

Team 13 Project Proposal

Team Number: 13

Team Members: Aaron Harpt, Peyton Doherty, Alec Horlick-Mills, Tyler Rains,
Tucker Wienke

Project Name: Musa

Project Budget: \$0

Project Synopsis:

Finance budgeting web application which allows users to track their expenses, income, investments, assets, and manage a monthly budget.

Project Description:

Budgets are powerful tools that anyone can utilize to help accomplish their financial goals. Having a useful application which makes it easier for people to manage their finances helps everyone. As big budget users ourselves, undertaking this project allows for us to make the perfectly fine-tuned budget application. This is a project that we will actually use everyday and be able to use even beyond this class. We are all going to be moving into our careers after our senior year, so our application will be very useful to help us manage our money and best set ourselves up for the future. By having a budgeting application that is easily accessible, we can make sure we are not overspending and can effectively save money.

Musa addresses our needs by being a useful budgeting application designed specifically to how we want it. It also addresses a great opportunity as we feel other

budgeting applications, especially web applications, do not organize and provide the features exactly how we would want it. This gives Musa an opportunity to fill this void and offer a better experience than current budgeting applications. The end result of the project will be a fully functional budget application. The user will be able to set budgets for different categories, track expenses and income, and view investments and assets. It will be created as a web application that users will be able to access online on any operating system.

Project Milestones:

First Semester

1. Decided project technologies (9/29 - 10/27)
2. Project Proposal and video (10/13 - 11/1)
3. Created rough sketch of the frontend (11/1 - 11/17)
4. Planned out backend/database/hosting (11/17 - 12/1)
5. Completed setup of JIRA, github repository, and coding standards document (10/20 - 12/10)

Second Semester

1. Finalized design document (1/19 - 2/2)
2. Built out frontend functionality (1/19 - 4/13)
3. Built out backend/database functionality (1/19 - 4/13)
4. Frontend and backend are fully connected, tested, and completed (4/13 - 4/27)
5. Finalized project video and quad chart (4/20 - 5/6)

GANTT Chart is on the next page

Task Name	Assigned To	Oct					Nov					Dec			
		Se	Oct 3	Oct 10	Oct 17	Oct 24	Oct 31	Nov 7	Nov 14	Nov 21	Nov 28	Dec 5	Dec 12		
1 First Semester															
2 Decided Project Technologies	Full Team														
3 Project proposal video	Full Team														
4 Completed project management setup	Full Team														
5 Created rough sketch of the frontend	Full Team														
6 Planned out backend/database	Full Team														
7															

Task Name	Assigned To	Jan					Feb					Mar					Apr				
		Jan 2	Jan 9	Jan 16	Jan 23	Jan 30	Feb 6	Feb 13	Feb 20	Feb 27	Mar 6	Mar 13	Mar 20	Mar 27	Apr 3	Apr 10	Apr 17	Apr 24	May 1		
1 First Semester																					
2 Decided Project Technologies	Full Team																				
3 Project proposal video	Full Team																				
4 Completed project management setup	Full Team																				
5 Created rough sketch of the frontend	Full Team																				
6 Planned out backend/database	Full Team																				
7																					
8 Second Semester																					
9 Finalized design document	Full Team																				
10 Built out frontend functionality	Peyton, Tucker																				
11 Built out backend/database functionality	Alec, Aaron, Tyler																				
12 Frontend and backend are Completed	Full Team																				
13 Finalized project video and quad chart	Full Team																				

Preliminary Project Design:

How the software works

There will be a few main UI elements for our users to view. The main page for the user will be the summary page, with a summary of the overall budget, any current balances and balances of the previous month, and any income and spending that the user did within the current month. Data from the other pages are pulled into this one to give the full summary. Another page the user can view is their entire budget. This page displays the monthly budget for every category the user sets, any expenses that were spent on each category, and any income that the user might be making. The next page

consists entirely of the user's income, whether it be investments, transfers, or job income.

To build our UI in the front-end, we have decided to use React JS. We chose React for its reusability and its quick learning curve. The reusability will be useful for our budgeting web app because the various pages of the budget use very similar designs and components. One of our team members has experience using React JS and found it straightforward, and easy to learn, with useful tools to help bring the best in our design. We also are going to use Bootstrap, which is a free, open-source, front end framework that helps develop faster, easier, and more responsive web pages, and is also compatible with all modern web browsers, such as Firefox, Chrome, Safari, and more. It contains various HTML and CSS templates, with optional JavaScript plugins that makes developing easier.

