

Human-Computer Interaction

CS 420/620

Project Part 2: Requirements Discovery and Specification

Team 01

Brianna Blain-Castelli and Nikkolas Irwin

November 1, 2019

Department of Computer Science and Engineering

College of Engineering, University of Nevada, Reno

Course Instructor: Dr. Sergiu Dascalu

Contents

1 Abstract	1
2 Requirements Discovery	2
3 Use Cases and HCI Scenarios	4
3.1 HCI Persona 1	4
3.2 HCI Persona 2	5
3.3 Use Case Diagram	6
4 Functional Requirements	8
4.1 Level 1 Requirements	8
4.2 Level 2 Requirements	9
4.3 Level 3 Requirements	9
References	10

List of Figures

1	A photo of a fictional male persona for the PyBank application [1].	4
2	A photo of a fictional female persona for the PyBank application [2].	5
3	A use case diagram, created using Lucidchart [3], for PyBank, an interactive desktop client banking application.	7

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 application will utilize the principles of human-computer interaction such as the Fundamental Principles of Interaction [4] and the Eight Golden Rules of Interface Design [5] so that the application provides an enjoyable experience to its end-users.

2 Requirements Discovery

Three users were interviewed regarding Pybank to determine what potential users would like to see from this application. The stakeholders who took part in this process were: Chase Carthen, Joseph Masset, and Vinh Le.

"What do you like about any banking or budgeting applications you have seen or utilized yourself?"

In response to this question, two users both mentioned that they enjoy being able to view their accounts and associated transactions. Chase elaborated that he specifically likes when banks offer options to auto group the transactions as well. On the other hand, Vinh noted the convenience of mobile banking instead, stating that he enjoys being able to scan checks and payments digitally without having to travel to a local bank.

"What qualities do you not like about current banking or budgeting application and web clients that you currently use or have seen?"

All three users focused on different qualities of applications that they currently utilize. Joseph commented that he dislikes how long it takes for data to pool from each location, resulting in inaccuracy of account summaries and transaction details while waiting for the summary to reflect the actual balance. Chase mentioned that he dislikes how functionality for many applications and websites directly employed by banks do not always reflect one another. For this reason, one version may not be able to support use by specific users. Lastly, Vinh responded that the lack of notifications from current banking apps and methods of notification delivery when an alert system is in place can be frustrating. He suggested that notifications for upcoming payments that display directly on a phone or computer would be very useful.

"If you had the option to have your banking and transaction information displayed visually in a graph, what would you like to see displayed?"

Two main suggestions were made by the users that were interviewed regarding this application. First, users mentioned that they would find a pie chart or bar graph showing a monthly breakdown of expenditures and budgeting based on transaction purpose to be helpful. Secondly, a line graph that shows income versus expenses per month was recommended in order to visually show successful and less successful saving and profits.

"What is your favorite type of graph to receive information visually and why?"

A general consensus was reached by all users that they had a preference for bar graphs and line graphs when looking at financial data. It was stated that, due to clearly marked divisions in categories, it is easy to compare spending of different types when viewing a bar graph. On the other hand, line graphs are better for continuous data and users find their analysis of trends easy to understand.

"What is the main operating system that you prefer to utilize?"

All three users interviewed utilize Windows and Linux as their primary operating systems and favor Windows for regular use. For this reason, they expressed interest in the application being compatible with both Windows and Linux; however, in the case that cross-platform functionality cannot be achieved, the users would prefer a Windows client.

"When utilizing a banking or budgeting application, on average what is the primary function you utilize that it provides? (Ex: Account summaries, transaction details, etc)"

All three users utilize banking applications with the same primary purpose, to see current account summaries and balances. While individual transaction details and transferring money between accounts were also mentioned in response, they were labelled as secondary to an overall summary of account statuses.

"Which features do you rarely utilize in banking or budgeting applications?"

All three users have different features that they rarely utilize in banking applications, providing different options for deciding priority among functions. Joseph commented that he rarely uses options to follow or graph a budget due to lack of granular control that results in incorrect reporting. Chase instead noted that he does not utilize functions to create spending caps as they are not useful. Lastly, Vinh stated that he does not check his monthly statement on banking applications because it downloads a PDF containing sensitive account information locally to a device. This option is not secure and allows easy access to user information.

"When adding a transaction, what information do you find necessary to provide to a banking or budgeting application?"

