INFO 1113

S10

Flow Tracker

Pavan Shergill, Matthew Sander

100379004, 100338852

[pavan.shergill1@email.kpu.ca](mailto:pavan.shergill1@email.kpu.ca), [matthew.sander@email.kpu.ca](mailto:matthew.sander@email.kpu.ca)

November 24, 2019

<https://sites.google.com/view/sattmander/home?authuser=0>

<https://trello.com/b/iom7HemC/who-did-whatflow-tracker>

Git Hubs

<https://github.com/shergill48/testProject>

<https://github.com/msander95/project-git>

Table of Contents

Page 1 – Title Page

Page 2 – Table of Contents

Page 3 – Introduction

Page 3 – Project Requirements

Page 4 – Executive Summary

Page 5 – Use Case Diagram

Page 5- 6 – Use Case Description

Page 7 – Class Diagram

Page 8 – Sequence Diagram

Page 9 – Interface Prototype

Page 10 – Project Experience

Page 10 - Conclusion

Page 11 – Works Cited

**Introduction**

Flow Tracker is an application designed to allow people to track their work hours through input amount earned into the program. The program can be set up for a scheduled paycheck to automatically update letting the user know how much they earned and how much was taxed or used for other fees. The app will display the breakdown of the users pay and will promote budgeting for a peace of mind in financial security. Additionally, the platform for the app is mobile devices for simplicity and easy access and can help the user determine if they were paid correctly.

**Functional**

User login- user will input their email, password.

* User input variables- user must input their hours/ pay and the tax that is taken off their check.

**Non-functional**

Calculations- compute the variables that the user inputted.

* Display- show the user the final value after calculations

Business idea: **Summary**

The idea of our product is to create an easy to use application that can help the people of the world keep track of how much they make passively. Without having to do any of the calculations or hours tracking that many people are terrible at doing. In todays world time is money. And when people are working 40-hour weeks and going to school or having a family. It gets hard to keep track, while also making it easy for companies to make errors. Resulting in the loss of wages. Which short term is but long term can accumulate into a grander sum of lost wages that can go completely un-noticed? Flow tracker can end this trend.

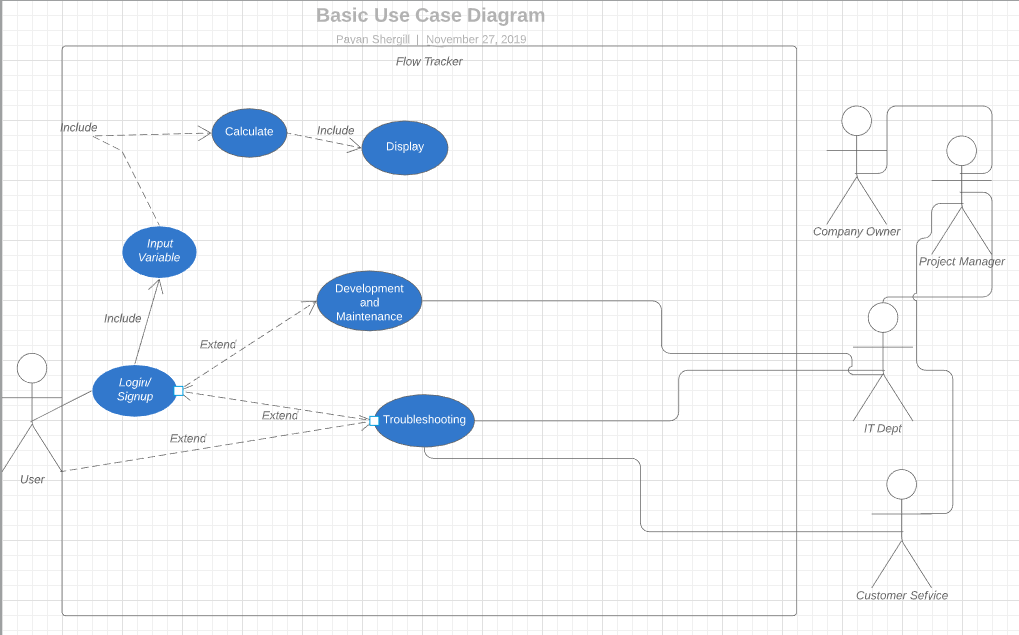
Key Points:

The main purpose and function of our application is to track hours of worked by the user. It will access the scheduled work events that are in the user’s calendar. Using these tracked periods of time the application will calculate the user’s money earned. In order to do this the application needs a few things from the user to complete the algorithm. First the user must input their hourly rate of pay, so that the application can properly track how much the user makes while at work. Then the user needs to input the rate that they are taxed in their state or province. This will allow the application to make the proper deduction of tax. This way the amount that the application presents to the user as the amount they will make on their next check is completely accurate right down to the cents.

