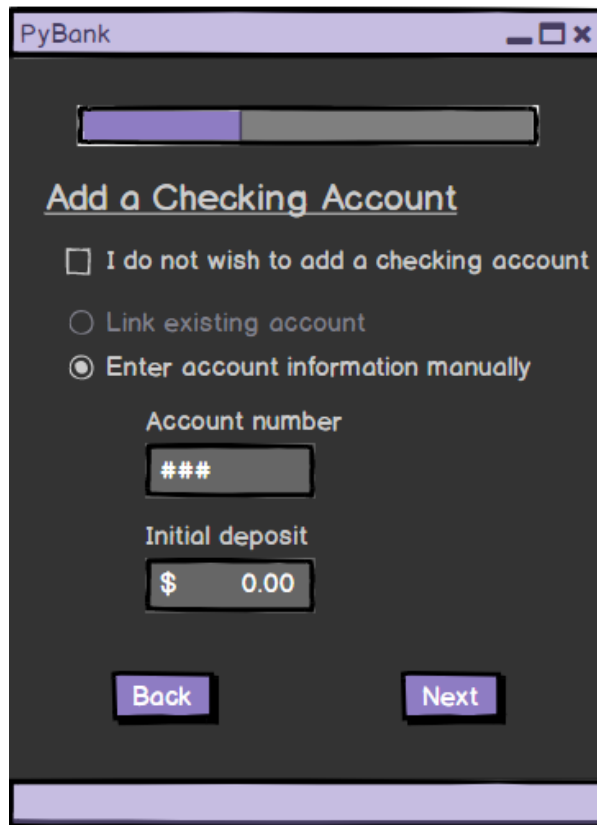


Figure 3: The Sign-in window for PyBank.

3.2 Sign-up Window Part II

Figure 4 is a mock-up of the second part of the sign-up window. Along the top is a progress bar, showing the user where in the sign-up process they are. Below the progress bar is a title, describing the purpose of the current step in the process. On this window, the user can utilize the checkbox that will grey out all options below besides next and back, keeping the user from entering information and allowing them to skip the window. Otherwise, the user may choose to link a bank account from a partnered bank, or enter account information manually for the user to track finances without linking all information.



The image shows a window titled "PyBank" with a standard macOS-style title bar (red, yellow, green buttons). Inside the window, at the top, is a progress bar with the first segment highlighted in blue. Below the progress bar is the title "Add a Checking Account" in a bold, black font. Underneath the title are three options: a checkbox labeled "I do not wish to add a checking account", a radio button labeled "Link existing account", and a selected radio button labeled "Enter account information manually". Below these options are two input fields: "Account number" with a placeholder "###" and "Initial deposit" with a placeholder "\$ 0.00". At the bottom of the window are two buttons: "Back" and "Next". The entire window has a light gray background.

Figure 4: The Sign-up window part II for PyBank.

3.3 Account Overview Window

Figure 5 is the account overview. This window provides a brief summary of account information for the user to access at a glance. This includes account balances, account number—which is entered by the user and serves as a form of account identification in the event the user has multiple of the same type of account—and an overview of the most recent transactions. Because this was the most common functionality utilized by stakeholders, it is the first window the user is taken to after signing in.