Users responded stating that the most important information to provide are transaction amount, vendor name, account pulled from, and transaction type, such as rent, groceries, or entertainment. Joseph also recommended the addition of an optional notes section that can be regularly edited. This allows the user to include additional details about the transaction that would not typically be provided to better organize why the expenditures are necessary or to create more granular types within the broader spectrum to which the transactions fit. While he has not seen this in practice, he feels it would be a helpful addition to current transaction recording forms.

"What information are you often asked for when filling forms do you find superfluous?"

A variety of issues were mentioned surrounding various form fields utilized by banks. In particular, Joseph and Vinh were concerned with personal identifying information that is often requested for the addition of transactions. They expressed that they disliked how much private information needed to be sent to budget and do not view this information as necessary. Chase also mentioned personal information; however, he was frustrated more by the redundancy surrounding this information than privacy. He stated that it is superfluous when banks asks repeatedly for a user's address on forms when the banks already have this information on file.

"Is there anything you would like to add?"

One user recommended support for various currency types internationally as well as an account option for various cryptocurrencies. This special account type would display information regarding trends in cryptocurrency values and how that value translates into their current holdings similar to stock trading boards. Additionally, the importance of security was mentioned by another user. Extra security is more marketable and will make the service more appealing.

3 Use Cases and HCI Scenarios

3.1 HCI Persona 1



Figure 1: A photo of a fictional male persona for the PyBank application [1].

- **Fictional Name:** John Doe
- **Job Title:** Accountant
- **Demographics:**
 - **Age:** 26
 - **Marital Status:** Single
 - **Degree:** B.A. Accounting
- **Goals and Tasks:**
 - Perform accounting services for local companies using PyBank’s desktop client.
 - Learn PyBank’s interface in-depth to become a power user.

Environment

John, an accountant for a local firm, typically performs work on a desktop computer system. He’s a busy individual, handling the accounts of both local and widespread businesses. John’s spends the majority of his time performing financial computations, answering phones, and writing complex reports with visualizations. Sometimes, when John has to travel, he performs work-related tasks on a laptop computer system instead of other mobile alternatives because he always wants to have the highest performance while working for his clients.

Scenario

John recently began merging all of his clients accounts into to the PyBank desktop application. During a recent trip, where internet activity was scarce, John was prepared. Thanks to Pybank, John was still able to perform accounting tasks for his clients that were processed, recorded, and once he regained internet access, updated in the cloud. PyBank’s flexibility and accessibility allowed John to perform work that other individuals were restricted from performing with traditional banking systems.

3.2 HCI Persona 2



Figure 2: A photo of a fictional female persona for the PyBank application [2].

- **Fictional Name:** Jane Doe
- **Job Title:** Financial Analyst
- **Demographics:**
 - **Age:** 28
 - **Marital Status:** Married
 - **Degree:** B.A. Finance
- **Goals and Tasks:**
 - Manage personal finances using PyBank's desktop client from the comfort of home.
 - Utilize PyBank's financial visualizations for insights and reports.

Environment

Jane is a freelance financial analyst who prefers to work remotely. Her work is generally done on a laptop since she often enjoys working at the coffee shop or other social venues. Jane prefers summarized data visualizations that help her make quick analytical interpretations.

Scenario

Recently, at Jane's favorite local coffee shop, Jane decided that she wanted to analyze her families' personal finances and present the results to her husband in an easy to understand manner. With this goal in mind she focused on creating histograms and pie charts for categorical data and a line graph for plotting her families spending over time. With the help of PyBank, she was able to generate these visualizations and produce a summary of her families spending in no time, all while consuming her favorite coffee drink!

3.3 Use Case Diagram

Listed below are brief descriptions of eleven use cases that describe a sequence of actions that can be performed by a PyBank customer (the end-user) interacting with PyBank, an interactive desktop client banking application. These use cases are further depicted in Figure 3 shown below after the use case descriptions.

