**Group 1 SDS**

**1/18/2019**

**Tyler Green, Tyler Milan, Bryce di Geronimo, Michael Zhang, Jarvis Dong**

**Table of Contents**

1. **Product Description**
2. **Design Description**
   1. **System Parts**
      1. **Database**
      2. **Control Module**
      3. **User Interface**
      4. **Static Model**
      5. **Dynamic Model**
3. **Design Rationale**

**1. Product Description:** This product is a simple calendar application which will look like a standard monthly calendar where the month and year is depicted at the top with the numbers in each box corresponding to the days. The boxes will all be blank until the user clicks on a box and adds an event that will contain the time and event description through a pop up box. If the event is added, a marker will appear on the calendar indicating that an event is going to take place at that day. To see an event or all events for a given day, the user will hover over the marker which will show the time and description for all events that day. If the user clicks the event, they will be given the option through the pop up box to edit or delete the existing event.

**2. Design Description:**

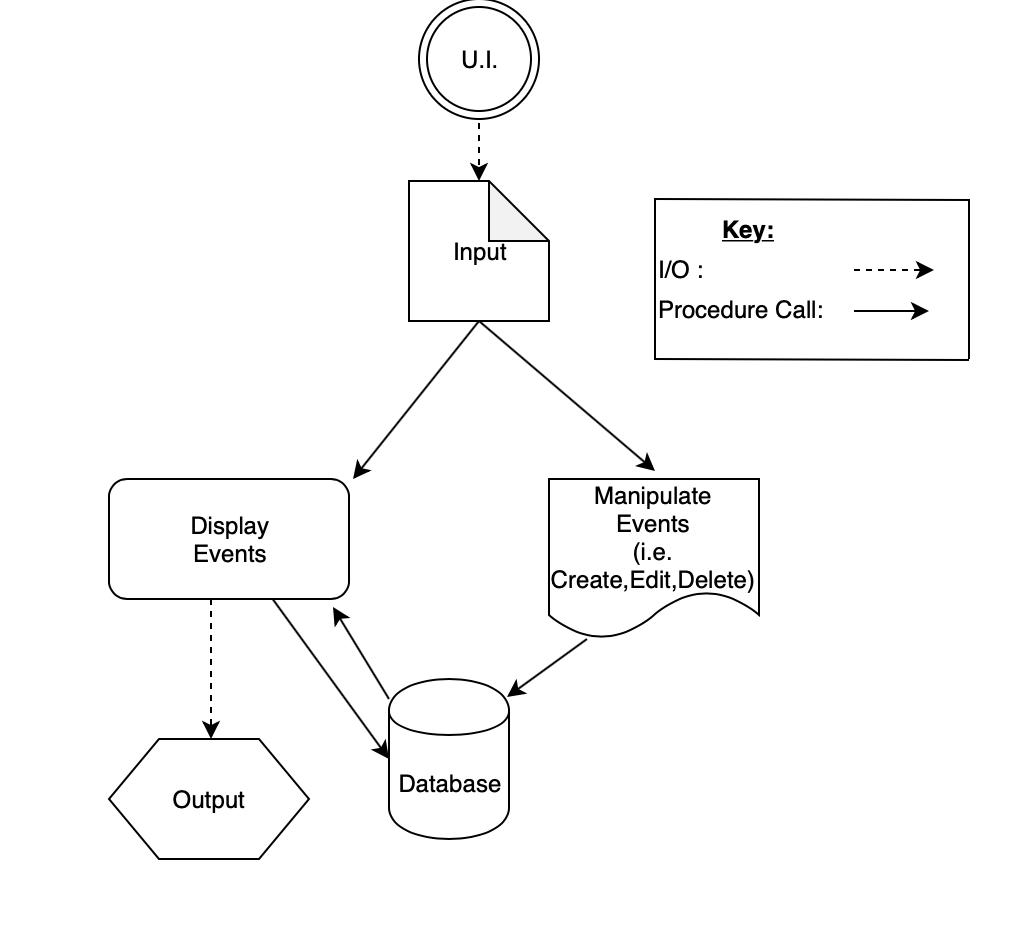
**2.1 System Parts:**

**2.1.1 Database:** A SQL database that will contain User Event Data which includes Date stored in DD/MM/YYYY format, time which will be stored in HH:MM format, priority of events stored as “high”, “medium” , or “low”, and categories of events which will be stored as “work”, “home”, or “school”.

