Budgetable Financial budgeting app

Development team:

Huiyu Jia, Nick Alessi, Michael Hook, Jate Li

Executive Overview

As the world we live in becomes more expensive and the fact that our personal incomes are reflecting those changes, it has become more crucial for the people to create a budget and change their spending habits. Budgetable is here to help.

With the price of living going up year after year and the job market being as competitive as it is, people aren't earning enough to live an easy life, where they don't have to worry about when their next bill has to be paid. Many people out there, including the TopCatAppDevelopment team, have to watch and budget their spending so they can make it to the next payday.

Therefore we are developing Budgetable – the app that allows users to track their spending over time, through individual transactions and compare that to a self administered budget. Our application will help people to see where their money is going and allow them to change the way they spend.

The idea for a budgeting app is not a new one, but Budgetable will be a simple, easy to use application with an intuitive design.

As the cost of living increases we all will need a helping hand to keep as much of our money as we can. Budgetable is here to help.

Introduction

Our group, under the identity of TopCatAppDevelopment, are developing a financial budgeting application for use on ios devices. The development of this software is for the COSC345 Software Engineering paper taught at the University of Otago.

The financial budgeting application will allow users to track their current and previous expenditure in relation to the budget they set themselves. Users will be able to use this comparison to see where they are overspending if applicable and improve their spending habits.

Project Description

TopCatAppDevelopment's financial budgeting application has currently been given the name "Budgetable". Budgetable is an app we believe we would use daily when we update our expenditure and keep track of our financial application. It will be an ios application, controlled through an easy to use interface and distributed through the ios app store. This description will be centered around the key functional requirements we have identified the app to need.

- A budget setting will be available to the user. We are currently looking at implementing it in terms of per week and per month. Per day may be added if deemed necessary.
- The user will be able to specify their income and modify it as needed. This is if they wish to have a budget for spending as well as a savings goal.
- The user will be able to set a savings goal.
- The user will have the ability to add, edit and delete individual expenses. This includes modification of the date of the expenditure or bill.
- The application will keep track of recorded expenses in a Timeline format. This timeline will be able to be scrolled through by the user so they can check their spending habits.
    - Total expenses over specific periods of time will be viewable. This may be shown in terms of a chart or a simple output table with specified values such as total spent on x and total overall.
- The user will be able to input fixed payments that occur at a regular interval, such as weekly or monthly. An example of this may be rent or mortgage payments.
- For the necessary functions above the system will have a calendar type function. We are still investigating how this will work but it is our aim to put this in as it is necessary for other functions.

A more in depth list of our functional and nonfunctional requirements can be accessed on github. Some of the requirements in the file only and not above in the report are under consideration may not be in the final version. The file link is below.
https://github.com/TopCatAppDevelopment/Budgetable/blob/master/Planning/Requirements.docx

Resource Requirements

Based on the fact that time is a resource, we expect the project to take 10 days in total to code and test. We have estimated that we will individually work on the project for at least 4-6 hours per week, in order to finish on time.

The software we have decided to use for this project will all be free. The only possible cost would be to register as an apple developer, for a cost of $99 USD, in order to publish our app. We will publish our app on the apple app store and it will be free for download.

In terms of scheduling, we have split up the project into different jobs, based on requirements. A certain number of these jobs are assigned each week (a sprint) and should be completed by the end of that week. We have set up weekly sprint meetings to discuss and communicate progress. This is so that we can keep track of what has been completed and what still needs doing. We are using multiple tools for project management, including github for version control, and the collection and sharing of files.

The software tools that we will be using for this project includes:

- C++ and SWIFT Programming Languages
- Xcode
- Make
- Gcc
- Travis CI
- Blackboard
- Stack Overflow
- Github
- Google
- Smartsheet

These resources will be used for development, testing, building and sharing the project, as well as learning the required skills.

