Wait-Less Product Scenario 2

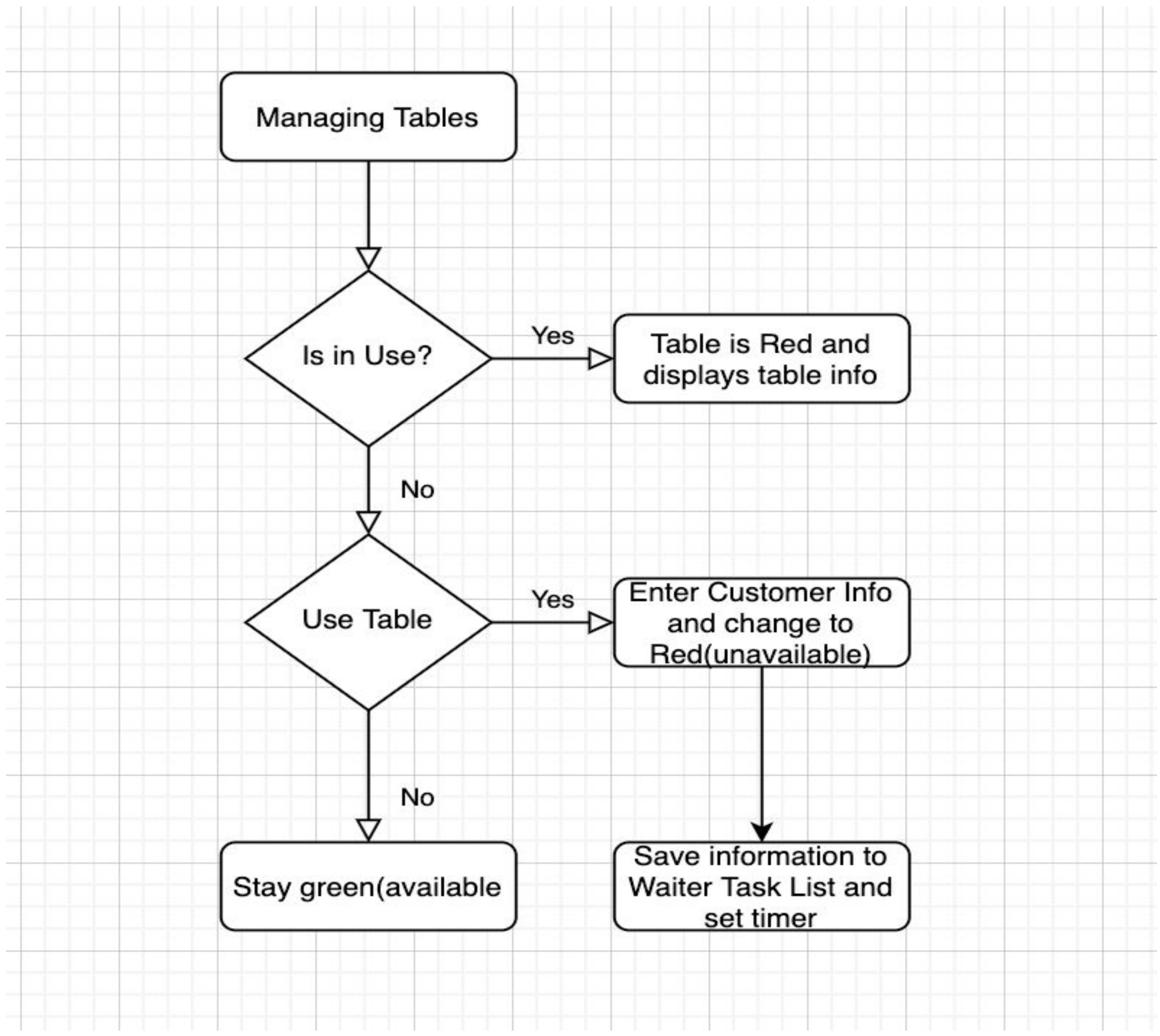
Maaz Ahmed, Ammar Idrees, Zaid Patel, Bryan Gutierrez

For our second release, we plan on adding a lot more functionality to the product, while also improving the overall UI of the product to appeal to customers a bit more. Regarding some functionality, we plan on also adding a menu to further improve performance within the app. The menu will feature different items within the restaurant that could be ordered by the customer. The menu along with the order the customers have arrived would determine the overall algorithm behind which customers should be served first. To determine the weights of how important the customers are, we are going to implement Dijkstra's algorithm to get the most optimal and shortest path of tables. Furthermore, we want to somewhat replicate actual restaurant apps and a lot of features that they provide. Some other features we had in mind to make the app even more convenient would be having a feature to order beforehand, bumping the status of that customer. For the actual ordering and menu portion of the application, we plan on implementing the builder design pattern. It will basically add food items as an object, to create meals for the customers through other objects. Furthermore, it will help change how the objects are represented within this feature if needed. Alternatively, in regards to in-app functionality, we will also have to implement some multi-threading to allow concurrent execution for the tables. This will give us the ability to update the tables all at once, in correlation to upcoming customers and those who have finished. To enable this, we would be extending from the thread class, while utilizing inbuilt methods.

One aspect of the project we aim to improve from the first release is the Graphical User Interface. As it stands currently, the waitless product uses a very basic interface utilizing JavaFX. When logging in, the app transitions to a screen that shows a small window containing tables as buttons and text boxes. We are going to research different applications that do exist to better understand and implement an effective design. Currently, an idea that is doable for JavaFX is to first implement a full screen, that way when the app is run it is easy to view. Furthermore, we can implement images for the buttons and tables. As a result, this would create a visually appealing display. Looking at some waiting apps on google images shows a lot of colorful displays, which could be something we can implement. In addition to the tables screen, the task list should be more complex in its functionality. As it stands, it simply is a list of tasks and a text box at the bottom to add more tasks. The first thing we would want to do is to give it some color to make it visually appealing. Beyond that, we would want to start labeling the tasks with employees that need to complete them. Moreover, we would want to add a feature that would allow a manager to assign a task to an employee when adding tasks to the list to improve functionality. Lastly, we would want to improve the stats display. As it stands currently, it's just a simple table with no significance. We would improve that for the second release by displaying calculated stats, and try to figure out what additional stats we could add that would be useful.

Another aspect of the project we aim to improve from the first release is to increase the number of features for the waiter. We want to improve their ease of access. The first feature we want to implement is a timer that sends push notifications on the application. The purpose of this is to give the waiters a reminder of periodically checking up on the tables. An added bonus to this is to make the tables blink on the UI as well. Furthermore, we want to have that once a table is set up and you enter the party name and number it becomes permanent once you click the table button. This means that the party name cannot be changed until the table is open again. Lastly, we want to be able to have a user create an account and also differentiate between manager vs waiter.

Table Management Diagram


Managing Tables
Is in Use?
Yes
Table is Red and displays table info
No
Use Table
Yes
Enter Customer Info and change to Red(unavailable)
No
Stay green(available
Save information to Waiter Task List and set timer