Criterion E - Evaluation

Success criteria for the application:

1. User inputs income once, states whether is monthly or weekly and the program accounts for that amount without the user having to input it again.
2. Income is added to the amount total at the right date
3. User can input several monthly bills such as rent and netflix once the program accounts for that amount every month without the user having to input it again.
4. Billed amount is deducted on the correct dates
5. User is able to cancel/delete the monthly bills if needed
6. User is able to input one time bills such as movie tickets or dinner.
7. User can ask whether they can afford a certain item
8. Application considered total amount, any bills due in the near future and answers the user
9. Application will let the user know if they are spending too much and are close to spending over the budget for the week
10. The user can set an amount they want to save rendering that money untouchable so the money is saved.
11. The GUI will make the application easy and pleasant to use

Evaluation of the product

My final application met the clients expectations and satisfied all of the success criteria. When the client inputted the information for he spending is he had to also mention what kind of spending it was and this allowed the application to distinguish between weekly and monthly bills. I used the timer API to schedule The program to re-run every day disallowed incomes to be added without the client having to input it again. This fulfilled my first success criteria. I use the date API in order to check whether the current day of the month Matched the day of the month of the date that was inputted by the client. This allowed for income to be added to the amount total on the right date hence fulfilling my second success criteria. Since the information was stored in a database table and then accessed through an arraylist for manipulation it allowed for several monthly bills to be inputted in the program. As mentioned above I also used the timer API which allowed for the bills to be distracted without the client and putting it again. This fulfills my third success criteria. Like mentioned before the use of the data API allowed for bills to be deducted on the right day. Since the information was stored in database table I was able to allow the client to cancel and delete monthly bills through the use of a SQL delete statement That's fulfilling My 5th Success criteria. I created a separate database table for one-time bills and the calculations were only implemented want Huntsville filling the 6th criteria. The application decided whether the client could afford a certain item by accessing the total amount and then deducting the potential cost from it and then checking to see whether it was still within budget . The clients wanted the budget to be $1,000. Hence if the total is ever less than $1,000 a pop-up window is displayed letting the client know that they are now below budget Hence this fulfilled my 7th, 8th and 9th success criteria. The client is able to input an amount they want to save and the money is then deducted from the total and stored separately this fulfills my tenth success criteria. The layout of the buttons and the structure of the application allows for easy use and accessibility Therefore fulfilling my 11th and last success criteria.

Recommendations for the future development of the product

The user mentioned a few recommendations and improvements that I can implement in the future. One suggestion was to turn the desktop application into a web application or a phone application. This would be convenient for the user as the user always carries his phone and would be able to access his finances at all times. Another suggestion was to program the application to generate an email or text message if the user was below budget. Currently the user only knows if he’s below budget while using the application hence having the application alert him through email would be helpful. The last suggestion was to program the application to generate a monthly summary so that the user can have an overview of his expenses every month.