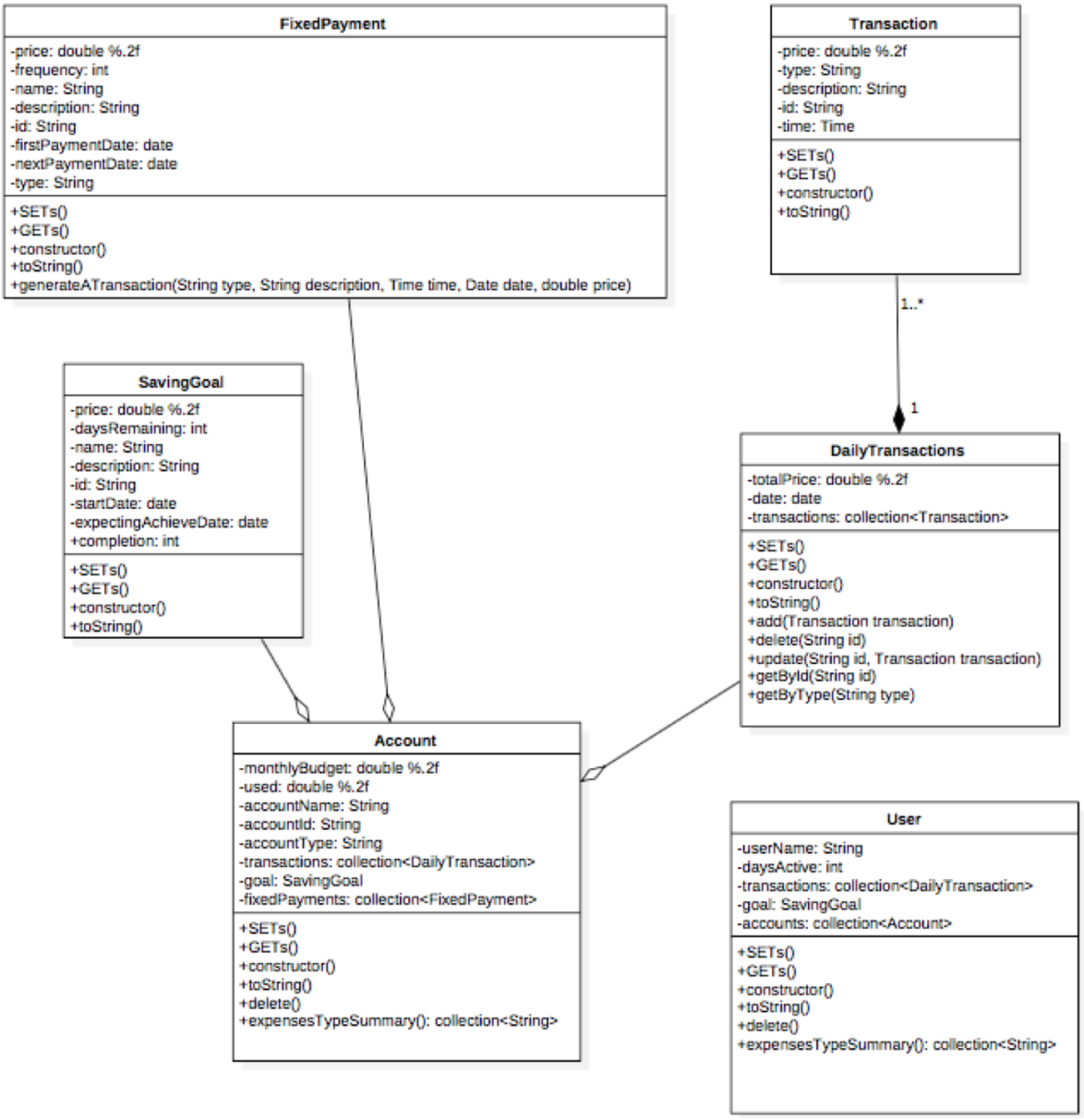
Organization

The four members of TopCatAppDevelopment are Huiyu Jia, Jate Lee, Michael Hook and Nick Alessi. In regards to app development our primary coders will be Huiyu Jia and Jate Lee. All members will be working on the code for this project but they are the primary coders with Michael Hook and Nick Alessi primarily working as code checkers, testers and debuggers. Depending on how development goes and problems that crop up members may take over other roles and as such there some fluidity within member responsibility. We are using the smartsheet website to assign jobs and notify those working on it, when they should be completed by.

Project Breakdown

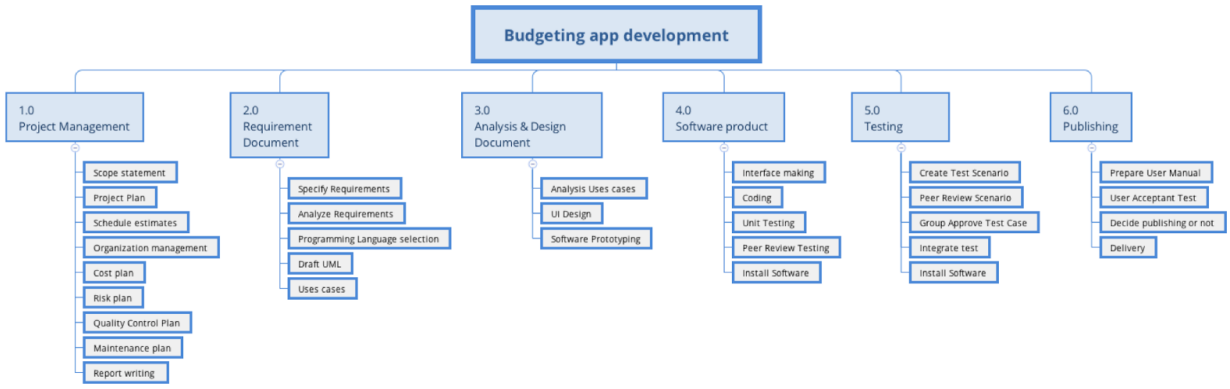
1. UML.
https://github.com/TopCatAppDevelopment/Budgetable/blob/master/Planning/UML.png

This UML is an initial design for the C++ code we'll be writing.
Each class has its corresponding system requirement.
The "User" class is a backup plan if we have enough time to include a signin function.



