Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4317: Senior Design II
SPRING 2020

Team 5
The Budgeteers

Roshan Shrestha
Bobby Kemp
Emily Knowles
Anil Karki
Hedges Omordia

Revision History

Revision	Date	Author(s)	Description
0.1	9.30.2019	RS, BK, EK, AK, HO	Version I
0.2	12.11.2019	RS, BK, EK, AK, HO	Version II
0.3	04.27.2020	RS, BK, EK, AK, HO	Version III

Contents

List of Figures

1 Vision

Our team's vision is to design, develop, and test a web-based budgeting application using various 'web languages' (e.g. JavaScript, HTML, CSS, etc.) that includes an interface to a banking API assisting an individual in managing his/her budget in the long and short term. The application will also include an application back-end to help manager users and transactions associated with users as well as communicating with the banking API. Moreover, we aim to achieve these goals while providing a suitable amount of user and data confidentiality, integrity, and accessibility. With the development of the application, the application vision broadened and goals were more visible. The application is visioned to provide knowledge about handling finances, problem solving skills and budget planning skills.

2 Mission

We strive to acquire proficient use of banking API and web development tools to provide better user interface and data security. Eventually, we plan to grow from web application to mobile platforms in order to be versatile and attainable to all potential users. We will also strive to provide/suggest users to budgeting platform or options along with creating a specific budget plan which has a long term or a short term scenario. Our features on the application will also help the users change their perspective towards hectic financial management and prove how finance can be better managed in simple steps across the application.

3 Success Criteria

Within first delivery of the web application, we expect the following features would indicate the perfectness of the program:

- Login screen and layout screen for application features.
- Feature to enter personal and expenses data.
- Graphical representation for data analysis.

Within second delivery of the web/desktop application, we expect the following features would indicate the perfectness of the program:

- Message notification and alerts.
- Multiple budget strategy feature (demo).
- Categorizing transactions.
- Connection with banking information.

Within final delivery of the web/desktop application, we expect the following features would indicate the perfectness of the program:

- Security in the context of banking and personal details.
- Full access to multiple budget strategy features.
- Greater data representation for data analysis and pattern view.

- Application features such as Budget and Goals.
- Demo feature for application feature demonstration.
- Complete interactive web-page with accessible dashboard and functions.

4 Background

The app market is over-saturated with budgeting apps, however articles have been written on how they don't help people save money. [2] We wish to change the relationship that people have with their budgeting apps, and with their money. On its own, the concept of budgeting apps is simple: let users generate a budget, and send them notifications on how they're doing. Fundamentally, the apps on the market don't help users set realistic or sensible budgeting goals, and they don't hold users accountable. In an article by CNBC, the interviewed investment advisor stressed the difficulty that young adults have when trying to budget around their student debt. [1] The final problem we have with traditional budgeting apps is the lack of ability for users to experiment with budgeting styles, if an individual wished to try different app styles, they would have to download five separate apps and transfer information between them.

We hope to change how budgeting apps are viewed. Another large goal of ours is making users feel safe, a survey reported that users felt wary of Mint, due to being unsure of what Mint was doing with their banking information. [5]. We plan to be totally transparent with where their information is being used. We want to make it easy for users to try different budgeting styles and provide reference or guides for the user to build their budget. We strive to make budgeting easy for everyone and teach our users how to maintain their financial health. We hope that through use of our app we can recommend healthy financial goals for our users and motivate them to keep themselves accountable for their spending habits.

Currently our team has no relationship with our sponsor, however we hope to work closely with them in order to provide sound financial advice to our users. We also believe that our sponsor can provide an expert view on what is wrong with budgeting apps already on the market and what features we need to implement to make our app the best of the best.

5 Related Work

The leading budgeting app on the market is Mint by Intuit. Mint allows users to link bank accounts, credit lines, and PayPal accounts. Mint offers several key features, including the ability to track incoming and out-going money, the automatic updating and categorization of new transactions, and the generation of a free credit score [3] [7]. Tracking user expenditure and income is something that we plan to implement in our budgeting app, as the team feels that this is something that even the most basic budgeting apps should be able to do. Intelligent categorization is something Mint has also inspired us to include in our app. We will strive to allow the user to have as much customization of these categories as possible, and at the same time attempt to make the app smart enough to make these categorizations on its own.

