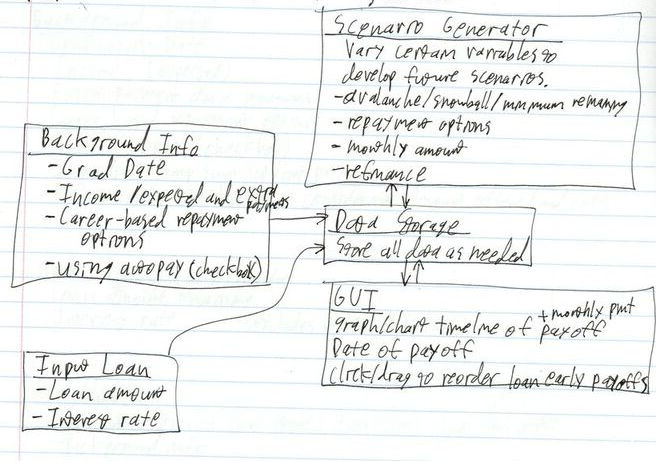
# Project Description

This senior design project will be a student loan calculator. Students can enter their financial information into a web app and it will calculate their payoff dates, total of payments, and give advice on which repayment scenarios to choose. Today’s college graduates have significantly more student loans than past graduates, and economists are predicting that student loan debt will slow down the US economy in the future. Many college grads do not understand how to calculate loan amortizations and cost/benefit analyses. There are currently no online, high quality, free calculators dedicated to student loans. My senior design project will fill this need.

# System Design Diagrams – Level I



# System Design Diagrams – Level IV

Input Loan

1. Loan amount remaining
2. Interest rate (excludes autopay deduction)

Background Information

1. Graduation date
2. Income (expected)
3. Extra monthly payment above minimums
4. Career based repayment options or forgiveness after a certain time
5. Using autopay (checkbox)
6. Refinance option the user is considering. Amount offered and interest rate.

Scenario Generator

1. Based on the starting information provided, try different possible scenarios.
2. Optimize in this order:
   1. Total of payments (minimize)
   2. Date finished with all loans (earlier)
   3. Risk of income loss (good = decreasing or constant payments. bad = payments increase over time)
3. Variables to experiment with:
   1. Total monthly payment
      1. Minimum payments only, based on repayment plan
      2. Extra payment that the user inputs with the starting information
      3. Extra payment that the user inputs with the starting information PLUS $100 per month.
      4. Custom changing amounts. These are input using the sliders on the graph.
   2. Which loan to focus on first/next:
      1. Snowball – pay the lowest minimum first. Explain that this is the not the least expensive, but it is the easiest to maintain psychologically.
      2. Avalance – pay the highest minimum first. Explain that this is the least risky, but the most expensive and hard to maintain psychologically.
      3. Highest rate – pay the highest interest rate first. Explain that this is the cheapest, but hardest to maintain psychologically.
   3. Refinance options:
      1. No refinance. Explain that this is less risky, but more expensive.
      2. Refinance. Explain that they cannot use deferment or other federal options. Note that this option is only visible if the user has input a refinance option.
   4. Alternative repayment options (career based, ect):
      1. Normal 10 year plan
      2. IBR
      3. PAYE
      4. REPAYE
      5. Forgiveness after a certain time period. Typically based on career. The user would input this option in the Background Info section if it is available to them.

Data Handler

1. Large data struct that stores everything that was input:
   1. Background info
   2. Loan(s)
   3. Monthly payments from interactive graph
   4. Data for scenarios
2. After any new data is input, trigger the outputs to recalculate. Handle the process/threads.

GUI

1. Display the options for the scenario generator. Under each of these options, explain what the option means. Use links to financial aid websites, Wikipedia, etc.
2. Display an amortization graph timeline. Shows each loan as a line which decreases over time.
   1. Bold lines indicate when more than the minimum is being paid toward that loan.
   2. Total combined loan amount is also graphed.
   3. Hovering over a point displays the name of that loan, the year and month, and the exact dollar amount at that point.
   4. Y axis is dollars which runs from zero to the total combined loan amount plus $1000.
   5. X axis is time which runs from the present time to 10 years and 4 months from the present time. Smallest point increments are one month, and graph axis is labeled in years.
3. Display a monthly payment graph timeline.
   1. Parallel to the other graph, below it. Dates match vertically.
   2. Points are the payment for that month, and they are connected by a line.
   3. Thin dashed line shows the minimum payment.
   4. Clicking a point shows a popover that allows the user to change the total payment for that month.
      1. Display the year / month selected.
      2. Input payment amount
      3. Apply to all payments after year / month, or only to this payment? (checkbox)
      4. Must ensure that number input is not less than the minimum payment allowed.
   5. X axis matches the main graph above.
   6. Y axis goes from zero to triple the minimum.
4. Total of payments and date that all student loans are finished. Large bold font.

