**Seating Arranger (SA) High Level Design**

**Overview**

Seating arranger (SA) feature goal is to create **suitable** seating arrangements for weddings. SA will generate an accurate as possible first seating arrangement and then will let the user modify the arrangement if needed.

**SA goal**

1. Create the most efficient and suitable seating arrangement as possible.
   1. Guests will be happy with there sit
   2. Guests will be happy with the company around them
   3. Fill each table as much as possible
   4. Decrease number of tables as much as possible.
   5. Save valuable time for the event organizer.

**SA Preconditions**

1. Guest list
   1. Full name
   2. Groom/bride side
   3. Group -each guest needs to be a member in a group

Friend from work, close friend, fathers work, etc. (a non group guest will be marked as other and will be seated accordingly).

* 1. Number of people.
  2. A guest list template will be included as part of the feature.

1. Event list
   1. List of tables:

|  |  |
| --- | --- |
| Table capacity | Number of tables |

**SA Flow**

SA flow can be divided into four main steps:

1. Waiting for user input
2. Generate data structures based on input
3. Generate a seating arrangement and present it to the user.
4. Wait for user additions, deduction or modifications.

**Data Model**

In order to achieve its goals SA will use some classes, the classes that are responsible for the main flow are presented in the diagram below.

**Class/Component Diagram**

* guest: This class is used to represent a guest.
* Guest list: This class works with guest to create a guest list for a specific guest group. Meaning for each guest group a guest list instance will be created.
* Table: this class is used to represent a table. This is a generic class that will have different inheritors based on the number of seats.
* Event Hall: this class will include data structures that contains the classes above and will be the class that the seating arrangement will be preformed on.