Why choose Flow Tracker:

Flow tracker will be the best option for people who want to keep track of how much they are making. Nobody likes having to fix incorrect paycheques but having flow tracker will make it easier to catch these errors and fix them. Whether the mistake was by accident or on purpose, fixing an honest mistake or catching a time thief. This application can protect your money as well as help you plan better for the future. Knowing how much you will have when pay day comes helps people not overspend. The answer to saving money is mindfulness and that is the overall goal of our application. To help employees calculate their money and by calculating their money we plan to give people the ability to know how much they make before they make it. By knowing how much you will make before you get paid will enable the people to also not spend more than they have, because they will now exactly how much will be in their account at any given time.

**Use-Case Diagram**



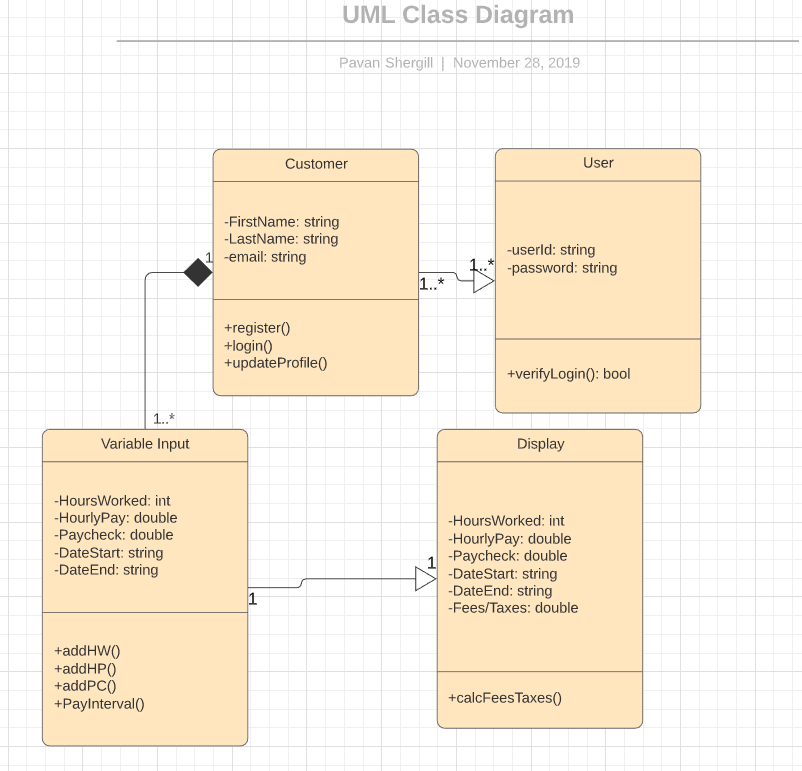
Use-Case Description

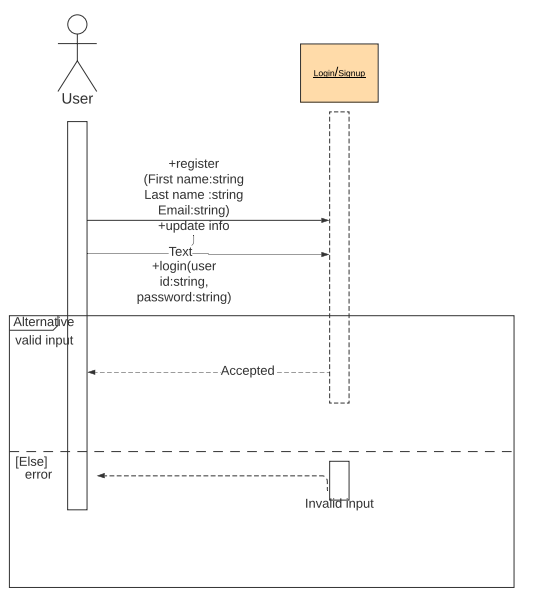
|  |  |
| --- | --- |
| Use Case Title: Login/ Signup | Use Case Title: Input variable |
| Primary Actor: Customer | Primary Actor: Customer |
| Level: kite | Level: kite |
| Stakeholders: Customer, Company Owner | Stakeholders: Customer, Company Owner |
| Precondition: None | Precondition: Login/ Signup |
| Minimal Guarantee: Error on signup customer supports contacted | Minimal Guarantee: Invalid data entered error shown |
| Success Guarantee: Signup and login successful | Success Guarantee: Valid data detected and processes |
| Trigger: Customer accesses app | Trigger: Customer logins and enters data |
| Main Success Scenario:   1. App loaded successfully 2. Login/ signup page pops up 3. User chooses to signup or login 4. User enters app | Main Success Scenario:   1. User logs in 2. User input data 3. Valid data recognized then added to structure |
| Extensions:  1.Error on signup/ login  - Message created to show that data already exist when signing up. Customer must click sign in instead  - Message shown that data is incorrect when logging in. Password or email incorrect fix mistake or contact customer support | Extensions  1.Error with variables inputted  -Message shown for invalid data. User re-enters variable with valid data |