1. **CreatePyBankAccount (UC_01):** A PyBank customer can create an account from PyBank's main application window by selecting the **Create An Account** button.
2. **DepositSavingsAccount (UC_02):** A PyBank customer can deposit money into an existing savings account by selecting the **Make A Deposit** button from the Savings Account application window.
3. **DepositCheckingAccount (UC_03):** A PyBank customer can deposit money into an existing checking account by selecting the **Make A Deposit** button from the Checking Account application window.
4. **SignInToPyBankAccount (UC_04):** A PyBank customer can sign in to an existing account from PyBank's main application window by selecting the **Sign In** button.
5. **WithdrawSavingsAccount (UC_05):** A PyBank customer can withdraw money from an existing savings account by selecting the **Make A Withdrawal** button from the Savings Account application window.
6. **AddCheckingAccount (UC_06):** A PyBank customer can request a checking account from PyBank by selecting the **Add A Checking Account** button from the main application window. A PyBank customer may have one checking account per PyBank account.
7. **WithdrawCheckingAccount (UC_07):** A PyBank customer can withdraw money from an existing checking account by selecting the **Make A Withdrawal** button from the Checking Account application window.
8. **ViewSavingsAccountGraphs (UC_08):** A PyBank customer can view graphical charts related to their savings account by selecting the **View Graphs** button from the Savings Account application window.
9. **ViewCheckingAccountGraphs (UC_09):** A PyBank customer can view graphical charts related to their checking account by selecting the **View Graphs** button from the Checking Account application window.
10. **AddSavingsAccount (UC_10):** A PyBank customer can request a savings account from PyBank by selecting the **Add A Savings Account** button from the main application window. A PyBank customer may have one savings account per PyBank account.
11. **SignOutOfPyBankAccount (UC_11):** A PyBank customer can sign out of an existing account from PyBank's main application window by selecting the **Sign Out** button from the navigation menu.