The other half of our project is the back-end component. This component is responsible for running everything such as, computing the math, running the server, and helping the front end react to users' inputs. For the back end, we decided to use Node JS and its application framework, Express JS. We chose Node and Express because they are both very useful tools for creating web applications. Express JS is seen as the de facto standard server framework for Node Js and is designed for building web applications. Like how Bootstrap made HTML and CSS easier to code. Express makes Node JS easier to code, and gives it additional features to extend our server-side coding.

We also will use a database in order to keep all of the information that the user will be inputting organized and easy to access. For our database, we chose to use MySQL. We chose MySQL due to the fact that it is fast, easy to use and learn, very reliable, and is a very good tool for web applications. It is also a pure relational database,

which organizes the data into tables that can each be accessed with a specific key, making it easy to get the information that we need. MySQL works better with web applications that have a simple structure, making it the best choice for our project. Finally, as students at KU, we also have access to the MySQL servers provided by the university, which will allow us to keep our data protected.

Finally, we also are going to use AWS's cloud computing platforms. AWS is a great platform because it's easy to use, offers many different services, and is compatible with all of the technologies that we are using. It will provide a great way for us to store our data and it provides database options for MySQL.

Design constraints

There are two types of design constraints, both of which can heavily affect the design of our project. Those two types are technical and business. For the technical constraints of Musa, the first issue to contend with is that it must have the basic attributes of a budgeting application. That means that it must have a way to track income, expenses, and plan a budget. This fundamental description of what the application will actually do as its core functionality is key to understanding what other decisions need to be made. Every other decision can come after the basics of what our budget application, Musa, fundamentally is.

The second technical constraint is that Musa must be a web application. We wanted it to be a web application since that would make it easier to access from multiple different platforms unlike a computer application. For a computer application, which creates an executable application, it requires much more specification on the type of operating system and other concerns at play. We also didn't want to make it on an app

store like the Windows Store or the Apple App Store. That similarly constrains it too much. Because of this, we decided that our design constraint would be in making it a web application. Obviously, this constraint drives the technology more than almost any other constraint. It was the driving decision behind why we chose JavaScript, since it is widely used to create web applications.

A third technical design constraint, which follows from the last, is the programming language. We have decided to use JavaScript as our fundamental language, and because of that, all of the following technology needs to work well with JS. That means the backend, database, hosting platform, etc... all need to synchronize well with JS. This requirement led to the adopting of technology such as Node JS and Express JS as the backend of our choice. These two technologies sync very effectively with JavaScript.