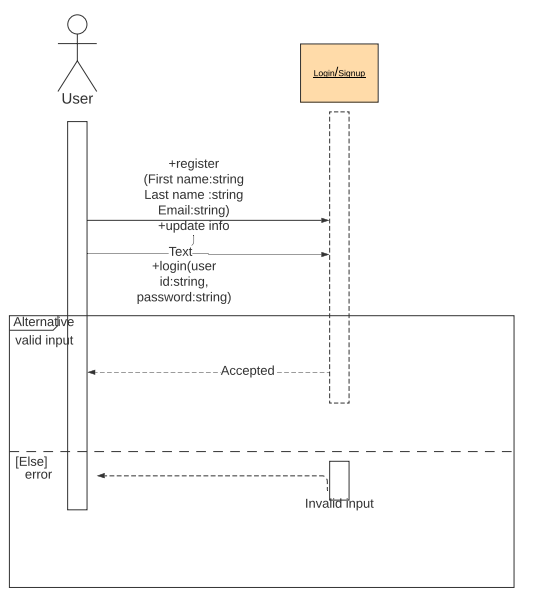
|  |  |
| --- | --- |
| Use Case Title: Calculate | Use Case Title: Display |
| Primary Actor: Customer | Primary Actor: Customer |
| Level: kite | Level: cloud |
| Stakeholders: Customer, Company Owner | Stakeholders: Customer, Company Owner |
| Precondition: Sign in and Valid data inputted | Precondition: Sign in, Valid data inputted, and calculation |
| Minimal Guarantee: error thrown for invalid data | Minimal Guarantee: won’t run if there is error with variable input |
| Success Guarantee: Work pay calculated | Success Guarantee: Will display pay based on hours worked and hourly pay |
| Trigger: User Clicks calculate after input variables | Trigger: Automatically activated when calculate clicked |
| Main Success Scenario:   1. Variable retrieves data from structure 2. Variable added to algorithm 3. Pay calculated then added to structure | Main Success Scenario:   1. Variable data retrieved 2. Work Hours and total pay displayed |
| Extensions:  1.Error thrown  -Customer variable input invalid must re-enter valid variables | Extensions:  1.No data displayed  -Customer must fix previous variables |

|  |  |
| --- | --- |
| Use Case Title: Development and Maintenance | Use Case Title: Troubleshooting |
| Primary Actor: IT Dept | Primary Actor: Customer Service |
| Level: Black | Level: Fish |
| Stakeholders: Customer, Company Owner | Stakeholders: Customer, Company Owner |
| Precondition: Wait for user to complete task | Precondition: Access App |
| Minimal Guarantee: Update issued | Minimal Guarantee: Customer contacts Customer support |
| Success Guarantee: Update issued and installed | Success Guarantee: Problem found and fixed |
| Trigger: IT Dept planning and testing updates then distributing it to users | Trigger: Customer experiences problems then reaches out to company support |
| Main Success Scenario:   1. IT Dept plans and tests update 2. Updated issued 3. Update installed | Main Success Scenario:   1. Customer goes on app clicks troubleshooting 2. Contacts customer service 3. Problem identified and fixed |
| Extensions:  1.App fails to update  -App must be uninstalled then reinstalled | Extensions:  1.Customer Service unavailable or long wait  -reference forum in troubleshooting |