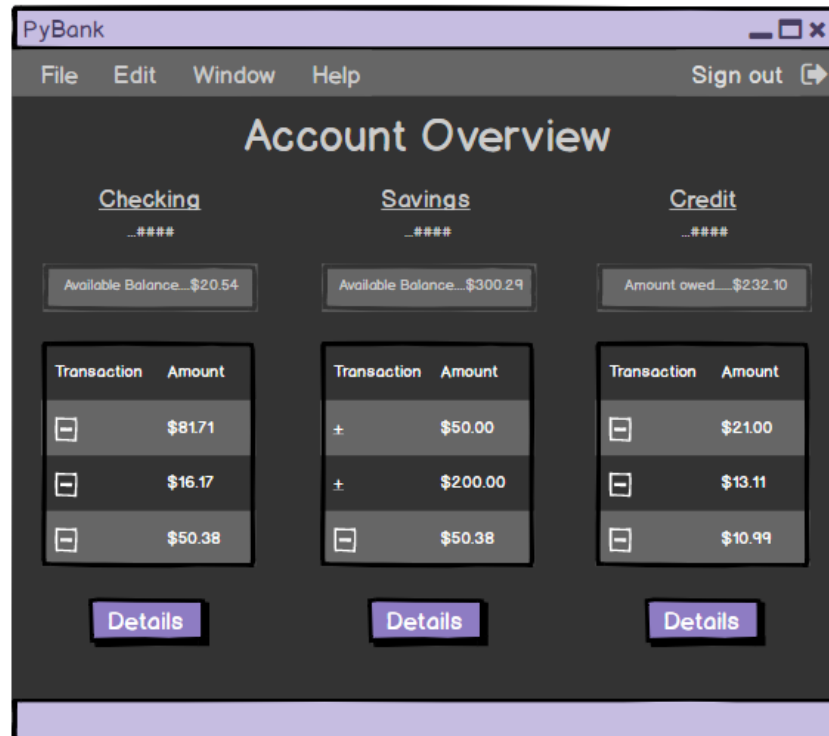


Figure 5: The account overview window shows basic information relating to all accounts owned by the user.

3.4 Checking Account Window

Figure 6 is the account details window. This window serves to show transaction details, upcoming bills, and a bill calendar. There is a search functionality as well that allows the users to find a specific transaction or bill. Along the top, this window provides options to access graph displays, send money, or transfer money.

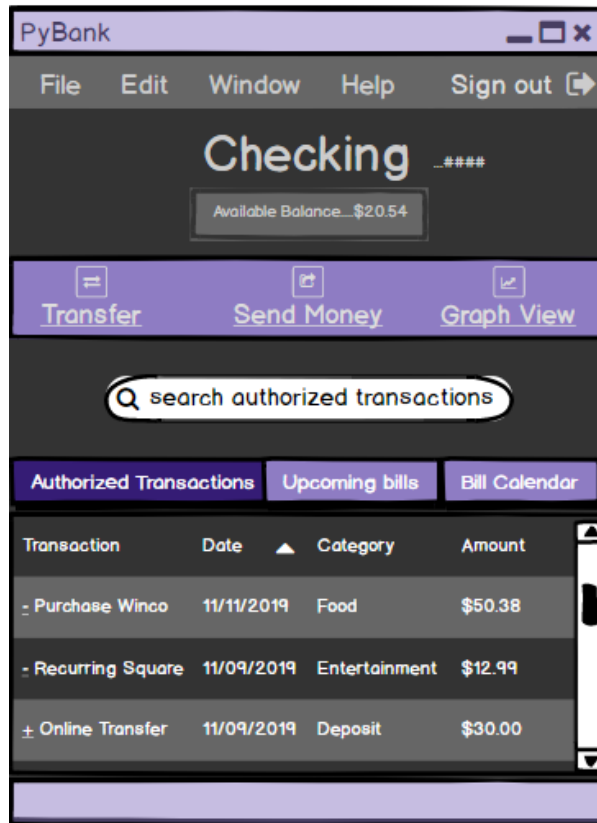
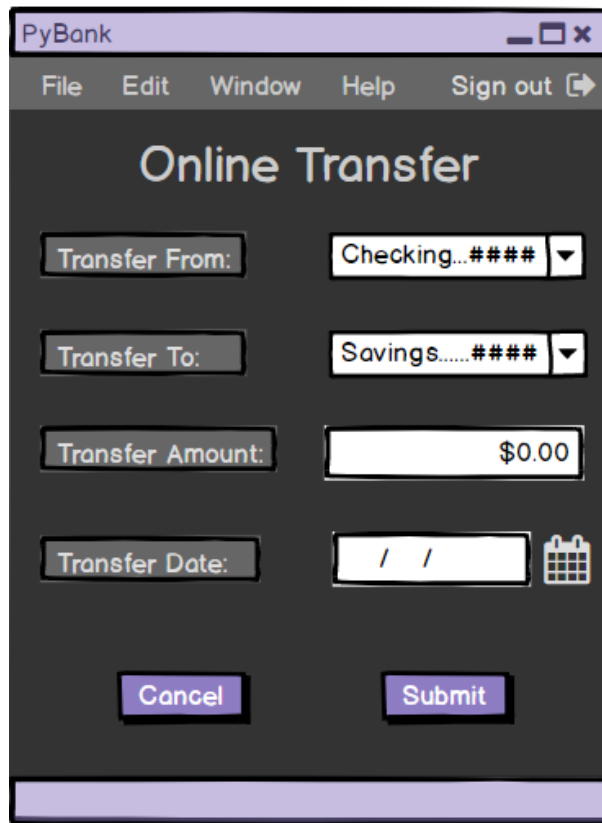


Figure 6: A detailed view of a specific account's checking account details.

