Case Study I
Date: 11/04/2016
Name: Rahulkumar Gaddam
Email ID: rgaddam@syr.edu

Question:

High-level requirements for this application consist of the following:

- The recipient can identify 5-10 submitters in their organization to request feedback to be given about the recipient.
- Feedback entered by the submitter consists of answering a number of questions about the recipient.
- Each organization may create their own questions or choose questions from a question bank pre-populated in the application.
- The submitter has the option for their feedback to be anonymous.
- The date of the feedback submission should be recorded and displayed to the recipient.
- The recipient can view the contents of the feedback – including date, feedback, and submitter (if not anonymous).
- The recipient can view a list of all feedback submitted to them, including average score across all submitters
- The recipient could request the above to happen multiple times (for example: in the beginning of the year and in the middle of the year) and average score need to be shown per time period and not across all answers

Answer:

We can think of two scenarios.

Scenario#1:

When the relationship between recipient and submitter is not Hierarchical. This means employee can be a recipient and submitter both.

Scenario#2:

When there is a Hierarchical relationship between submitter and recipient. This case Recipient cannot be submitter.

Example: Government asks citizens to submit feedback.

Considered Assumptions before building the both models:

1) Submitter and recipient are from same organization.
2) Individual profiles and roles are maintained under different model.
3) System automatically picks up recipients from the database and excludes current user from the list.
4) Ideal environment for the system is assumed.
5) Timestamp is picked up from the current server time and inserted when a user submits a Feedback.
6) If submitted as Anonymous, then the submitterID will be Anonymous(Predefined).
7) Custom Questions inserted in to the form in the UI will create a new entry in QuestionBank.

Scenario#1:
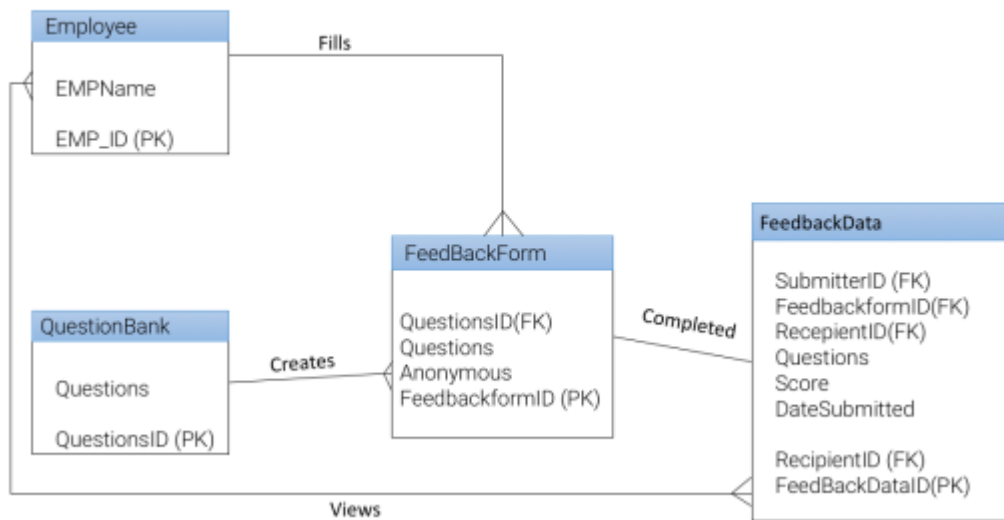
In this Data Model, Organization creates feedback form from a question Bank and submitter fills the feedback for the recipient. After receiving feedback data from the recipient score is calculated. When user decides to be anonymous a default value is given under submitterID.