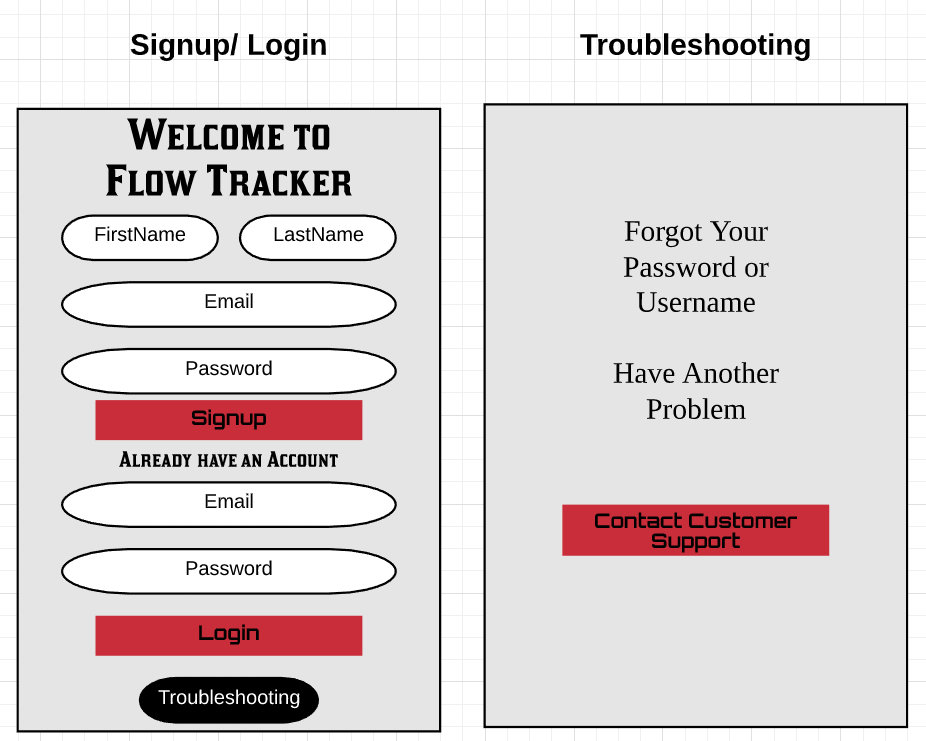
**Class Diagram**

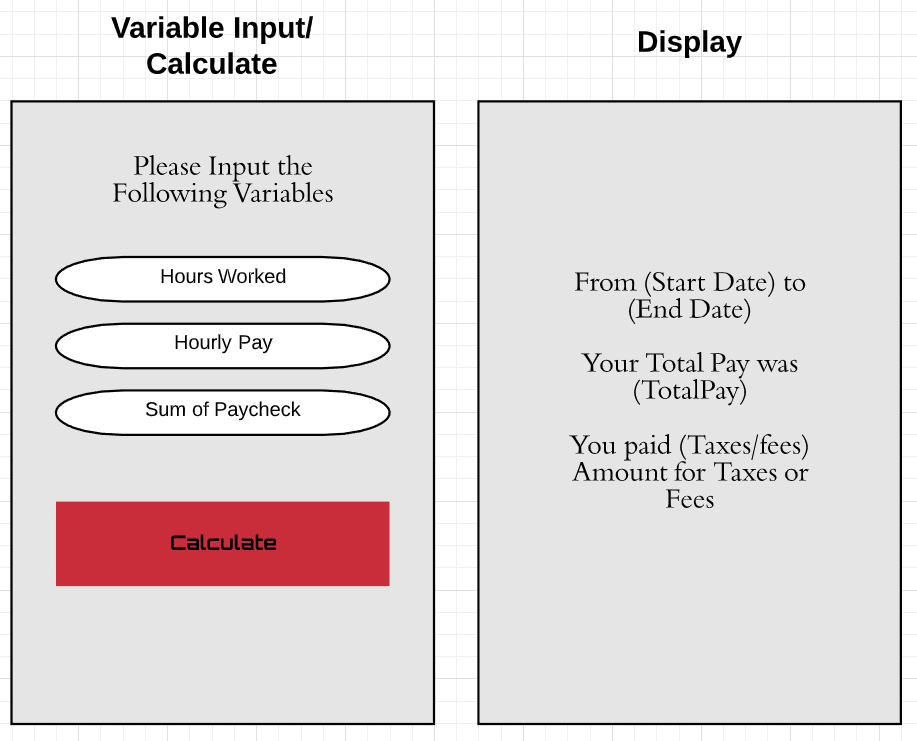






**Interface Prototype**





This project proved to be challenging in some areas for me while not for Pavan, and vice versa. There were a couple of hiccups while we tried to complete the tasks given to us Pavan had a better time making his git-hub repository commit. While mine got jumbled a few times forcing me to restart. While I had a better time understanding how to draw up the use case diagram. Mostly since the project idea that we decided to use was mine. Which made visualizing and physically representing the processes in the use case diagram easier for me. The team work between Pavan and myself was awesome. We met up outside of class time and got what needed to be done finished. While also having a good time doing. We would talk about it in our other classes to make sure we were on the same page. Overall this project was a straight forward one because what we didn’t understand individually, we were able to figure out as a team.**Experience**

**Conclusion**

The ease of access of Flow tracker being on mobile gives the power to ensure the proper amount of pay is received by the user. The easy to understand break down of said amount will allow for peace of mind and financial security for the user. Flow tracker requires very simply tracks the user’s hours worked and can be set up to coincide with your scheduled paycheques. This application has the potential to revolutionize the confidence people will have in the amount they are paid and how much they can spend.

Works Sited

<http://www.ehow.com/how_16566_write-executive-summary.html>

8th Modern Systems Analysis and Design- Joseph S. Valacich & Joey F. George