Stories and tasks

# Card 1 (1.5 days)

**Story:** As a dietician I want to create recipe.

**Back of the card:** A recipe would contain name, list of ingredients and list of steps. This could be represented by a class object in Java. We will probably have multiple recipes in the system so we need some type of list or pool to contain all of them. A recipe should also have an ID.

**Tasks:**

1. Create a class for a recipe that contains a name, list of ingredients and a list of steps.
2. Create some type of method for displaying a recipe. This could probably be done by overriding the toString method of the recipe object.
3. Create some class (preferably named *RecipePool*) that can hold all recipes.
4. Add some method for fetching a Recipe from the RecipePool
5. Add some 15 or so mockup recipes to the RecipePool that we can use for testing.
   1. Find 15 real recipes and add to this RecipePool
   2. Select dietician as user and select view all recipes to show list of all recipe
   3. Select a recipe to show details of the recipe selected.

# Card 2 (0.5 day)

**Story:** As a user I want to be able to generate a food schedule for each week since I don´t have time to plan everything myself.

**Back of the card:** We could pick 7 random recipes from the RecipePool. These should be stored in its own class(presumably named *RecipeWeek*).

**Tasks:**

1. Create a prototype on how to select 7 random recipes from the RecipePool.
2. Create a class named RecipeWeek that contains a recipe object for each day of the week.
3. Create a class to hold the map of week number and RecipeWeek so that user can select a week and see the recipes for the particular week

# Card 3 (1 day)

**Story**: As a user I don't want to eat the same dish every day, since this is not healthy.

**Back of the card:** We need to add some type of algorithm such that a recipe does not show up twice in a row(in two following weeks). This could probably be done through some type of cool-down period.

**Tasks:**

1. Add an attribute to the recipe class that holds the number of times the recipe is selected
2. Increase the counter created in above step each time the recipe is selected to be added in the week
3. Choose recipe with the least count and repeat step 2&3 for all seven days
4. For the cood-down period of 2 weeks, add an attribute lastAddedInWeek which contains the week number in which the recipe was last added
5. While selecting the recipe, check for the lastAddedInWeek attribute, it should not have value equals to current week number, previous week number and next week number.
6. For example if the current week is 32 then lastAddedInWeek should not contain values 31,32,33.

# Card 4 (1 day)

**Story:** As a user I want to view a recipe so I can prepare a meal.

**Back of the card:** Show options to 1. View weeks 2. View current week 3. View today’s recipe and 4. Add week. 5. View recipe pool 6. Go to main menu

**Tasks:**

1. While selecting the option 1, show list of weeks created by the user
2. Create a utility to provide the current week number and fetch RecipeWeek for the current week number
3. Use the created utility to get current week number and create a utility to get the week day to fetch the current day recipe from the fetched RecipeWeek
4. When user selects option 4 to add new week then ask for the input for week number, check if the week already exist,
   1. If already exists then show alert message
   2. If not already exists then create the requested week with recipes received from random recipe selector prototype generated in the story number 2.

# Card 5 (0.5 day)

**Story:** Users can view the recipe details while dietitians can edit recipes also.

**Back of the card:** when showing recipe details to dietitian, show an option to edit the recipe

**Tasks:**

1. Add edit options
   * + - 1. Edit name
         2. Add ingredient
         3. Add step
         4. Delete ingredient
         5. Delete step

(Please note that we can have option of adding the ingredients and steps at a particular index but keeping it out of current project scope)

1. Reuse the methods for adding ingredients and steps from the create recipe class
2. Ask for save recipe or edit more after each edit.

# Card 6 (0.5 day)

**Story:** continuation of tasks

**Back of the card:** each menu should have option to go back to previous menu or main menu or likewise

**Tasks:**

1. In the main menu add option to exit with a goodbye message
2. In dietitian menu and user menu add option to go to main menu
3. In other menus, add options to go to respective back menus or probable other options that user can opt so that user is not stuck anywhere.