**Software development requirement & planning document for Wedding APP**

**statement of purpose**

**1.What the application will do**

It is a wedding planner command-line application, which will help the user to choose the best option for their wedding ceremony.

**2.What business problem it addresses**

Traditional wedding planning is manual. It requires the user to actually go or call the venues to get information. This app will help couples to plan their special night in a more integrated way and it will reduce their amount of pressure. They can also estimate the price they are going to expect and play around with the packages to find their best option based on their budget.

**3.The target audience**

As the app name suggests, this application will target any couple who wants to plan their wedding. Regardless of their age, and sex.

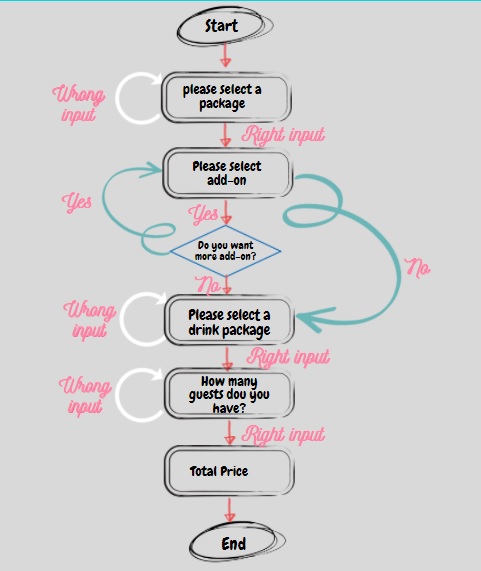
**list of features that will be included in the App**

The app's start point is by greeting the user and show them the list of available wedding packages. It will ask the user to choose their desirable wedding package, they can decide between different wedding lengths, styles, and prices. Users can pick the package by entering the package number.

After the package being chosen, then the application will display a list of additional items which can be added to the wedding package. The users can add zero or many additional to their package. They can move forward to the next step by entering no.

Next, the App will present the list of available drink packages with their details and price. the user is required to select one of the drink packages to continue with the application.

Finally, the application will ask the user to enter the number of guests they are planning to host, to calculate the total price. After the estimated total price being presented, the app will ask the users if they wish to provide their phone number in order to be contacted for further discussion. If they do not wish to be contacted, they will simply enter no and the application will reach its end point. You can observe the app flow chart provided in the folder.



**Implementation**

The first step to Implement the wedding application was brainstorming. Deciding the necessary classes and models and how to map them together. The Wedding App has several Models and classes which are as follows:

**1.Class Main**

This is the main entry point of the project, which will run the methods required for the app to do its job.

**2.Class Play**

This class contains all the methods and their actual code. The methods in the class main are static and can be accessed directly from the class play.

**3.Class Wedding Packages**

This class is a model class for wedding packages. Which stores the package id, hour (the length of the function), List of the Add-ons, Drink package, Style of the wedding, and the price of each package. Add-on and DrinkPackages are a class themselves that are included in the Wedding package.

**4.Class Drink Packages**

Drink packages class is another model class to store each drink package; this class includes the drink package's id, price, name, and detail. Users can only choose one drink package.

**5.Class Add On**

The class add-on also is a model class, which is used to store the add-on id, name, price, and detail. Each user can add one or a few add-ons to their package.

**6.Total Price Calculation**

Finally based on the user selection of add-ons, drink packages, wedding packages, and the number of guests they plan to have; the app will calculate the total estimated price.

After implementing the classes, I started to work on the methods required to get the desired outcome. I added the first method(choosePackages()) which uses the getWeddingPackages() to choose a wedding package first. Afterward, the rest of the methods were implemented to get the correct output. The caculateTotalPrice() method will eventually calculate the final price based on the data retrieved from the other methods. At the end, I called all the methods in the main method in order to run the application.

**Error Handling**

I conducted adequate error handling operations throughout the implementation part to make sure the user enters the correct input. Otherwise, the application will ask the user to enter the input again.

**Testing the application**

I made sure that the application runs smoothly and correctly by testing it manually and also by operating JUnit testing.