# Module Descriptions

1. Input loan -> allow user to input the mathematical properties of their loans.
2. Background Info -> allow user to input general information about themselves and their financial situation.
3. Data Storage -> data struct that stores information and handles data flow in the program.
4. Scenario Generator -> perform mathematical calculations and simulations.
5. Output GUI -> show the user the scenarios, and allow them to adjust the scenarios.

# Wireframe

See <https://wireframe.cc/iUoWRX> for the wireframe source.

# Project Task List

1. Research repayment options, deferrment, refinance, etc
2. Buy domain name(s)
3. Set up website host. Integrate with GitHub.
4. Build html form for inputting a loan
5. Build data handler
6. submit button on input loan form - event handler
7. Build html form for inputting other info
8. submit button on other info - event handler
9. Write a heuristic for scenarios
10. Write scenario generator
11. GUI to compare scenarios side by side
12. Plot amortization graph
13. Write informative text, how loans work
14. Polish website and publish it. Advertize on subreddit, etc
15. Extra time - buffer

# Timeline

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Task Name** | **Difficulty** | **Duration** | **Start** | **Finish** | **Predecessors** | **Done** |
| 1 | Research repayment options, deferrment, refinance, etc | 8 | month | now | Dec 11 | none | 80% |
| 2 | Buy domain name(s) | 1 |  | now | now | none | Done |
| 3 | Set up website host. Integrate with GitHub. | 4 |  | Nov 15 | Dec 1 | 2 | 90% |
| 4 | Build html form for inputting a loan | 7 | week | Dec 11 | Dec 18 | 3 | 10% |
| 5 | Build data handler | 5 | 3 days | Dec 17 | Dec 21 | 4 |  |
| 6 | submit button on input loan form - event handler | 4 | week | Dec 15 | Dec 25 | 4, 5 |  |
| 7 | Build html form for inputting other info | 4 | 3 days | Dec 25 | Dec 29 | 4 |  |
| 8 | submit button on other info - event handler | 3 | week | Dec 26 | Jan 4 | 4, 5, 6 |  |
| 9 | Write a heuristic for scenarios | 9 | week | Jan 1 | Jan 11 | 5 |  |
| 10 | Write scenario generator | 8 | week | Jan 5 | Jan 18 | 5, 9 |  |
| 11 | GUI to compare scenarios side by side | 7 | week+ | Jan 13 | Jan 27 | all prev |  |
| 12 | Plot amortization graph | 6 | week | Jan 23 | Feb 3 | 11 |  |
| 13 | Write informative text, how loans work | 3 | week | Feb 1 | Feb 10 | 11, 12 |  |
| 14 | Polish website and publish it. Advertize on subreddit, etc | 6 | 3 days | Feb 5 | Feb 13 | all prev |  |
| 15 | Extra time - buffer |  | month+ | Feb 13 | April 1 |  |  |

# Effort Matrix

# ABET Concerns Essay

Adam White

ABET Constraints Essay

Student Loan Calculator

11/02/2015

Economic

The student loan calculator website will only cost me a small amount of money. Purchasing the URL domain was $15 for 12 months, and hosting the website will also be inexpensive. If the website traffic increases heavily in the future, I may decide to put in a donate button linked to PayPal. On the other hand, the tools on the website will allow all college graduates in the United States to get accurate advice on their student loans. Users of the website should be able to pay off their student loans faster, which will save them money and allow them to spend their hard earned dollars on useful items instead of Sallie Mae. If enough people make wise choices about their student loans, citizens will have more money in their pocket, and loan financiers will experience very slightly decreased returns.

Ethical

Personal finances can cause a person to have negative emotions. This web tool will tell users the truth about their financial situation. Since debt is ‘negative money,’ users are likely to be unhappy about their student loan debt. However, ignorance is not bliss, and knowing their loan details will allow them to take control of their financial situation. I will provide popup text descriptions that explain why certain recommendations are made. Language which is non-threatening and non-accusatory should be used.