Use Case Diagram

Brianna Blain-Castelli and Nikkolas Irwin | October 27, 2019

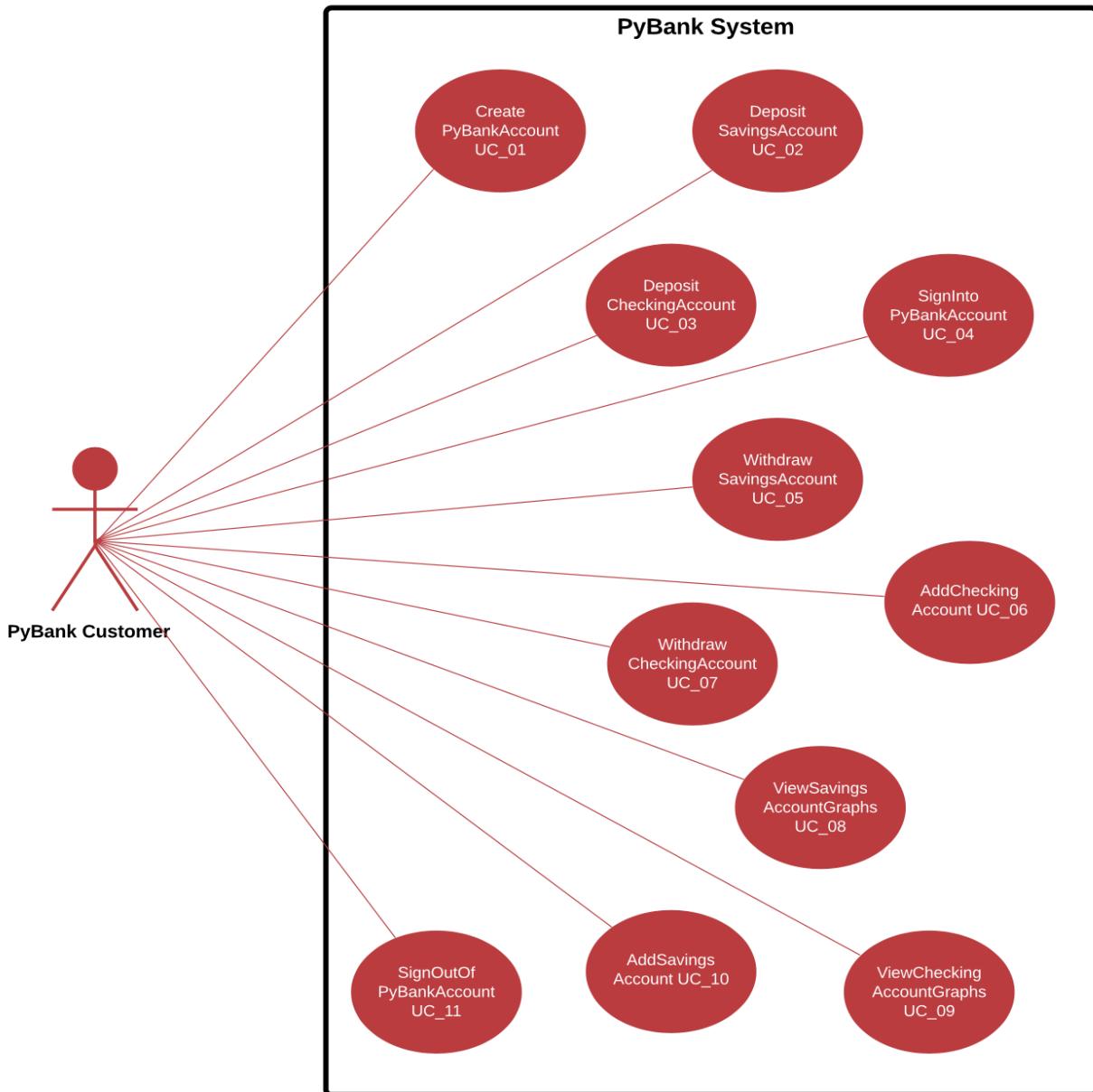


Figure 3: A use case diagram, created using Lucidchart [3], for PyBank, an interactive desktop client banking application.

4 Functional Requirements

4.1 Level 1 Requirements

Listed below are the level 1 functional requirements which detail functions and features that will be covered in the prototypes interface due in December 2019 and will be fully implemented (from an execution point of view).

1. The system shall provide the end-user access to PyBank through an interactive graphical user interface.
2. The system shall provide an application window for creating a new account or signing into an existing account using text-entry boxes and buttons.
3. The system shall provide username and password checking to verify an end-user's login credentials.
4. The system shall provide the end-user with error dialog windows as feedback for rejected actions.
5. The system shall provide an account overview for an existing user account in the main application window.
6. The system shall provide the end-user with dialog windows as feedback for successful transactions (deposit/withdraw).
7. The system shall request end-user verification before proceeding with any transaction.
8. The system shall provide the ability to request a checking account from the main application window.
9. The system shall provide the ability to view an end-user's checking account from the account overview main application window once signed in.
10. The system shall provide the ability to request a savings account from the main application window.
11. The system shall provide the ability to view an end-user's savings account from the account overview main application window once signed in.
12. The system shall provide various menu options via a menu bar at the top of the main application window.
13. The system shall provide visualizations related to the end-user's checking and spending accounts.
14. The system shall provide the end-user with buttons in their checking and/or savings account(s) to perform actions for depositing and withdrawing money from PyBank.

4.2 Level 2 Requirements

Listed below are the level 2 functional requirements which detail functions and features that will be covered in the prototype's interface due in December 2019 but will not be fully implemented.

1. The system shall provide the ability to transfer funds to other PyBank customers.
2. The system shall provide a 'switch theme' menu option.
3. The system shall provide the ability to request a credit card account.
4. The system shall provide the ability to view an end-user's credit card account from the account overview main application window once signed in.
5. The system shall provide a 'help' menu option.

4.3 Level 3 Requirements

Listed below are the level 3 functional requirements which detail functions and features that will not be covered in the prototype due in December 2019 but would be useful in a possible continuation of the project beyond the time frame of this course.

1. The system shall use modern encryption techniques to protect the account creation and sign-in processes.
2. The system shall provide advanced visualizations and using statistical techniques.
3. The system shall add additional options for customizing the applications interface.
4. The system shall provide predictive models based on spending trends.
5. The system shall provide the end-user with detailed reports related to their existing accounts.

References

- [1] Pexels, “Man Smiling Behind Wall.” <https://www.pexels.com/photo/adult-beard-boy-casual-220453/>, November 2016. Accessed on 2019-10-27.
- [2] Pexels: The best free stock photos & videos shared by talented creators., “Woman In Collared Shirt.” <https://www.pexels.com/photo/women-s-white-and-black-button-up-collared-shirt-774909/>, January 2018. Accessed on 2019-10-27.
- [3] Lucidchart, “Lucidchart: Online Diagram Software & Visual Solution.” <https://www.lucidchart.com/pages/?noHomepageRedirect=true>, 2019. Accessed on 2019-10-27.
- [4] Don Norman, *The Design of Everyday Things*. Basic Books, Revised & Expanded ed., 2013.
- [5] 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.