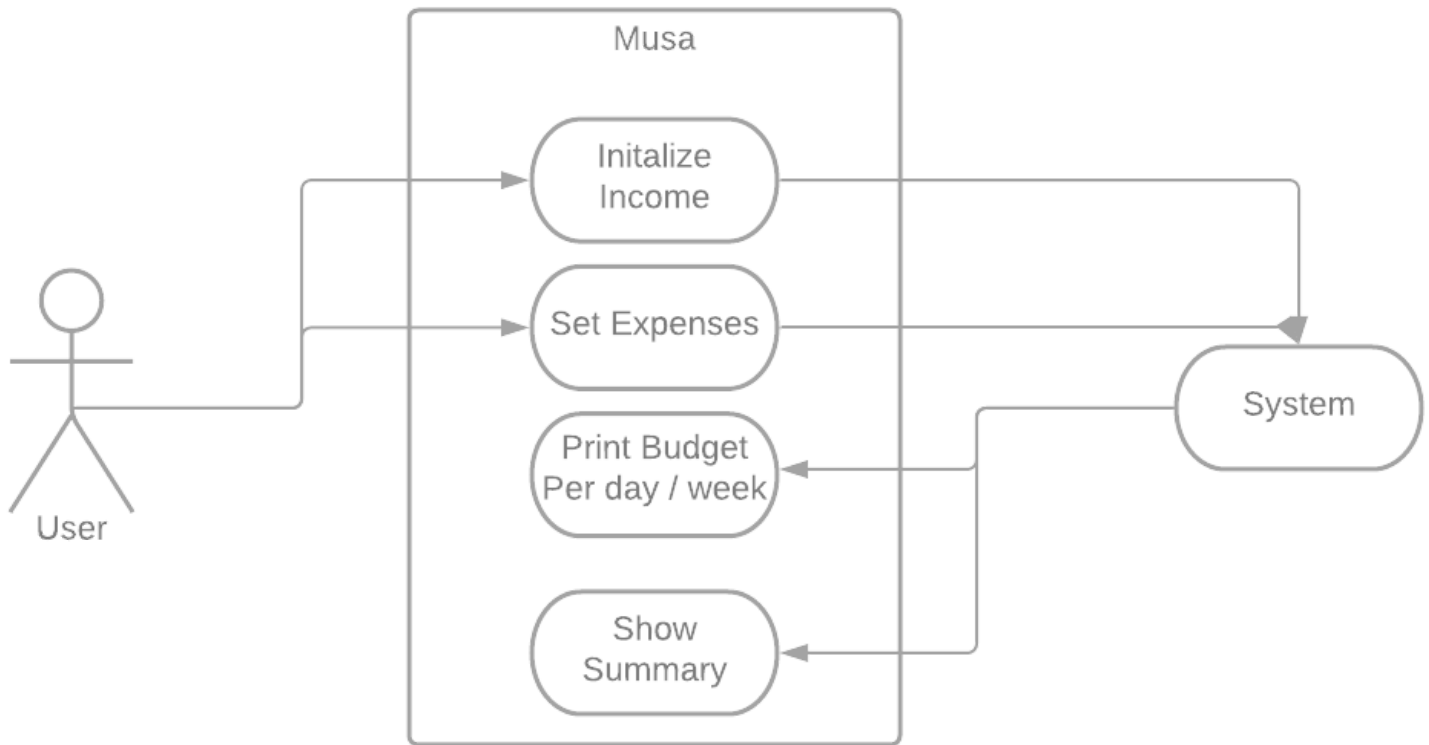
A fourth technical constraint is that it must be mobile friendly. We want this budget app to be able to be used on someone's phone through the website. This means that the frontend design needs to be reactive. To make it reactive, certain design decisions will have to be taken to accommodate this need. For example, this constraint drove the idea behind using Bootstrap with JS. Bootstrap allows for a better developing experience when making a reactive frontend, which is what is needed in this case.

The first business design constraint is the schedule. This project being part of a year long class means that there are specific time restrictions that have to be followed. For example, the application needs to be finished by the end of next Spring. This constraint decides how the entire schedule is constructed and when the milestones are placed. There are other smaller required milestones that are spread throughout the class

as well which present themselves as a constraint that has to be followed. One example of those is this very document, which has a set date where it must be completed.