3.5 Online Transfer Funds Window

The window for performing online transfers is shown in Figure 7. In this view, the user selects an account from the drop down menu to dictate where money is to be transferred from and where it is to be transferred to. The user then enters a value to transfer between accounts and the date on which the transfer is to take place.



The image shows a software window titled "PyBank" with a standard menu bar containing "File", "Edit", "Window", "Help", and "Sign out" with an icon. The main content area is titled "Online Transfer" and contains four input fields arranged in two columns. The left column has labels "Transfer From:", "Transfer To:", "Transfer Amount:", and "Transfer Date:". The right column has a dropdown menu showing "Checking...####", another dropdown showing "Savings.....####", a text field containing "\$0.00", and a date input field with slashes and a calendar icon. At the bottom are two buttons: "Cancel" and "Submit".

Figure 7: A detailed view of a specific account performing an online transfer.

3.6 Visualization Window

Graphical representations of finances and budgeting are shown in Figure 8. This view allows an alternate way to be able to see summaries of important information to the user. Along the top of the screen is the current graph being displayed. Below that is a set of radio buttons where the user is able to pick from a list of different graphs.

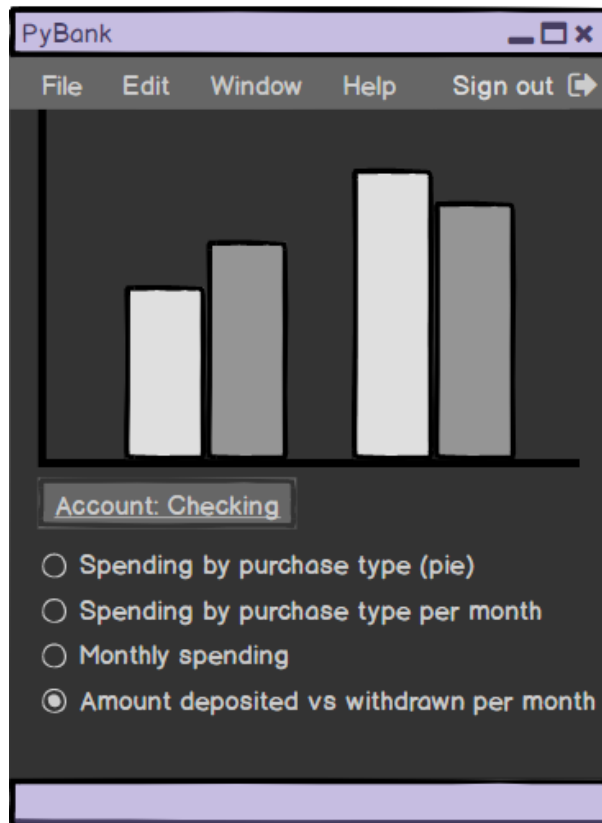


Figure 8: A detailed view of a specific account's visualizations.

4 Alternative Designs

Shown below are two alternative designs that were considered for PyBank but ultimately were discarded for their counterparts which were shown in Section 3. The first design shown below in Figure 9 displays an alternative Sign-in window and the second design also shown below in Figure 10 displays an alternative checking account visualization window.

4.1 Alternative Sign-in Window

The alternative sign-in window displayed below in Figure 9 was not chosen for several reasons including the following: the avatar shown in the window should not be placed at the login before the user is known, the window's navigation menu and title do not have a color scheme that helps contrast the bright text colors, and some common window functions (minimizing/maximizing) are also absent from the design. For these reasons, the sign-in window presented in Section 3, Figure 3 was chosen.



Figure 9: A medium-fidelity prototype alternative for the sign-in window.

4.2 Alternative Visualization Window

The alternative visualization window for checking accounts shown below in Figure 10 was not selected for PyBank because it does not provide the user with a wide selection of options. The visualizations presented in the alternative design are limited to *spending vs. time* analysis whereas the chosen design provides radio buttons with multiple criteria that a customer can look at. Other reasons for dismissing this design are similar to the alternative sign-in window where the window's navigation menu and title do not have a color scheme that helps contrast the bright text colors, and some common window functions (minimizing/maximizing) are also absent from the design. For these reasons, the visualization window presented in Section 3, Figure 8 was chosen.

