

# **DATA MODEL**

CSCI375

**Team Benz**

March 19, 2018

## **Document Overview**

In this document, we begin by providing an update on the current state of the project and noting the changes made to the project. We then move to the data model section of the document, in which we fully describe our product's logical data model with respect to entities, attributes and relationships.

In the data model section we start by giving an overview of the data provided in the model. Follow by a descriptions of all the components to be presented in entity relation diagram. The components mainly cover the entities, attributes and relationships. This will then be followed by the entity relation diagram along with the relational schema. We end the document with preliminary description of how these logical entities, attributes, and relationships will be represented in the real physical model

## **Project Update**

We will be coming up with an updated set of Task Plan on wednesday now that we have a better understanding of all the components and each skills needed to complete them.

# Data Model

For this data model we have four identities or tables. All the entities and attributes are relative to the idea entity. Each such entity or table essentially holds data we need to store and use for our system. The following is an overview of the data included in this data model:

1. Idea ID
  - a. Idea Name
  - b. Idea Description
  - c. Idea password
2. Idea Wishes
  - a. Wish number
  - b. Wish name
  - c. Wish status
3. Idea Steps
  - a. Step number
  - b. Step name
  - c. Step status
4. Idea Note
5. Idea User
  - a. User ID
  - b. User name
  - c. User location
  - d. User email

## Description of the components:

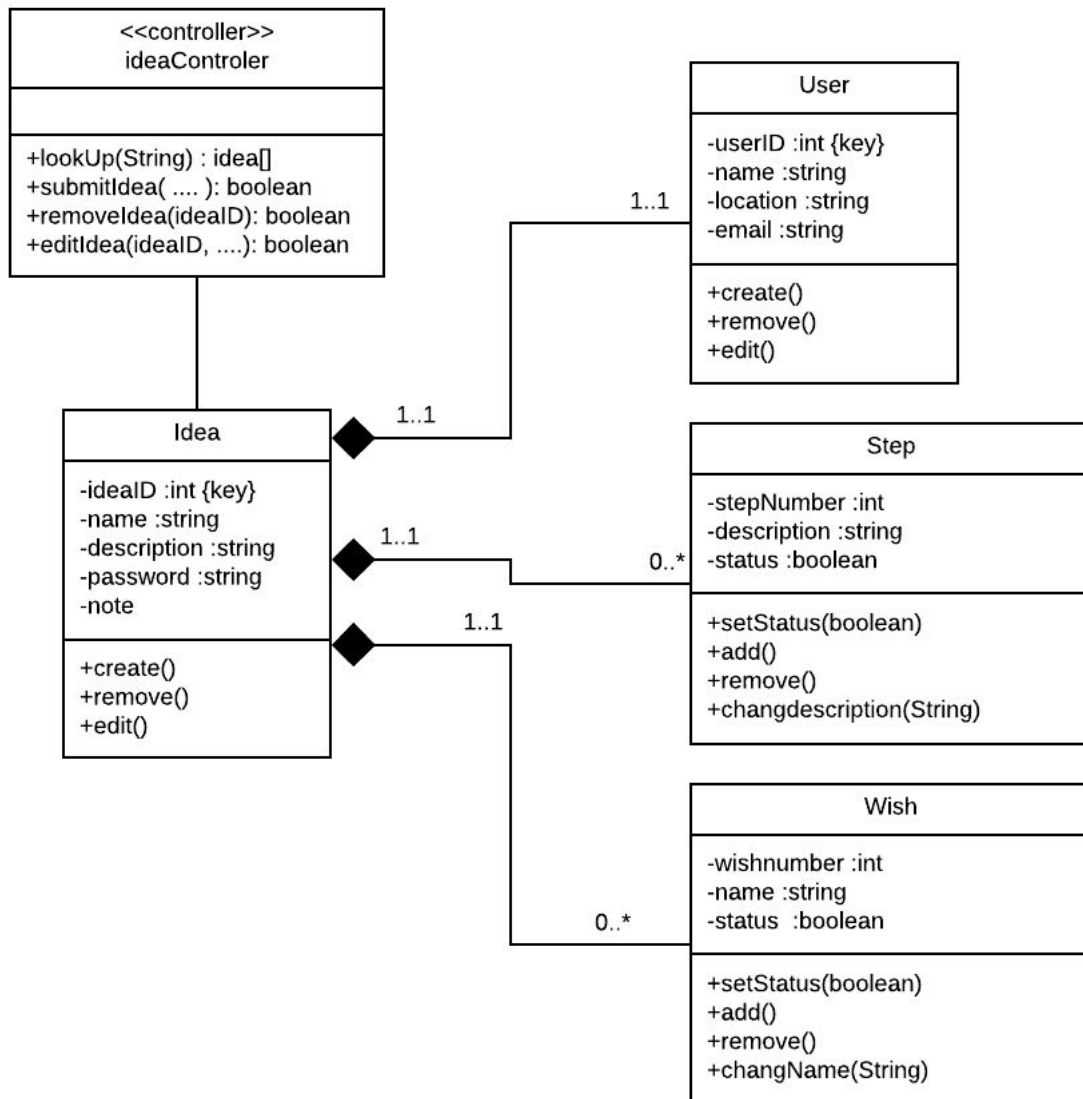
1. Idea - This table holds data associated with an idea and holds the following attributes:
  - a. IdeaID (int): This attribute acts as an identifier for all the different ideas in the table. This should be automatically generated.
  - b. Name (string): This attribute is the name given to the idea.
  - c. Description (string): This attribute is the description associated with a given idea.
  - d. Password (string): This attribute is the password associated with a given idea.