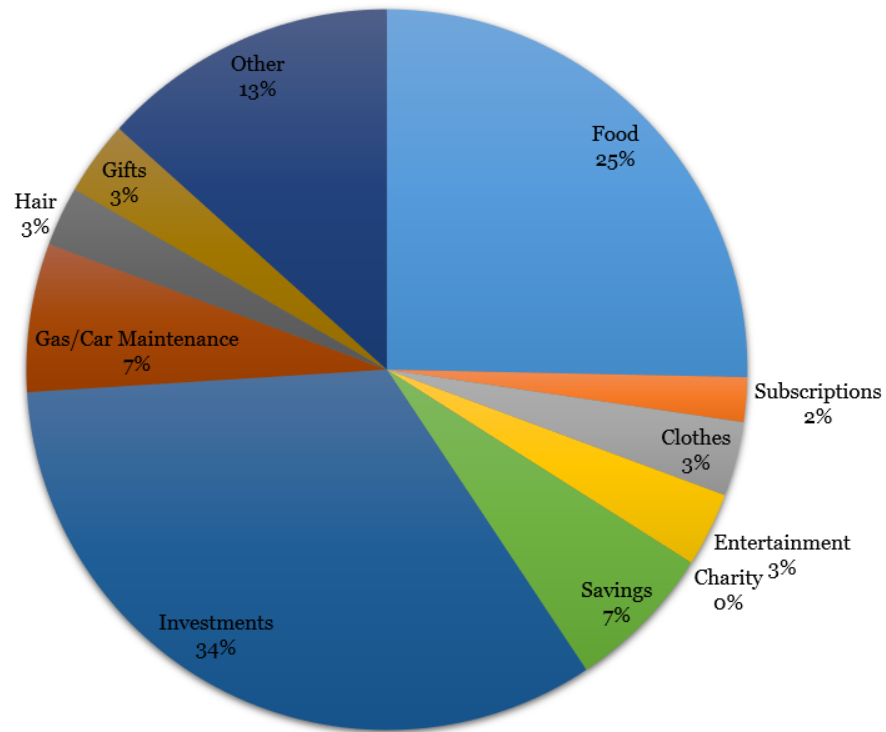
A second business constraint is the team composition and make-up. Four of our five-man team picked each other to be teammates while the fifth person was randomly selected. Also, the team is not allowed to grow or shrink which means there is very little flexibility when it comes to this important aspect. Because of this constraint, each team member really needs to make sure that they are on the same page and completing their required work in order for the project to go smoothly. There is a third business constraint that is typical but doesn't actually constrain us much for this specific project. That constraint is the budget. We decided to go with a budget of \$0 from the school, as we didn't want the school to potentially own any part of the project. Because of this, we aren't constrained much in the budget department, and are only constrained with how much we personally want to spend. There being five of us, we can split costs between us and keep it relatively cheap, so we aren't as concerned with the budget constraint.