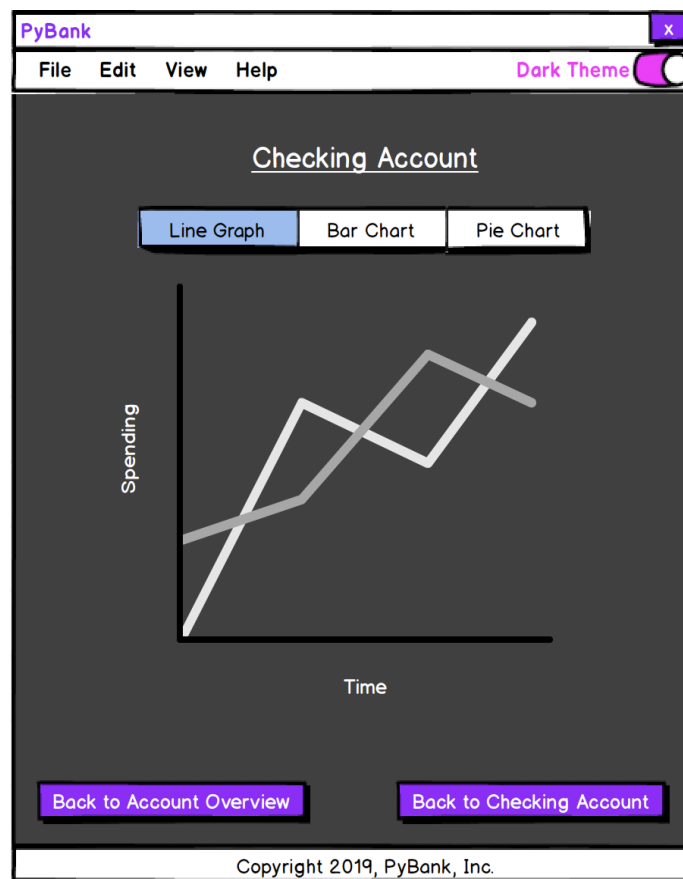


Figure 10: A medium-fidelity prototype alternative for the checking account visualizations window.

5 Annotated References and Resources

Listed below are annotated references which helped produce this document covering PyBank’s design. Two of the references, *The Design of Everyday Things* [1] and *Designing the User Interface* [2] are utilized by CS 420/620 taught by Dr. Sergiu Dascalu. The other five annotated references helped produce our diagrammatic solutions or have aided with the development of PyBank. All annotated references contain hyperlinks to the **References** section which contains the bibliographic information for each reference or resource found at the end of this document.

5.1 The Design of Everyday Things

The Design of Everyday Things [1] is a book that details good and bad designs in everyday things and presents fundamental properties of good design that are applicable across domains. For PyBank, we’ve utilized two concepts heavily during our design process. The first concept, *signifiers*, are described by the author Don Norman as “[...] any mark or sound, any perceivable indicator that communicates appropriate behavior to a person [1].” Our design incorporates signifiers through the use of a variety of PyQt5 widgets such as buttons, labels, text boxes, etc. All of these widgets serve to communicate to a PyBank customer the appropriate behavior for an action within the application. We also use another concept presented in the book, *feedback*, to help provide a user with immediate communication related to the results of an action. PyBank accomplishes this feat through the use of error and information dialog boxes.

5.2 Designing the User Interface

Designing the User Interface [2] provides strategies for effective human-computer interaction. This book covers usability, guidelines, principles, theories, the design process, interaction styles, and design issues. One important concept covered in Chapter 4, Section 4.5.5 is prototyping [2]. The book presents three types, low-fidelity, medium-fidelity, and high-fidelity prototypes. Our static interface designs found in Section 3 and our alternative designs found in Section 4 utilized Balsamiq Mockups 3 [3] which helped us create our medium-fidelity prototypes.

5.3 Balsamiq Mockups 3

Balsamiq Mockups 3 [3] is a software application that provides rapid prototyping through wireframing. One feature of this application are UI Controls such as buttons, text input, drop down menus, radio buttons, menu bars, icons, and many more controls. Another feature of *Balsamiq Mockups 3* [3] is the ability to

share wireframes with other designers and collaborate across platforms and devices using the technologies like Google Drive, the cloud version of the application, among other things.

5.4 Lucid Chart

Lucid Chart [4] is a cloud-based diagramming and visualization software solution that is extremely popular. It is endorsed by 99% of fortune 500 companies and it provides over 500 templates for speeding up the diagramming process. *Lucid Chart* [4] supports multi-page diagram solutions and allows many designers to collaborate together on the same project. An additional benefit of *Lucid Chart* [4] is that it supports templates for technologists and business professionals making it a solution for a variety of practitioners.