Another very popular budgeting app on the market today is You Need a Budget (YNAB). This budgeting software, while popular and fairly robust, is well-known for the hefty price tag that comes with it: using YNAB will cost $83.99 per year (or $6.99 per month) [3] [7]. Unlike YNAB, our budgeting app will be free-to-use, like Mint. Potentially, depending upon what the client of the application ultimately desires, some form of in-app advertising may be used later on in order to generate some form of revenue from the application.

Lastly, another major contender for top budgeting app is PocketGuard [4]. PocketGuard takes a somewhat more simplified approach than the other applications mentioned so far. PocketGuard puts less emphasis on budget planning, and more emphasis on how much money is available to spend "right now" with the "in my pocket" number available on the app's Overview page [7]. The team took a few weeks to try PocketGuard for themselves, and ended up liking it greatly for the simplicity it provided. Therefore, one of our major design goals for our own application will be to keep the user-facing view as simple and intuitive as possible, while also trying to go above and beyond just the "here-and-now" like PocketGuard focuses on.

One thing that users dislike about Mint is the inability to choose other budgeting options, as well as the inability to allocate their budget before income has come in for the month. Our sponsor is interested in changing how budgeting apps work; we plan on allocating money for bills and necessities first, so that users can budget more responsibly and not end up overspending. [6] Other leading budgeting apps allow users more flexibility in budgeting styles but can cost over 80 dollars a year. The current budgeting app market doesn't allow for users to fully customize their budgeting styles without sacrificing a portion of their budget to paying for a subscription to the budgeting service.

6 System Overview

The budgeting application will be composed of roughly four main components/sections. The first section is the means by which the user will access the application, being either a web browser or an operating system (for a desktop application). The second main section is the budgeting app itself, running on either a remote server and accessed via a web browser, or running locally on the user's machine. Third, the application back end runs on a remote server and will be used to store user data and authenticate users with the fourth and final major component, the Banking API. These components and their relationships to one another are shown in Figure 1 below.

The basics of a budgeting app starts with the budgeting style, there are five main budgeting strategies that are utilized by budget savvy people these strategies are: subtraction budgeting, cash budgeting, proportional budgeting, two-bank budgeting, and automatic budgeting. Our first solution to the problem is to allow users to pick between these budgeting styles in our app, removing the problem of having to get multiple apps or reenter their financial data. The spotlight will be on the proportional budgeting strategy which will allow the user to have a basic knowledge on managing their budget.

Because one of the main problems with currently popular budgeting apps is rigidity in budgeting structure, our application will ideally allow for users to dynamically switch between different budgeting styles. The application will provide a host of various pre-built budgets based on popular budgeting strategies. It will also provide budgets that are more bare-bones, allowing the user to customize them from the ground up.This feature will be available in the budget tab once the user has created an account. All of these budgeting styles will be able to be edited and tweaked by the user to ensure maximum flexibility and satisfaction. Any budget created will be related to a specific goal that the user will create.

A major design goal of the app is the emphasis of simplicity. Therefore, menus, home screens, etc. will ideally be laid out as sparsely and simply as possible. Budgeting can be a very overwhelming task that is intimidating for people to delve into, and we seek to mitigate this intimidation by providing a clean, simple interface to allow people to manage their money. All the features/tab will have the same base layout throughout the application.

We plan for our system to contain a variety of pictorial representations of of users budget such as graphs and charts to help users visually understand the financial situation, while not feeling overwhelmed or swallowed in complex charts and pictorials. Therefore, graphs, charts, and other visual aspects of the applications will be minimalistic, yet informative. We plan for our home screen to be customizable so that users can choose the best visual for easily determining the state of their finances at a glance.

The application will include a demo feature that the user will be able to access without creating an account. This feature will allow the user to have a vision/perspective of how the application components are interconnected and how the overall applicaiton works.