Entity:
- Employee
- Question Bank
- Feedback Form
- Feedback Data

Primary Keys:
- Employee ID(EMP_ID)
- Question Bank ID: (QuestionsID)
- FeedBack Form ID: (FeedbackformID)
- Feedback Data ID: (FeedbackDataID)

Foreign Keys:
- Recipient ID
- Submitter ID : (derived from Employee ID) for Feedback Data
- QID : (derived from Question Bank ID) for FeedBack Form
- FID : (derived from FeedBack Form ID) for Feedback Data

Relationships:
- Fills
- Creates
- Completed
- Views

Calculating Score:

For instance, if a person receives all positive comments, such as good, excellent etc., then it adds a predefined score. If a person receives a negative comment as bad, poor, etc., then it removes score by predefined score. We have to provide the predefined score for each of the value. So, if a person receives 6 positive comments and 4 negative comment so his score will be 2 out of 10. Then the given data along with individual score and the date on which feedback was recorded will be stored in database taking from server time. Or we can calculate the score at runtime when displaying the responses but then we have to calculate individual scores to find out average score which is explained below.

Calculating Average Score:
After everyone submitted their scores then again we can calculate the average of all scores by taking sum of all individual scores and then dividing by total number of submitters. (Avg= (1+2+3+ …..+n)/no. of counts).

Recipient can see the feedback whichever day he wants by finding the pre calculated individual scores till that day by limiting the date field.
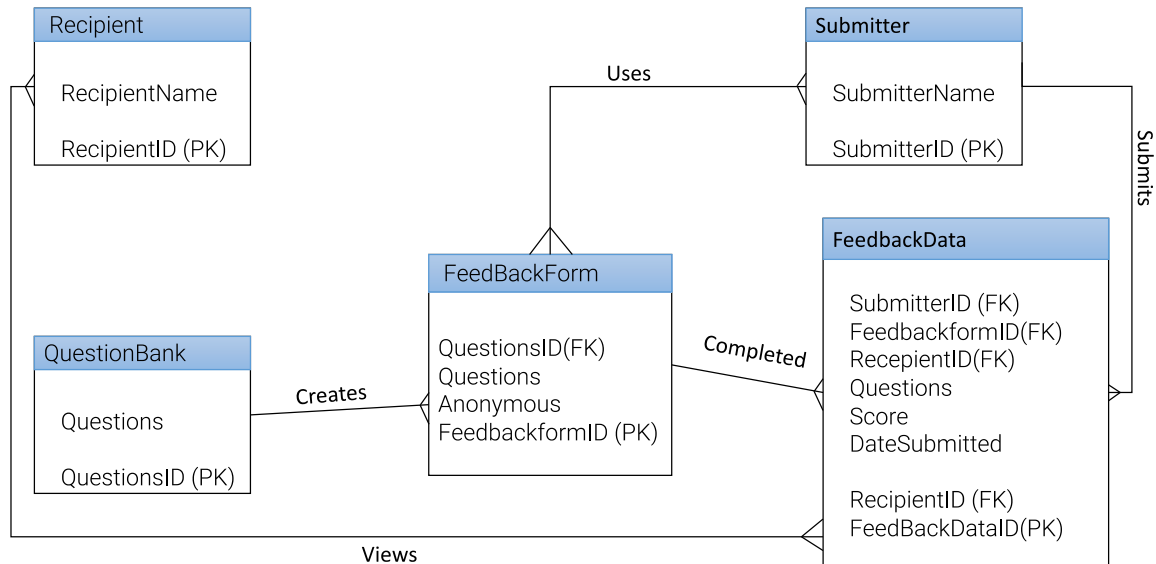
Scenario#2:

Recipient

RecipientName

RecipientID (PK)

Submitter

SubmitterName

SubmitterID (PK)

Uses

Submits

FeedBackForm

QuestionsID(FK)
Questions
Anonymous
FeedbackformID (PK)

FeedbackData

SubmitterID (FK)
FeedbackformID(FK)
RecepientID(FK)
Questions
Score
DateSubmitted

RecipientID (FK)
FeedBackDataID(PK)

QuestionBank

Questions

QuestionsID (PK)

Creates

Completed

Views

In this data model we are assuming that we follow hierarchical model and organization creates feedback form the Question Bank . Even through recipient creates the Question the created Questions will be inserted in to the QuestionBank (Will restrict UI to create a data that is suitable for the Question Bank Database). Submitter fills the feedback data for the recipient. Once the feedback data is filled score is calculated and Anonymity is being stored by utilizing predefined value for anonymity.

