$implicity: A Simple Financial Tracking Software

General Report 2

5 December 2014

Engineers:

Michael Esry

Bill Kolega

Nicholas Roudebush

Tyler Steiner

Prepared for

KU EECS 448—Software Engineering I

Instructor: Swapan Chakrabarti, Ph.D.

Fall 2014

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 4 Dec 14 | Preliminary Version | Tyler Steiner |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Michael Esry | Engineer |  |
|  | Bill Kolega | Engineer |  |
|  | Nicholas Roudebush | Engineer |  |
|  | Tyler Steiner | Engineer |  |

**Table of Contents**

Revision History 3

Document Approval 3

1. Agile Method Choice 5

1.1 Method Choice 5

1.2 XP Steps to Aid in Completion of Project 5

2. Use Case Diagram 6

3. System Architecture 7

4. Class Diagram 7

5. Sequence Diagram 8

6. Github 8

6.1 Github Use 8

# 1. Agile Method Choice

## 1.1 Method Choice

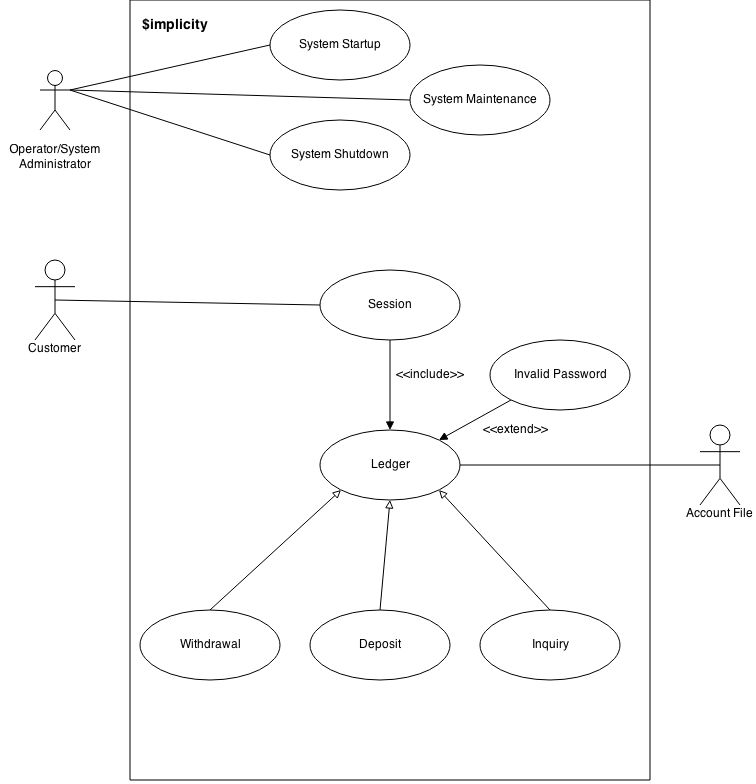
For this project, our team chose to use the Extreme Programming (XP) agile method. XP’s thorough testing and verification process is very appealing when it comes to making sure a financial tracking software, like $implicity, works correctly. When it comes to an end product that deals with money, making sure it works correctly every time is extremely important to customers. After considering the different options, Extreme Programming was definitely the best choice for this project.

## 1.2 XP Steps to Aid in Completion of Project

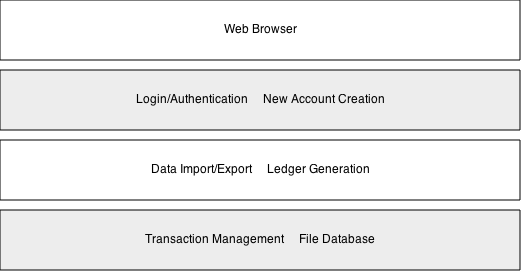
There are a number of different steps or parts of the XP method that will help in completing this project.

1. Coding/Peer Programming
   1. Because of the way our team is using Github, everyone is able to view and edit every part of the project. Everyone also gets notified when changes are made. This way, if a team member sees something that is incorrect, it is easy to remedy the problem.
2. Testing
   1. Testing, especially for a financial software, is vital. Extreme Programming is known to take testing to the extreme to find any and all flaws in the program before release.
   2. Our program development is separated into 3 main webpages that can each be developed individually. Unit testing will allow these individual pages to be tested as they are developed before integrating them into the whole.
   3. Acceptance testing will allow us to test all 3 webpages as a whole and verify that the requirements specified are met.

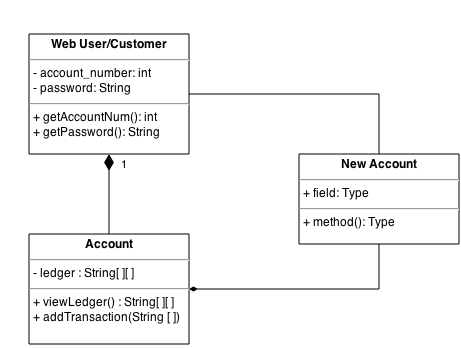
# 2. Use Case Diagram



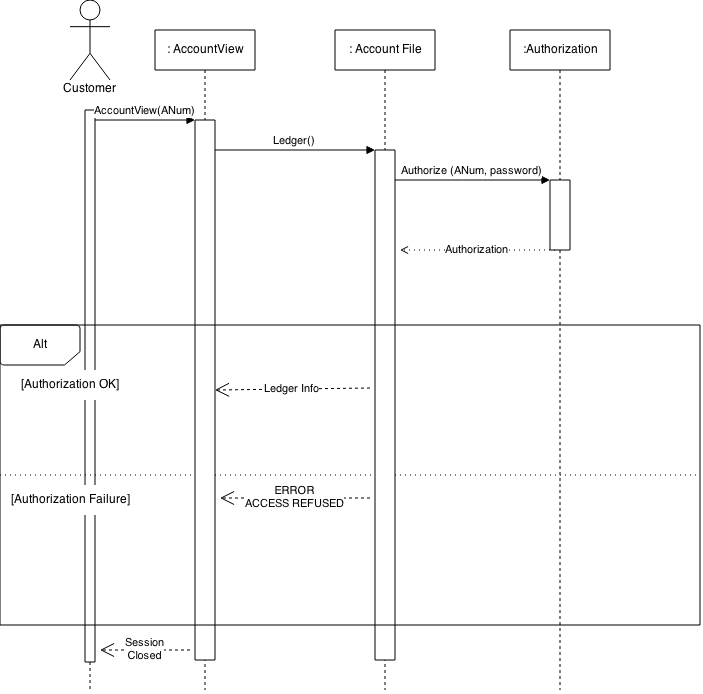
# 3. System Architecture



# 4. Class Diagram



# 5. Sequence Diagram



# 6. Github

## 6.1 Github Use

This project introduced the new element of using Gitlab or Github as a repository system to coordinate among group members. Our group elected to use Github even though we risk not having any support should something go wrong. As was discussed briefly earlier, Github is allowing us to be more proactive with double-checking all the code that goes up because each engineer gets a notification about changes made to the master branch of the repository. We are also using the incidents feature of Github to ease communication among the team regarding things that need to be done or things that need to be kept track of during the development process (example: future enhancements). However nice these are, though, the best way that our team is making use of Github for this project is in allowing us to each program/develop the software wherever we individually choose to and the changes will be automatically shared with the rest of the engineering team.