Financial Management Application
for UTM CSCI 352 Fall 2017

Lucian Freeze and Brett Whitson)

Abstract

Abstracts are generally a very concise, short description of what you are writing about. It's good to get into the habit of writing them. Give a 1 paragraph explanation of your project, who the target is, and what you feel you have achieved so far in the project.

This project is an application which allows a user to view and manage financial accounts and budgets. The program will initiallly use, as a basis, the factory method pattern to generate and edit instances of accounts and budgets. Target audience for this application includes any persons looking to better manage their finances.

1. Introduction

Your introduction should give an introduction to your project. What are you trying to accomplish (high level), who do you expect to target with your project? What do you expect your target audience will get out of it?

Should you need to cite anything, use the *cite* keyword, and refer to something from your bibliography. For example, this was put together with the help of a LaTeXguide [1].

Make sure that by the end of your introduction the reader knows what your project is and why you are doing it.

Everyone likes to save money. Having a simple and intuitive application to manage ones finances would make this much easier. This application intends to serve that purpose by giving the user the tools needed to achieve this goal.

A section to keep track of accounts and balances is to be implemented, allowing a user to view this information and add/remove accounts with possible customizations.

Another section for budget creation and editing is also to be implemented.

1.1. Subsection Heading Here

Occasionally you need to break your sections into separate parts, you will likely not need a subsection for every section

1.1.1. Subsubsection Heading Here. Occasionally you will need to break your subsections into separate parts, if you find yourself using this often, you're likely going overboard. Don't try to go any lower down than this...

1.2. Background

In this section, give a brief background to any concepts the reader may need to know to make it through your paper. Are there any specific terms the reader needs to be familiar with?

Also, consider going into your personal connection with this project – why did you decide to do it?

This project derived from the demand for better money management and closely matches the type of design patterns covered in CSCI352.

1.3. Challenges

For your first deliverable, this should be a list of where you believe you will get stuck – what do you see as the most difficult portion of your project?

As you overcome these challenges, this section becomes more of a retrospective: don't only identify the challenge, but also briefly outline the solution!

Possible challenges for this project include implementing graphical components and managing informations security within the application.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Add item to cart	Shopper	Med	1
2	Checkout	Shopper	Med	1

TABLE 1. SAMPLE USE CASE TABLE

2. Scope

This section is a bit tricksy. You are going to do your best to set up ground rules: How will you know when your project is done?

If you were doing this under contract for a company, this would be your checklist to make sure you get paid. We will be going into this in more detail over time, but you should start planning your major goals of the project as soon as possible.

For every sub(sub)section below, make sure to mark which items are basic goals (project won't be done without it) and which ones are stretch goals (it would be really cool to do...). We will be meeting one-on-one to help identify which goals go where.

Our project aims to develop a simple, operable program in which users can securely and easily manage their finances. When we have fully implemented account and budget functionality along with some aesthetic aids and security measures, the project will be at completion.

Additional goals for this project include bank account integration and the inclusion of investment features.

2.1. Requirements

As part of fleshing out the scope of your requirements, you'll also need to keep in mind both your functional and non-functional requirements. These should be listed, and explained in detail as necessary. Use this area to explain how you gathered these requirements.

2.1.1. Functional.

- User needs to have a private shopping cart – this cannot be shared between users, and needs to maintain state across subsequent visits to the site
- Users need to have website accounts – this will help track recent purchases, keep shopping cart records, etc.
- You'll need more than 2 of these...

2.1.2. Non-Functional.

- Security – user credentials must be encrypted on disk, users should be able to reset their passwords if forgotten
- you'll typically have fewer non-functional than functional requirements

2.2. Use Cases

This subsection is arguably part of how you define your project scope (why it is in the Scope section...). In a traditional Waterfall approach, as part of your requirements gathering phase (what does the product actually *need* to do?), you will typically sit down with a user to develop use cases.

You should have a table listing all use cases discussed in the document, the ID is just the order it is listed in, the name should be indicative of what should happen, the primary actor is typically most important in an application where you may have different levels of users (think admin vs normal user), complexity is a best-guess on your part as to how hard it should be. A lower number in priority indicates that it needs to happen sooner rather than later. A sample table, or Use Case Index can be seen in Table 1.

Use Case #: 1
Use Case Name: Create User Account
Description: A user would like to create an account within the application. They will click on a "Create User Account" button. This event will transition the user to a new page where they can enter their user information and confirm their account.

Flow:

1) User left-clicks on "Create User Account" button on the initial page.
2) User is presented with a new page to enter user account details such as username, password, etc.
3) If username is not taken, user account is created and the Dashboard page is shown. Otherwise, new username must be chosen.

At Termination: The user now has a user account and access to the rest of the application's utlilities.

Use Case #: 2
Use Case Name: Create Financial Account
Description: A user would like to create a financial account within the application. They will click on a "New Account" button. This event will trigger a new page to appear, asking the user which type of account to create. The user clicks the appropriate button and is presented with another page to enter account details.

Flow:

1) User left-clicks on "New Account" button on the Dashboard's Accounts tab.
2) User is presented with a new page with "Checking" and "Savings" buttons. User left-clicks the appropriate choice.
3) A new page is presented which allows the user to edit details for the chosen account type (name, balance, etc.)

At Termination: The user now has a financial account of the chosen type.

Use Case #: 3
Use Case Name: View Budgets
Description: A user would like to view their budgets within the application. They will click on "Budgets" tab. This event will trigger the "Budgets" page to appear. The user can then view their budgets.

Flow:

1) User left-clicks on the "Budgets" tab.
2) User is presented with the "Budgets" tab/page.

At Termination: The user is able to view their budgets.

Figure 1. First picture, this is a kitten, not a use case diagram

2.3. Interface Mockups

At first, this will largely be completely made up, as you get further along in your project, and closer to a final product, this will typically become simple screenshots of your running application.

In this subsection, you will be showing what the screen should look like as the user moves through various use cases (make sure to tie the interface mockups back to the specific use cases they illustrate).

3. Project Timeline

Go back to your notes and look up a typical project development life cycle for the Waterfall approach. How will you follow this life cycle over the remainder of this semester? This will usually involve a chart showing your proposed timeline, with specific milestones plotted out. Make sure you have deliverable dates from the course schedule listed, with a plan to meet them (NOTE: these are generally optimistic deadlines).

4. Project Structure

At first, this will be a little empty (it will need to be filled in by the time you turn in your final report). This is your chance to discuss all of your design decisions (consider this the README's big brother).

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure 2.

Figure 2. Your figures should be in the *figure* environment, and have captions. Should also be of diagrams pertaining to your project, not random internet kittens

4.2. Design Patterns Used

Make sure to actually use at least 2 design patterns from this class. This is not normally part of such documentation, but largely just specific to this class – I want to see you use the patterns!

This application will use the Factory Method pattern

5. Results

This section will start out a little vague, but it should grow as your project evolves. With each deliverable you hand in, give me a final summary of where your project stands. By the end, this should be a reflective section discussing how many of your original goals you managed to attain/how many desired use cases you implemented/how many extra features you added.

5.1. Future Work

Where are you going next with your project? For early deliverables, what are your next steps? (HINT: you will typically want to look back at your timeline and evaluate: did you meet your expected goals? Are you ahead of schedule? Did you decide to shift gears and implement a new feature?) By the end, what do you plan on doing with this project? Will you try to sell it? Set it on fire? Link to it on your resume and forget it exists?

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.