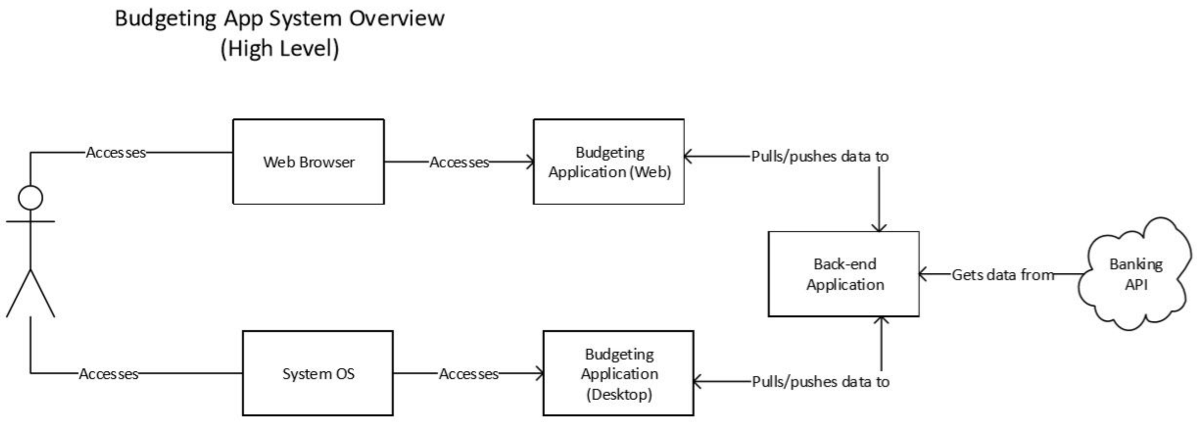

Figure 1: System Overview diagram

7 Roles & Responsibilities

Who are the stake holder of the project?

The stakeholders of the project are parties with any interest in our project outcome.These stakeholders are the members of the team (Roshan Shrestha,Anil Karki, Bobby Kemp, Emily Knowles, Hedges Omordia),the project sponsor (in this case The University of Texas at Arlington), the project customers (whoever makes use of the application when we finish), the project manager (the professor - Shawn Geiser) and the review team.

Who will be the point of contact with the sponsor?

The point of contact for our project is the same as our project manager, Professor Shawn Geiser. He will be the person that we approach for information or assistance on a specific topic and problems.

Who are the team members and what areas of responsibilities?

The team members are Roshan Shrestha,Anil Karki, Bobby Kemp, Emily Knowles, Hedges Omordia. Because our project is web-based we split into two groups, front-end and back-end. Based on our experience we sorted ourselves according to where we were comfortable working, with Hedges being the leader of the front-end group and Bobby being the leader of the back-end group. Except from group specific tasks, we as the team members will strive to share equal responsibility in everything we do. During Senior Desing II, upon agreement, Hedges Omordia, Roshan Shrestha and Anil Karki worked on developing the front-end. While, Bobby Kemp and Emily Knowles worked on developing the back-end of the applicaiton.

Will your team maintain the product owner and scrum master for the whole project, or will that change periodically?

We decided to periodically change the scrum master among ourselves, interchanging with each team member for a given amount of time. This way all team member participate in being scrum master at least once. This was also followed during Senior Desing II, where every team member got a chance to be the scrum master and present the updates.

8 Cost Proposal

The project is sponsored by CSE department of UT Arlington. The total fund available for the project is eight hundred dollars. However, team must show the proof of valid expenses to get the money. During the Senor Design II team discussion, we will not be using a host that would require a payment. This will be true for the Banking API, as free version of the API will be usedd.

8.1 Preliminary Budget

Topic	Expected expenses(dollars)
Banking API	0/month (free verison)

Table 1: Topics of expected expenses

8.2 Current & Pending Support

The only funding source available for the team right now is the fund provided by CSE department of UT Arlington. Currently, there is not any pending support for the project. The approximate fund available to the team is eight hundred dollars. Since it is a complete software project and we will not be using any paid version of the software, the funding will be in minimal use.

9 Facilities & Equipment

In terms of development tools that the team will use, the main facilities to be used for this project include the team's home computing environments (standard desktops and laptops). For version control, the team will utilize the GitLab hosting service for all remote code repository requirements.

For facility and equipment needs of the application itself, it will also make use of a Banking API, which will be a remotely-hosted service that the application will make requests to and receive data from. Potentially, the application will eventually be hosted remotely. This will likely be done on a form of hosting service such as Amazon Web Services (AWS) cloud-based web application hosting, or Heroku, another web application hosting platform that several group members have utilized successfully in the past. During the Senior Design II, the team as agreed on using Heroku for hosting the application. Typically, these web-hosting services require payment of a fee in order to keep a website up and running. Free-tier options are commonly available, but often these free-tier services do not offer various "extended" services or security features, such as SSL encryption for web traffic. Therefore, it is likely that the team will have to pay for hosting. With that being said, we will be using a free version of the web-hosting application which will lead us not use any budget amount.