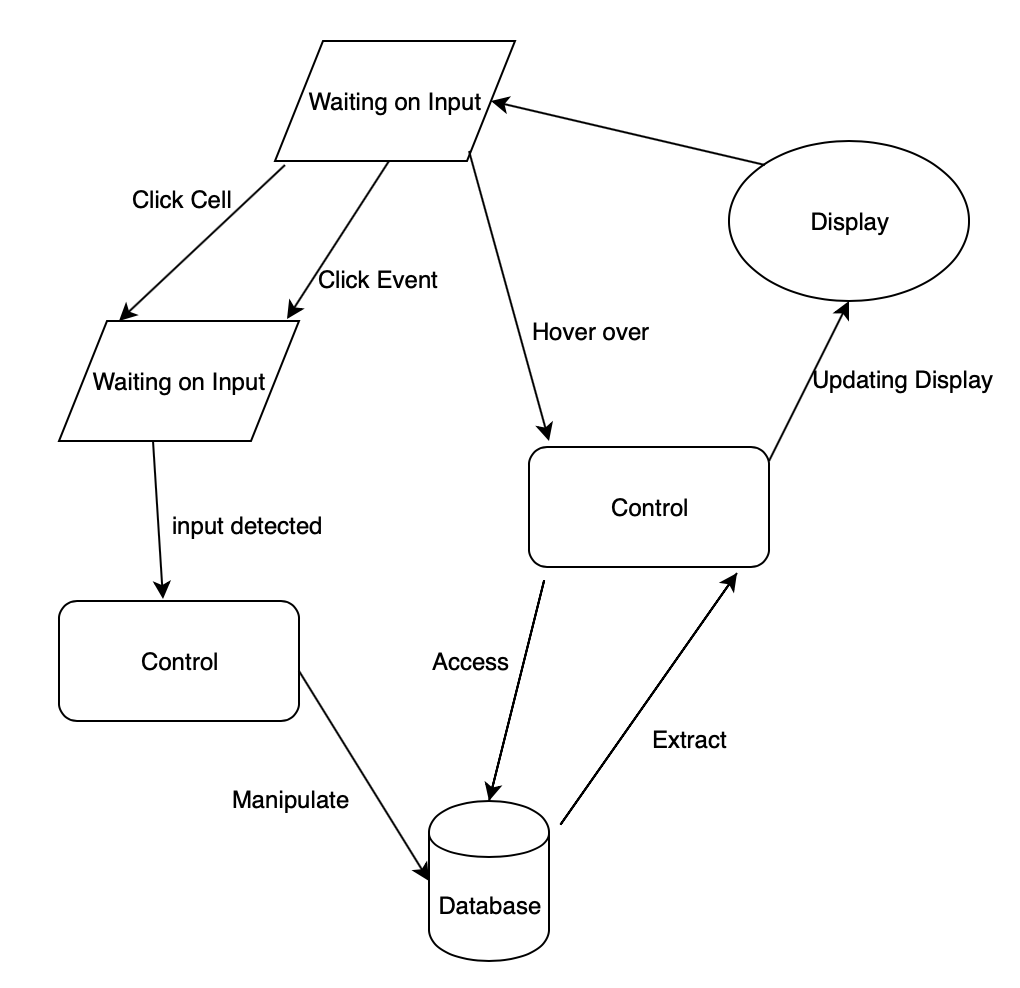
**2.1.2. Control Module:** The control module contains functions that interact with our database described above. These functions include a add function which will execute when the user wants to add an event to the calendar, an edit function which changes a pre existing event information, and a delete function which deletes an event. These can all be invoked through the user interface. All these functions are written using PHP which allows us to implement the various SQL commands required to perform these functions.

**2.1.3. User Interface:** Initial Calendar view uses HTML, CSS, and Javascript for functionality. Users are able to scroll using a left and right arrow to find desired month which ranges from January 2018 to December 2020. When the desired month is found, users are able to see all days of that month and any days that have already had an event added to them which will be represented by a marker. When a user clicks on a date and an event has already been created, information regarding this event will be shown. User will also be able to edit and delete this pre-existing event a pop up box, which in part triggers the respective control module function. If a user clicks on a day that has no pre-existing events, the popup will appear blank.

**2.1.4 Static model**

****

**2.1.5 Dynamic Model**

****

**3. Design Rationale:** Our design model that is explained in section 1 and 2 of this document was chosen as it provides an intuitive full month calendar design that is easy for the user to interact with. We have also eliminated the need to scroll by implementing a hover over function to view events on specific dates.

The main organizing principle that was implemented in our project design was the idea of modular programming. Modular programming is a software design technique that emphasizes separating the functional aspects of our program into individual modules. This allows more flexibility in dividing work amongst group members. It also allows independent modules that allow greater flexibility in division of work and easier debugging of our program as each function is contained to a specific module.