2. WBS.
https://github.com/TopCatAppDevelopment/Budgetable/blob/master/Planning/BreakdownStructure.png

The WBS is the breakdown structure of our project.
1.0 - 6.0 are activities of our project. And their sub-topics are our milestones.



Risk Analysis

There are many risks that come into play when working on a team project. The following are the 5 most important risks, ranked in order of highest to lowest damage to the project (eg. cost, time damage).

Risks

1 - Member Sickness / Absence (a member not being able to complete their workload)
2 - Lack of Experience (xcode, c++, objective c++)
3 - Hardware Failure (project data being lost or corrupted)
4 - Software Failure (compilers, other software breaking or containing project-breaking bugs)
5 - Time management (members have time in their days to work on the project)
6 - Specifications Availability (we may not have access to all of the OS's or hardware or software needed for testing)
7 - Requirements Change (we may need to add or remove features from the app as we go along)
8 - Project Size Underestimation (we may estimate wrong and this may lead to not finishing on time)

Solutions

1 - Do not rely heavily on one member, evenly split jobs so that work can be done in parallel
2 - For the estimations of our schedule, estimate slightly higher to give members time to learn code and get used to using software
3 - Keep local backups and cloud storage of all files. (including github files, frequently)
4 - Plan for bugs and failure, choose the best software for the job. Test a lot! Use formal inspection.
5 - Allocate a certain amount of hours for each person to work on the project. This will be estimated for each job
6 - Using Travis CI for testing and emulators in XCode
7 - Initially specify as many requirements as we can and do not depend heavily on a few number of requirements
8 - Estimate using a proper technique and take lack of experience (learning) into consideration when estimating

Project Schedule

The following link is a gantt chart that we made on smartsheet.com. It specifies the duration for every milestone we've set, in order to let everybody learns as much as possible. We decided to learn the techniques for doing our milestone together. So basically everybody will have his own version of each milestone. Then we'll have a group meeting to merge them together or pick a best one.
https://github.com/TopCatAppDevelopment/Budgetable/blob/master/Planning/GanttChart.pdf

Monitoring and Reporting

1.When working in a group, it is important to do more than just talk about our project in motion and the steps we need to take. Every group member needs to be kept up to date on the progress of the project. If the project needs to be tested, debugged or adjusted, the group together, then effectively and efficiently find and fix the problem. Therefore when any member edits a part of the project or completes a milestone the project record in the GitHub repository needs to be updated with the time, date and what was modified/completed.

2. Regular meetings and communication. Regular meetings and communication are necessary to keep all the members up to date and on schedule in regards to the development and any problems that pop up along the way. A record of the topics and content are necessary for review and reference when going forward with the project. A document could be created for the content of each meeting and either kept on github or a shared google drive.

3.The checklist is an important and necessary part of the project schedule. Individual work should be arranged in an efficient manner so multiple parts of the project can be developed simultaneously. We will need a clear set of guidelines for our coders to follow. To make sure our code has a logical structure, variables will have meaningful names and comments will clearly explain code function (McConnell's Checklist Lecture Document).
https://github.com/TopCatAppDevelopment/Budgetable/blob/master/Planning/CodeChecklists.pdf

4.Divide the project into manageable segments. A milestone is something to be accomplished in a day. Checkpoints will be set within a milestone to make the project easier to track, test, adjust and debug.

5. The Apple Test Flight developer's application will allow us to gather feedback from multiple test sources. This feedback can be used to identify bugs and improve the app for the user. We will also ensure quality control by using formal inspection of the code, good test cases, component testing and system testing.

When reports are to be delivered?

Each deliverable will be aimed to finish a week or more out from the deadline. This buffer zone will allow for changes and account for problems we may face.

Conclusion

In conclusion, the app we will be building, will be a financial/budgeting app. It will provide functionality for keeping track of income and expenses and setting goals. The development team is a group of 4 people who will all be working on the project functions together. We will all take part in planning and designing the app in the planning phase. We will add functions incrementally to the app, during the development phase. We are going to build the project in c++and build the gui using XCode, using objective c++. We will ensure quality in the source code and product by formal inspection of the code, good test cases, component testing and system testing. We have estimated that the overall project will take X hours to make. We expect to use this software every week because as students, we need to keep track of our income and spending to make sure we have enough money to pay rent etc. We would like to be notified of upcoming payments, calculate funds, and set a goal of an amount to save.