The expenses for this will come out of the team's budget. If the application will be hosted remotely, depending upon the client's requirements or wishes, it may also be necessary to purchase a domain name under which to host the application. This may also invoke a fee, which will also be withdrawn from the team's budget. The domain name will be chosen by the team and/or with supervision from the client.

Campus facilities were not used in the course of developing the application. Since this was a complete software project and sudden occurrence or COIVID-19, CSE Senior Design lab space was not used. Instead, there were online meetings held for every sprint presentation, review and team meeting.

10 Assumptions

The following list contains critical assumptions related to the implementation and testing of the project.

- A suitable API will be found and that API will be readily available and able to connect to various bank accounts
- The customer will provide good documentation, and will establish clear expectations for the functionality of the application
- High-level libraries/frameworks will be available to help in completing some of the more rigorous aspects of the design, such as graph and chart generation
- The project will target one of the following, but not a combination: Android, iOS, or web
- For whichever build platform targeted, a testing library/framework will be readily available
- There will be no use of any hardware devices for project completion.
- Application functionalities will not depend on browser capabilities.

11 Constraints

The following list contains key constraints related to the implementation and testing of the project.

- Any data relayed via a banking API must be stored and transferred securely, to the best possible effort

- The final prototype for demonstration must be completed by the end of the final sprint
- The final prototype must meet all design requirements, as closely as possible, as specified by the customer
- Total development costs must not exceed $800
- A back-end application will need to be written and configured to support authentication and other functionality
- The application database should be able to hold user data including transaction and personal details.

12 Risks

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

Risk description	**Probability**	**Loss (days)**	**Exposure (days)**
High-level libraries are not applicable to our project	0.25	15	3.75
Project features implementation and API security on banking information	0.25	15	3.75
Documentation is unclear/requires much deliberation	0.20	8	1.6
Availability of API access due to downtime	0.15	15	2.25
A testing framework for the project is not readily available	0.10	5	0.5
Project prototype must target multiple platforms	0.05	10	0.5

Table 2: Overview of highest exposure project risks

13 Documentation & Reporting

13.1 Major Documentation Deliverables

13.1.1 Project Charter

The project charter will be edited under a team account on Overleaf.com. It will receive revisions at any point that revisions are necessary in order to keep it in line with the current scope/goals of the project. The initial version will be delivered at the end of the first Senior Design I sprint (September 30th 2019). The final version will be delivered near the end of Senior Design II (May 15th 2020).

13.1.2 System Requirements Specification

The System Requirements Specification (SRS) will be edited under a team account on Overleaf.com. It will receive revisions only when the team decides to make major changes to the design of the system. The initial version will be delivered at the end of the second Senior Design I sprint (October 21st), and the final version will be delivered near the end of Senior Design II (May 15th 2020).

13.1.3 Architectural Design Specification

The Architectural Design Specification (ADS) will be edited under a team account on Overleaf.com and under a team account at draw.io. It will receive revisions only when the team decides to make major changes to the design of the system. The initial version will be delivered at the end of the third Senior Design I sprint (November 11th), and the final version will be delivered near the end of Senior Design II (May 15th 2020).

13.1.4 Detailed Design Specification

The Detailed Design Specification will be maintained by hand by the group and/or on an appropriate cloud-based design software/website. It will be updated according to the team's discretion, but most notably upon any decision that is made about a key design component. The initial version will be delivered at the end of the fifth sprint, in Senior Design II (February 24th). The final version will be delivered near the end of Senior Design II (May 15th 2020).

13.2 Recurring Sprint Items

13.2.1 Product Backlog

Items were be added to the product backlog as they're broken down in the SRS. Items were generated from the features and functionality required. Items will be prioritized based off our sponsor and group vote to what is most important. We utilized a free software like iceScrum to track our backlog progress. Most discussion were done thorugh an online GroupMe application due to social distancing.

13.2.2 Sprint Planning

We had planned each sprint at the end of the previous sprint. We had analyzed our progress and see where we need to spend more time. We plan on having 8 sprints through the course of the project. Half of the sprints occured in Senior Design II.

13.2.3 Sprint Goal

Our sprint goal was determined by our team, however if we cannot come to an agreement we will defer to our team lead: Roshan. We had proper and regular communication with our professor during this process to ensure we are staying true to what the project demanded.

13.2.4 Sprint Backlog

Items were added to the sprint backlog from the product backlog. Items were added in accordance to the Sprint Goal. We had used the same tool as our product backlog to stay consistent.

