Criterion A: Planning

The Scenario

My brother Daniel has a daily allowance of $15, which is $450 or $465 per month, depending on the month. However, he is expected to spend only $10 per day. The remaining $5 per day (or $150 per month) are in case of emergencies. However, Daniel often spends nearly all his allowance every month, and cannot recall in detail how he spent the money. My brother has looked for budgeting apps to track his spending, but mentioned they are needlessly complicated, request too much additional information, and don't have the option to show how much of your daily budget you have remaining to spend per day.

The Proposed Solution

I proposed to develop an easy-to-use app that will only require his budget, as information, and allow him to track his expenditures. I suggested the app could show how much more money he can spend during each day, month, or year, to stay within his budget of $10 per day. Daniel loved the idea, and so we started to discuss the details and functionalities of the app[1].

Daniel emphasized this app should be on his iPhone, so he can conveniently track his spending throughout the day. He also said the app should let him track the amount, date & time, and spending category of every expenditure, and the input process should be quick. Furthermore, Daniel requested that the app show analytics of whether he's been spending over or under the budget recently, his daily spending on the most spent category this month, his total daily

[1] See Appendix A for transcript of discussion.

spending over all time, and his monthly and yearly spending on each category, so that he can get an idea of his spending overall.

Rationale for Proposed Product

Daniel has an iPhone, therefore Swift and the XCode IDE were chosen because of the numerous advantages they provide for developing iOS applications being native tools to iOS development created by Apple.

Primarily, Swift works directly with Apple's iOS UI building framework: UIKit, which allows me to programmatically create beautiful UI elements for iOS applications. These can be created in the form of classes and re-used to save time and prevent unnecessary code. Furthermore, Swift is compatible with Core Data, a native database framework that allows data to be permanently stored on a device. Since this app will have to fetch spending data every time it is opened and will be regularly refreshed when the user logs new expenditures, it is important to have a fast database framework, which is why Core Data was chosen. Additionally, Swift is an OOP language, and since the application is going to have different views, Swift's ability to make different classes responsible for different areas of this large app is essential to maintain readable and easy to follow code.

XCode was chosen because is very well integrated with both UIKit and Core Data and provides additional tools such as 3D representations of view hierarchies, automatically managing Core Data subclasses, and built-in support for Git version control, which will make debugging and development faster.

Success Criteria of the Project

1. The user can add an expenditure quickly after opening the app, and the user can edit their daily budget.
2. The user can log every expenditure's: Amount, Date & Time, and Category. With at least 8 different spending categories to choose from.
3. The user can see and delete from the list of all his expenditures sorted chronologically and with details about: Amount, Date & Time, Category, for each.
4. The user can see how much money they have remaining to spend within their budget for the following time-ranges: Day, Month, Year.
5. The user can see whether his spending was over or under the budget for the last 10 days.
6. The user can see his daily spending on the most spent category this month and can see his total daily spending over all time.
7. The user can see how much money was spent in different categories in the current month, and in the current year.

Word Count: 500