Entity:
- Employee(Recipient,Submitter)
- Question Bank
- Feedback Form
- Feedback Data

Primary Keys:
- RecipientId
- SubmitterId
- QuestionId
- FeedBackFormId
- FeedbackDataId

Relationships:
- creates
- completed
- Uses
- Views
- Submits

Algorithm for Calculating Score:
1) Create a training set
   - Positive words are given a score of 1
   - Positive words such as good, awesome, nice etc.
   - Negative words are given a score of -1
   - Negative words such as bad, poor, pathetic etc.
2) Count the number of positive words
3) Count number of negative words
4) Subtract the numbers
   - If you get a number 7 or greater set that as a positive feedback
   - If you get number 5 or 6, then it is a neutral feedback
   - If you get a number less than 5 then it is a negative feedback

Algorithm for Calculating Average Score:
1) Get all the scores for a particular individual across all submitters
   - If user wants across a particular date then retrieve accordingly by using appropriate query
2) Count number of scores and store that in count variable
3) Sum all of the scores
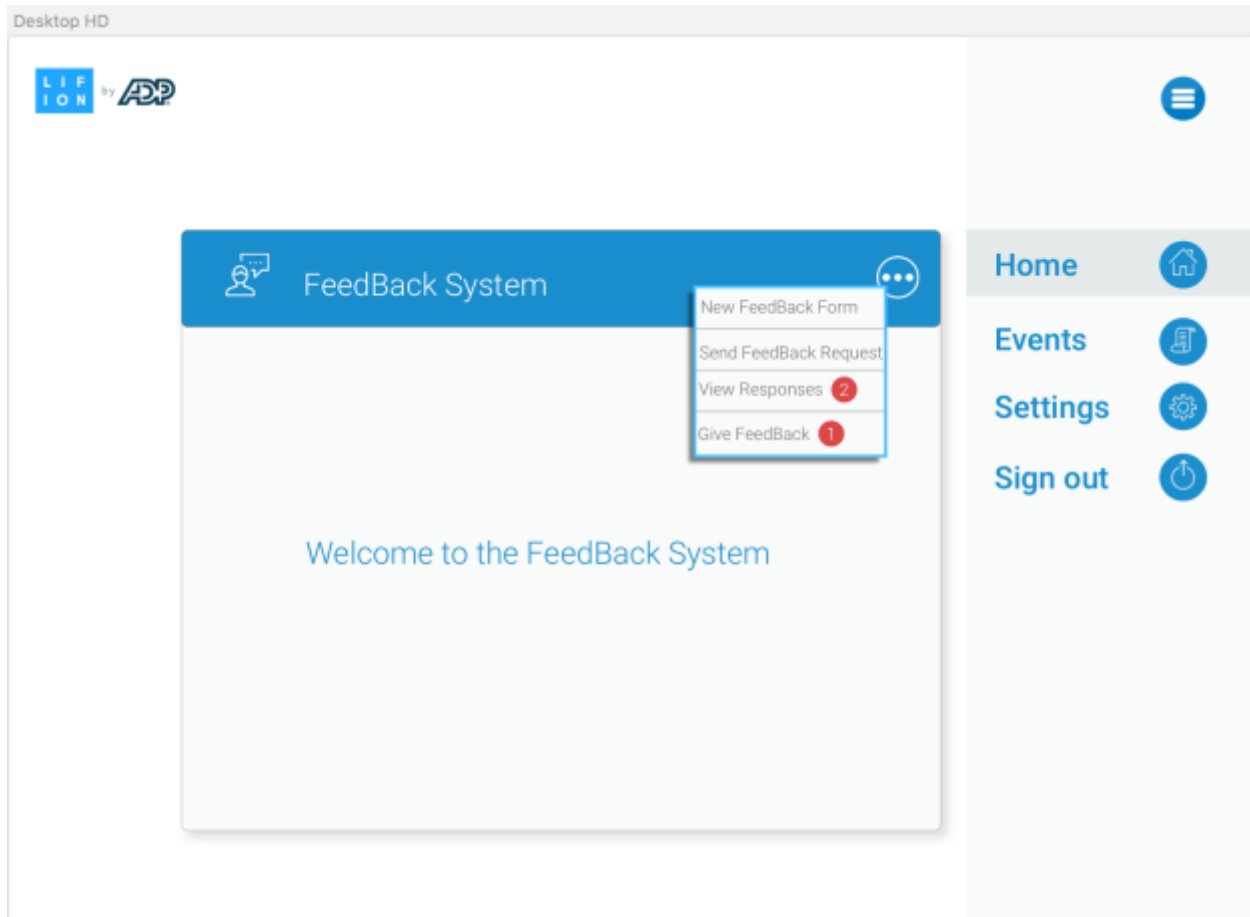4) Divide the sum by count and store as average
5) Thus you get the average score