Charts/graphs/illustrations

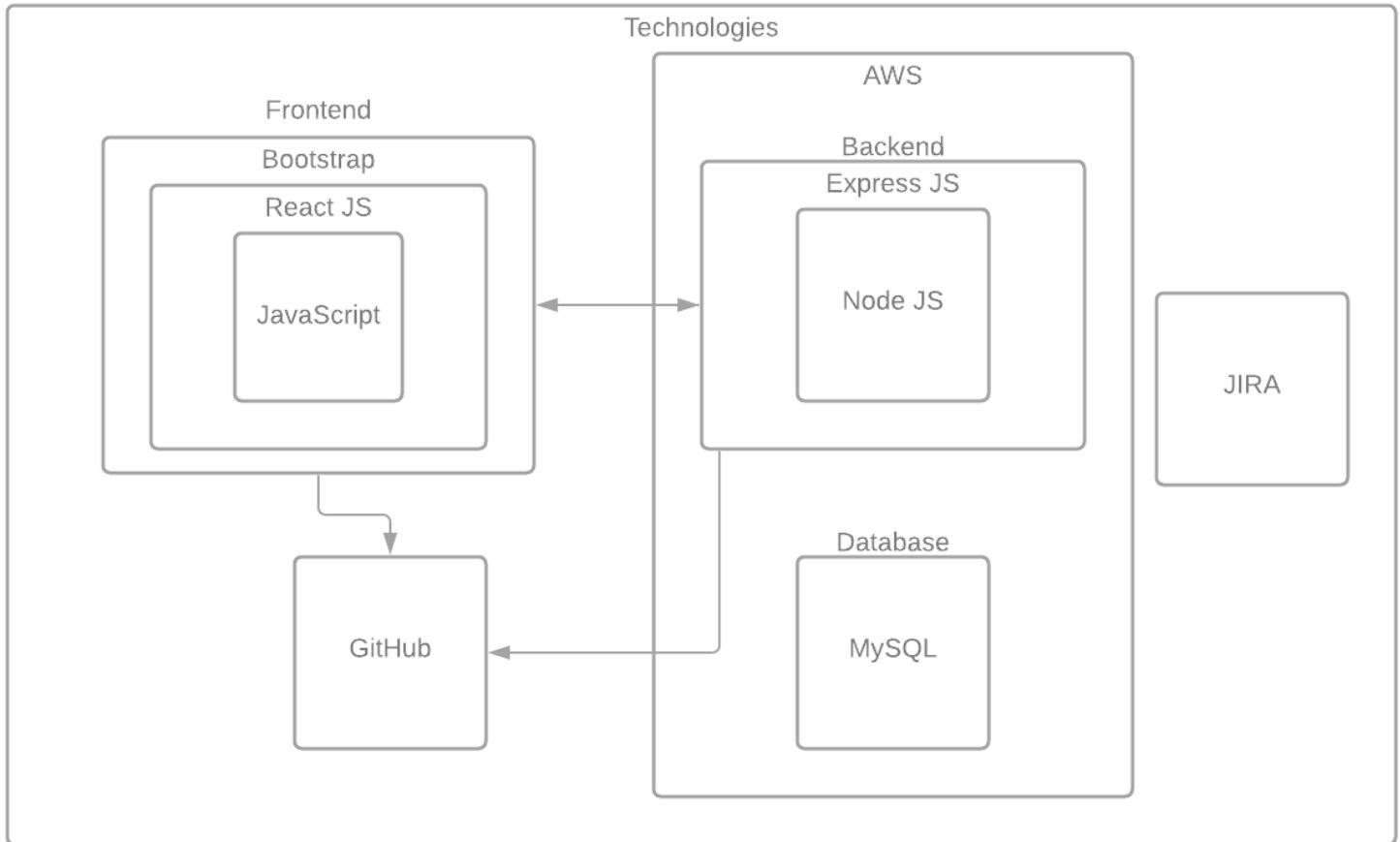


Use Case Diagram. Shows what the user will need to input and then what they will see as a result.

Monthly Allotments



Example pie chart breakdown showing what percentage of your money you have dedicated to certain expense categories from your projected total income for that month.



Our technologies include:

- Frontend: JavaScript with React JS and bootstrap to make it reactive for different devices.
- Backend: Node JS with Express JS.
- Database: MySQL
- Host: AWS will host the Backend and the Database.
- GitHub as our git repo manager and Jira as our issue tracker.

Ethical Issues:

1. Saving individual's financial information

There are two potential ethical issues in our application. With regards to the Code, it would mainly conflict with 1.7, Honor Confidentiality, and 2.9, Design and implement systems that are robustly and usably secure. Musa will handle peoples budgeting information, which to many is private information that they do not wish to be public. There will need to be a table that tracks which user contains which budgeting information. We could attach an ID to each user's account that will serve as a user's key to unlock their budget from the SQL server. This would be done completely in the background without the user knowing.

As programmers creating this web application it is our responsibility to ensure that their information will not be accidently exposed to other users. We will need to implement a security system that prevents malicious users from trying to login to accounts that are not theirs. Our login authentication system will need users to create passwords that are not easy to guess, this will mean passwords that contain uppercase characters, numbers, and special characters. The code that accepts the login information itself should just use a simple encode function to ensure the login and password are not executed as code itself when they should be interpreted as strings.

Intellectual Property Issues:

1. MIT license agreement for Software

We are planning to use both React.js and Reactstrap for the user interface of our web application. Both of these projects are licensed under the MIT license which allows us to use their source code for any personal or commercial purposes. Therefore, it is fine for us to use this source code even if our budgeting application earned revenue and became a commercial application for sale on the market. The MIT license is very lenient and allows us to do basically anything with the code it licenses except two things. First, we may not hold the creators of the source code liable in any way. Therefore, in a scenario where there was some vulnerability in our application that came back to the third party's source code, we may not hold the creators liable in any way. By using their code, we agree to not hold them responsible for any issues we encounter as a result of using their code. Secondly, we may not remove the licensing or copyright notice that the creators of the source code have put inside of their respective source code files.

2. Project Name:

For the name of our budgeting application, we have decided on "Musa", after a man named Mansa Musa who was one of the richest men who ever lived. Even though we did come up with this name on our own, we must still ensure that we are not stealing this name from a competing budgeting application or other company that is extremely similar to a budgeting application. To do so could be an intellectual property concern since a company name could be legally protected through a trademark. Even if a company name is not legally protected in any way, using the same name could appear to consumers that we are simply leveraging off of someone else's name and reputation or that we are affiliated with another company who picked the same name as we did.

Change Log:

1. Project Milestones

A few of the project milestones were updated to better reflect the progress that was made to make it more accurate. All of the changes had to do with the first semester. The initial technologies decision date was moved to through 10/27, the completion of JIRA, GitHub, and the Coding Standards document was moved to finishing by 12/10, the backend was moved up to 12/1, and creating a rough sketch of the frontend was moved up to 11/17.