**Red Raider Planner - User Document**

2/14/2020

Cale Enright

Thomas Urech

Omer Qureshi

Brianna Rodriguez

Stephen Parternoster

**Goal of the Project:**

The Red Raider Planner Project we are currently undertaking is a flexible, scalable project with a few certain end goals in mind. In this first iteration of the project we have implemented the basic use cases of a planner/scheduler. Our vision of the project is to have a calendar application with a GUI and all the abilities that one would expect from a planner. So we want the user to be able to add, remove, edit events as well as be able to download and upload calendar templates such that certain things are put on the calendar by default. The default template upon running the application will be that for a regular TTU student, that is, with all the TTU home games and school holidays listed. We also want the user to be able to print out their schedule as a word document or a text file that can be printed in a visually pleasing format. We also want to add a way to add recurring events, ie if you want to add something that wouldn’t change week to week (like a class schedule), you be able to add that event for as long as you wanted. Those are our goals for at least the next release of our project, if we find that it is too ambitious or doesn’t provide enough work for all of our project members, we will scale our goals accordingly.

**Potential Users:**

The potential users of our project are quite general, although we have a specific set of users in mind. Truthfully everyone can use a planner and calendar, however, we plan on tailoring our specific planner towards Texas Tech Students, hence the name Red Raider Planner. In our future releases, we want a default template to be available that includes the college home games, holidays and other days when school is canceled. Anyone at Texas Tech who would be interested in a free calendar would have access to it, they would simply need to download the files and use whatever template they wanted.

**Use Cases:**

addEvent: The basic use case we currently have, when a user wants to add an event to a specific day they choose addEvent, which goes through a series prompts to the user asking the user for information about the event they wish to add. The prompts ask for Date, start and end time, name of the event and description. After the inputs have been received, the event is added to the specific day on the calendar.

removeEvent: This counterpart of addEvent, if a user wishes to remove a specific event, the user selects removeEvent, and the user is then prompted for a date. Upon receiving a date a list of events on that day is listed, and the user then chooses an event to delete.

editEvent: This function is called if a user wants to change an event without deleting it completely and just adding a second event. The user chooses editEvent, and is then prompted for a date. After entering a date the events on that day are displayed. From there a user chooses which event to edit and then is prompted to select which element of the event they want to edit. They then enter the updated version of the event they wish to change, and the event is updated.

viewCalendar: This function is called if the user wants to see all events the user has currently entered. Once called every single day in the calendar is incremented through and if there are events located on this day they are outputted. Days without any events are not outputted. Essentially this function outputs the list of all events.

viewWeek: This function is called if the user wants a view of events in the next seven days. The user must call the function with a date, and then the program finds the next seven days and displays them. If there are events in those days they are listed along with the date they are attached too, but even if the days hold no events they are still listed with a notice that there are no events on that date.