- e. Note (string): This attribute is the note associated with a given idea. This can be null.
- 2. User - This table represents the user and holds the following attributes:
  - a. UserID (int): This attribute acts as an identifier for all the different users in the table. This should be automatically generated.
  - b. Name (string): This attribute is the name associated with a given user.
  - c. Location (string): This attribute is the location associated with a given user.
  - d. Email (string): This attribute is the Email associated with a given user.
- 3. Wish - This table holds all wishes associated with a given idea and has the following attributes:
  - a. Wishnumber (int): This attribute is the number that identifies one wish within a number of wishes associated with one idea.
  - b. Name (string): This attribute is the name associated with a given wish.
  - c. Status (boolean): This attribute gives the status of the wish, either true for complete or false for to be completed.
- 4. Step - This table holds all steps associated with a given idea and has the following attributes:
  - a. stepNumber (int): This attribute is the number that identifies one step from all steps associated with one idea.
  - b. Description(string): This attribute is the description associated with a given step.
  - c. Status (boolean): This attribute gives the status of the step, either true for complete or false for to be completed.

### **Relationships:**

All the entities and attributes are relative to the idea entity. The user entity has a one to one relation with the idea entity. The user is linked to an idea by the user's ID stored within the idea entity. Although with a one to one relationship, it's common practice to just place every in one table. We divided them into two tables in case in the future the system is upgraded to give a user some form of relationship outside one idea. One example of such an upgrade, would for a user to have all the ideas show under one account. Then we would have a many to one relationship and therefore will need two separate tables. As for the wish and step entities they have a many to one relationship with the idea entity and their existence depends on the existence of the idea they relate to. If one idea is removed, the steps and wishes should also be removed.

## Entity relation diagram



## Relational Schema:

Idea (**idealD** , owner, name, description, password, note)

User (**userID**, name, location, email)

Wish(**idealD**, **wishnumber**, name, status)

Step(**idealD**, **stepNumber**, description, status)

**Preliminary description of representing the real physical model based on the logical entities, attributes, and relationships presented:**

In the real physical model we have about six functions that will deal with the the entities, attributes and relationships presented. We will describe each bellow:

1. Creating an Idea: When the user creates an idea they will essentially be creating a row in the idea table that hold all the attributes of the idea. They will input a name, description, password and a note if they have a note or could be left blank (NULL). The idea id will then we automatically generated. When the user inputs their name, location and email, a row will be creating in the user table to hold that info and a user id will be created which is also automatically stored in the idea as the user id. When the user adds a wish and enters the name and status, a row for each such wish will be created in the wish table, and the idea's id will automatically be stored in each wish row along with a step's number. When the user adds a step and enters a description and status for the step, a row will be created, and the idea's id will automatically be included and a step number created.
2. Removing an Idea: When the user remove an idea, all the rows inserted when the idea was initially created will be removed.
3. Editing an idea: When the user edits an idea, they have access to all the ideas attributes except for IDs and numbers. They can change as they wish and the system will make the change in the database.
4. Listing the ideas: When in list mode, the system will select and output all the idea's IDs and names from the idea table.
5. Viewing an idea: If they want to view, the system will join all the table together relative to the ideas id and output the content of the table except for the password.
6. Searching for an idea: When the user does a search, they are essentially searching for the name of the idea, and therefore the system will look for all the idea names that are substings of that pattern and output each idea id and name.