Legal

Since I am not directly managing other people’s money, I do not need to register with the Securities and Exchange Commission, or their equivalents in Ohio and Indiana. I should put a disclaimer text at the bottom of the website.

Security

I should not store any user data on the website – all data that a user enters should stay on the client inside their browser. All data calculations will be performed in JavaScript. No data should be returned to the server.

Social

Individuals who use my tool will be able to pay off their loans sooner. This will allow them to spend money on useful things once their loans are paid off. Student loans as a whole are forecast to become a heavy drag on the economy as many young people reduce their spending on goods and services. This student loan calculator adds benefit to society.

# Self-Assessment Essay

Adam White

Senior Design

September 20, 2015

Self-Assessment Essay

My senior design project will be a student loan calculator. Students can enter their financial information into a web app and it will calculate their payoff dates, total of payments, and give advice on which repayment scenarios to choose. Today’s college graduates have significantly more student loans than past graduates, and economists are predicting that student loan debt will slow down the US economy in the future. Many college grads do not understand how to calculate loan amortizations and cost/benefit analyses. There are currently no online, high quality, and free calculators dedicated to student loans. My senior design project will fill this need.

Senior capstone is the culmination of 5 years of school studies, and it is where we can demonstrate what we have learned. In Data Structures, I learned the ‘basics’ of programming and how to use computer languages. In Algorithms, I learned how to think and solve problems in the way that computer scientists think. Most of my current knowledge of JavaScript comes from one assignment in Artificial Intelligence. I wanted to specifically do this project because it combines both my major in computer science and my minor in economics. Personal finance has long been an interest of mine, and in 9th grade I scored in the top 1% of the National Financial Capability Challenge.

All five of my co-op terms were at Siemens PLM. During the second and third co-ops, I worked on a web app which was written in Java and compiled to JavaScript using Google Web Toolkit (GWT). Since the source code was written in Java, I did not interact with JavaScript very much during these co-ops. I learned some about UI design relating to web apps, but in my opinion the UI design was lacking, since it was built by software engineers and not graphics designers. Also on my second and third co-ops, the product I was designing used a scrum team methodology. I learned how to build a web app tailored to the customer’s needs, and how to break apart a large project into bite size pieces.

Personal finance has always been a strong interest of mine. I believe that people can experience freedom when they put effort into planning their financial future. The powers of compound interest seem nonsensical or ‘magical’ to people who don’t understand them. This hurts people by causing them to take out loans that are too large or to pay the loans back too slowly. The web app will allow users to input data about each loan that they have. It will calculate their expected repayment time and total of payments over several time periods and for several monthly payment amounts.

For the senior design class we will be using a form of the waterfall method. I’ll write out the design of the product first, including UML diagrams, wireframes, and use cases. I would like to have the design completed and be able to start coding by the beginning of December. At some point I’ll need to get the advice of a graphic designer, since I am not good at that. By the end, the product should be a fully functional and user-friendly web app that is helpful to people with student loans. I will know that I have done a good job if I am proud to show off the product during interviews.

# Professional Biography

Adam White

adam.nathaniel.white@gmail.com

513-502-9288

Co-op History

Siemens PLM.

Title: Software Developer Co-op.

5 co-op semesters, January 2013 to August 2015.

Tasks:

1. Built an iOS app to view JT files. Jt2Go is currently in the App Store.
2. Converted GWT with Java to use CSS in Active Workspace.
3. Active Workspace - Framework UI Team. Implemented new features and fixed UI bugs.
4. Wrote a utility to convert from JT format to Direct Model. Allows NX and other products to 3d print JT files directly.
5. Built an ‘advisory program’ to fix problems in 3d models before being sent to a 3d printer.

Skills and Expertise

* C++
* Java
* Python
* Objective-C for iOS

Areas of Interest

* Algorithm implementation
* Financial data processing (combine my minor in economics with my senior design project)
* Mobile development / consumer-oriented product

# Budget

Purchasing the domain names “studentloancalc.com” and “planstudentloans.com” cost me $30 for the year starting in October. This was paid to EasyDNS, located in Toronto.

Hosting is provided by Modulus. I am currently on their free trial service, which will run out during the spring semester.

Billable hours so far on the project are 48 ½ hours. At a cost of $75 per hour, this is equivalent to $3,637.