5.5 Qt

Qt [5] is the official source for documentation for the Qt API. In our case, our work relates specifically to the Python API for Qt. Qt more broadly is a cross-platform software solution for embedded systems and desktop systems. The documentation provided by *Qt* [5] includes: examples, tutorials, quick-start guides, development tools, and the Qt Creator IDE which supports “[...] a powerful cross-platform integrated development environment, including UI designer tools and on-device debugging [5].”

5.6 ZetCode

ZetCode [6] “[...] brings tutorials for programmers in various areas. The main are Graphical User Interfaces, databases, and programming languages. The website’s mission is to provide competent, quick and easy to understand tutorials for modern-day technologies [6].” This website is maintained by its author, Jan Bodnar. It has provided useful tutorials for the development of PyBank.

5.7 Material Design

Material [7] “[...] Material is a design system – backed by open-source code – that helps teams build high-quality digital experiences [7].” Material provides four main selections from the homepage which are the following: Design, Components, Develop, and Resources. All of these selections have many sub-categories. Material provides a vast set of design tips based on its philosophy for creating user interfaces. Everything from layout to navigation to typography, etc. is covered by the website. Material also provides icons, Google fonts, and many other resources.

6 Contributions of Team Members

6.1 Brianna Blain-Castelli

Individual contributions

- Abstract, Section 1
- Static Interface Design, Section 3
 - Initial sketches.
 - Mock-up design prototyping utilizing *Balsamiq Mockups 3* [3].
 - * Created and wrote the description for the sign-in window shown in Figure 3 using *Balsamiq Mockups 3* [3].
 - * Created and wrote the description for the sign-up window part II shown in Figure 3 using *Balsamiq Mockups 3* [3].
 - * Created and wrote the description for the account overview window shown in Figure 5 using *Balsamiq Mockups 3* [3].
 - * Created and wrote the description for the checking account window shown in Figure 6 using *Balsamiq Mockups 3* [3].
 - * Created and wrote the description for the online transfer funds window shown in Figure 7 using *Balsamiq Mockups 3* [3].
 - * Created and wrote the description for the visualization window shown in Figure 8 using *Balsamiq Mockups 3* [3].
 - Section write-up and captioning.

6.2 Nikkolas Irwin

Individual contributions

- High-Level Design, Section 2
 - Created the architectural diagram shown in Figure 1 using Lucidchart [4] and provided the written description of the architectural diagram.
 - Created the activity diagram shown in Figure 2 using Lucidchart [4] and provided the written description of the activity diagram.

- Alternative Designs, Section 4
 - Created and wrote the description for the alternative sign-in window shown in Figure 9 using *Balsamiq Mockups 3* [3].
 - Created and wrote the description for the alternative checking account visualization window shown in Figure 10 using *Balsamiq Mockups 3* [3].
- Annotated References and Resources, Section 5
 - Wrote the annotated reference for *The Design of Everyday Things* [1], Section 5.1.
 - Wrote the annotated reference for *Designing the User Interface* [2], Section 5.2.
 - Wrote the annotated reference for *Balsamiq Mockups 3* [3], Section 5.3.
 - Wrote the annotated reference for *Lucid Chart* [4], Section 5.4.
 - Wrote the annotated reference for *Qt* [5], Section 5.5.
 - Wrote the annotated reference for *ZetCode* [6], Section 5.6.
 - Wrote the annotated reference for *Material* [7], Section 5.7.

References

- [1] Don Norman, *The Design of Everyday Things*. Basic Books, Revised & Expanded ed., 2013.
- [2] B. Shneiderman, C. Plaisant, M. Cohen, S. Jacobs, N. Elmqvist, and N. Diakopoulos, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson Education Limited, Sixth ed., 2018.
- [3] Balsamiq, “Balsamiq, Rapid, effective and fun wireframing software..” <https://balsamiq.com>, 2019. Accessed on 2019-11-14.
- [4] Lucidchart, “Lucidchart: Online Diagram Software & Visual Solution.” <https://www.lucidchart.com/pages/?noHomepageRedirect=true>, 2019. Accessed on 2019-11-14.
- [5] Qt, “Qt: The Qt Company.” <https://doc.qt.io/qt-5/>, 2019. Accessed on 2019-11-14.
- [6] Jan Bodnar, “PyQt5 tutorial.” <http://zetcode.com/gui/pyqt5/>, January 2018. Accessed on 2019-11-14.
- [7] Google, “Material Design.” <https://material.io/>, . Accessed on 2019-11-14.