CP367 Assignment 6/7 Project Plan

# Members

Amardeep Sarang – ***Features 1 and 3***

Shawn Davis – ***Features 2 and 4***

Kreishanth Raveindiraseelan – ***Feature 5 and Project Plan***

# Implementation

The implementation of the features was divided based on what the feature actually implemented. For example, features 1 and 3 are features that implement some sort of constraint on the intersection of holidays and the birthday of the user. Therefore, implementing these two features together seemed reasonable and efficient in the long run. The same principle was applied to features 2 and 4. Features 2 and 4 both affect the general functionality of the program itself, such as implementing command line features and implementing a loop. Therefore, these general functionalities were grouped together. The final feature, feature 5, seemed to be in its area of expertise of its own, therefore it was implemented on its own.

# Order

The order we have chosen to implement the features are features 2 and 4 first, then 1 and 3, then finally feature 5. The reason for this order is based on a bottom-up concept. We start off by implementing core functionality of the overall program by implementing features 2 and 4. These two features will implement general core functions such as offset and the ability to check multiple birthdates. After implement the core functionality, we move on to the addition of additional functionality. This consists of features 1 and 3. Both features implement a check constraint on holidays intersecting with birthdates. Both features are implemented together because they build upon each other. The final feature added is simply for extra utility. Feature 5 indicates the user age if the current date is the birthdate of the user. The reason this is implemented last is because it does not affect the flow of the entire program, but simply adds extra utility on top of the main functionality of the program. The benefits of this order are that we can divide our workflow based on similar functionality and not have to worry about one person implementing completely different features, but more so implement features that are in similar domains of functionality. This allows for the features to be implemented efficiently, and we would not have to worry about one feature depending on the implementation of the other because similar features are being developed together by the same person.