13.2.5 Task Breakdown

Tasks were chosen by teammates on what they feel comfortable doing and time spent were be documented in our engineering notebooks.

13.2.6 Sprint Burn Down Charts

One teammate was in charge of generating the burn down charts per sprint, they achieved this by collected everyone's estimated time worked each week. We had used an agile sprint burn down chart

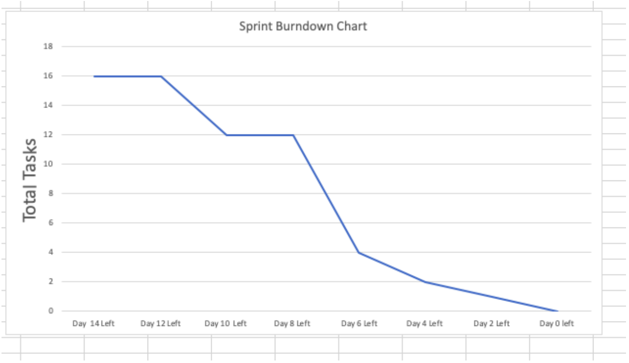

Figure 2: Sprint burn down chart

13.2.7 Sprint Retrospective

The team retrospective was handled by the scrum master and we reconvened every Monday at 12pm. Progress was documented and information was shared among each teammate and development was logged into our engineering notebooks with GroupMe being our main means of communication.

13.2.8 Individual Status Reports

Individual statuses were determined by the task was assigned to that team member. Individual progress was reported every Wednesdays but frequent updates are encouraged for any development that is deemed important.

13.2.9 Engineering Notebooks

The engineering notebook was be updated every sprint or in the case of any new ideas being generated. The ideas were be put down in the engineering notebook with the respective date,time and signature of witness if needed. The use of page can varies among the team members during every sprint.

13.3 Closeout Materials

13.3.1 System Prototype

The final system prototype will be a functioning and reasonably well-tested application that meets, at least, the minimum specified design requirements. This will be demonstrated to all relevant stakeholders, including the customer via a video-demonstration meeting, that will display the app's full spectrum of functionality. Before this meeting, acceptance testing/demonstrations will be held, as often as possible as determined by the development team and customer, involving the customer in order to ensure the team is on the "right track" (agile). Due to the COVID-19, there will not be any in person demonstration of the application, instead we will be preparing a video demonstration.

13.3.2 Project Poster

The project poster will contain the major feature of the app as well as guideline for installation process. It's size will approximately in the range of 30-40 inches of width and 40-50 inches of length depending upon the requirement of space. It will delivered after the completion of project(SD2 final presentation day).

13.3.3 Web Page

The information on web page will be similar to the one in project poster but will be in detailed. It will be a public domain. Our aim is to start and update it through out the project. Our aim has not changed through out the Senior Design II.

13.3.4 Demo Video

The demo video will be on the guideline to install the app as well as on the major feature. It will be approximately 2-5 minutes and cover up application functionalities.

13.3.5 Source Code

The source code will be maintained using the GitHub. The customer will not have access to the source code so, only binaries is provided but both will be provided to Dr.Geiser at the end of the project.

13.3.6 Source Code Documentation

This project will be done using java programming language. So, we will be using GITHUB to store, exchange and update necessary codes. Through out Senior Design II, GITHUB was mostly used to push or pull any updates on codes by members.

13.3.7 Hardware Schematics

This project is solely a software project. There were no additional hardware components added during Senior Design II.

13.3.8 Installation Scripts

No script installations will be required since the project will only have a web-based application. The user will only require to access the application and enter the required details. Set up details will be available in the README of project file.

13.3.9 User Manual

A video user manual will be provided to the user.

References

[1] Evan Kuh. Budgeting apps have major flaws when it comes to helping users actually save, Jun 2019.

[2] Kelley Long. 5 reasons budgeting apps don't work for most people, Oct 2017.

[3] A O'Shea and L Schwahn. The 7 best budget apps and personal finance tools of 2019, Mar 2019.

[4] Eric Rosenberg. The 8 best budgeting apps of 2019, Nov 2019.

[5] Anne Sraders. Is mint safe? what to know about the budgeting app in 2019, Dec 2018.

[6] Steward. Why we're breaking up with mint, Dec 2017.

[7] Kristin Stoller. The best budgeting apps in 2019, Dec 2019.