User Interface for the Feedback System is given below:

Desktop HD

FeedBack System

New FeedBack Form
Send FeedBack Request
View Responses 2
Give FeedBack 1

Welcome to the FeedBack System

Home
Events
Settings
Sign out

Main Screen

Case Study I
Date: 11/04/2016
Name: Rahulkumar Gaddam
Email ID: rgaddam@syr.edu

Desktop HD

FeedBack System - Create   Done

Section 1   Edit   Delete

Create From the list

Home
Events
Settings
Sign out

Desktop HD

FeedBack System - Create   Done

Section 1   Edit   Delete

Create From the list

Question 1
Question 2
Question 3
Custom Quesiton

Home
Events
Settings
Sign out
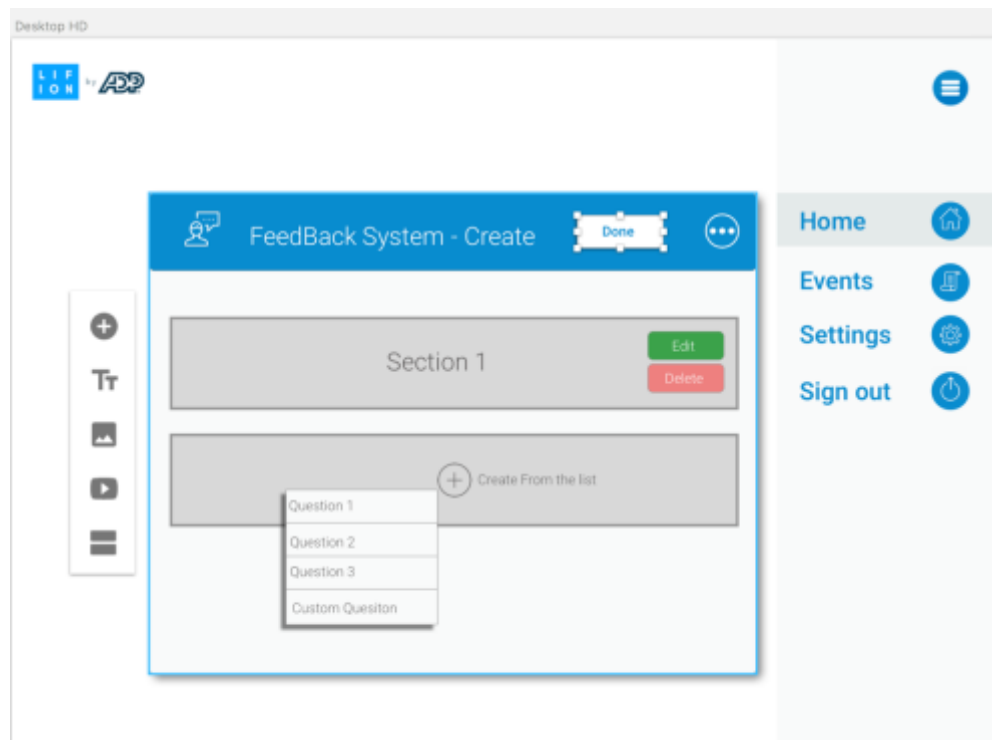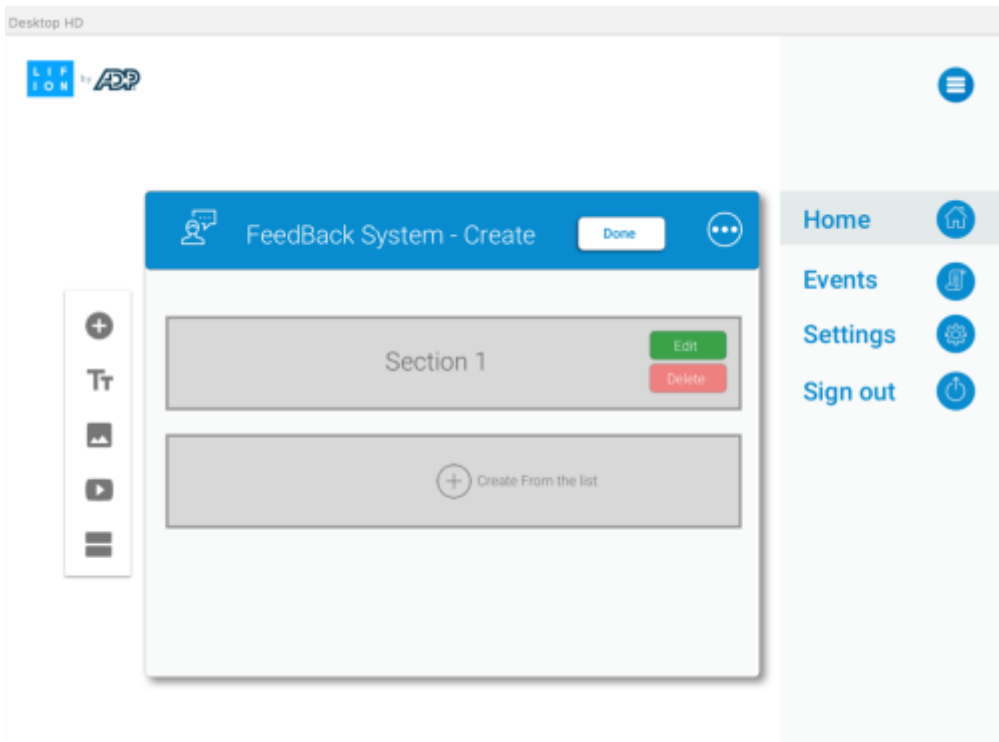
New Feedback Form Creation Screen

Case Study I
Date: 11/04/2016
Name: Rahulkumar Gaddam
Email ID: rgaddam@syr.edu

Send Feedback Request Screen

Desktop HD

Give FeedBack Screen

Case Study I
Date: 11/04/2016
Name: Rahulkumar Gaddam
Email ID: rgaddam@syr.edu

LIF ION by ADP

u Page
Desktop HD

LIF ION by ADP

FeedBack System

Home 🏠

Events

Settings ⚙

Sign out ⏻

Select a FeedBack :   Select Option

Overall Score : 7   Select Range 📅

| > Alan Turing | | | Last Submitted on:11/04/2016 |

| v Rahulkumar | | | Last Submitted on:11/04/2016 |

Punctuality	Right on time	LeaderShip	Excellent	Select Range 📅
Helpfulness	Good	Motivator	Excellent	Score 10
Innovations	Excellent	Team Spirit	High	

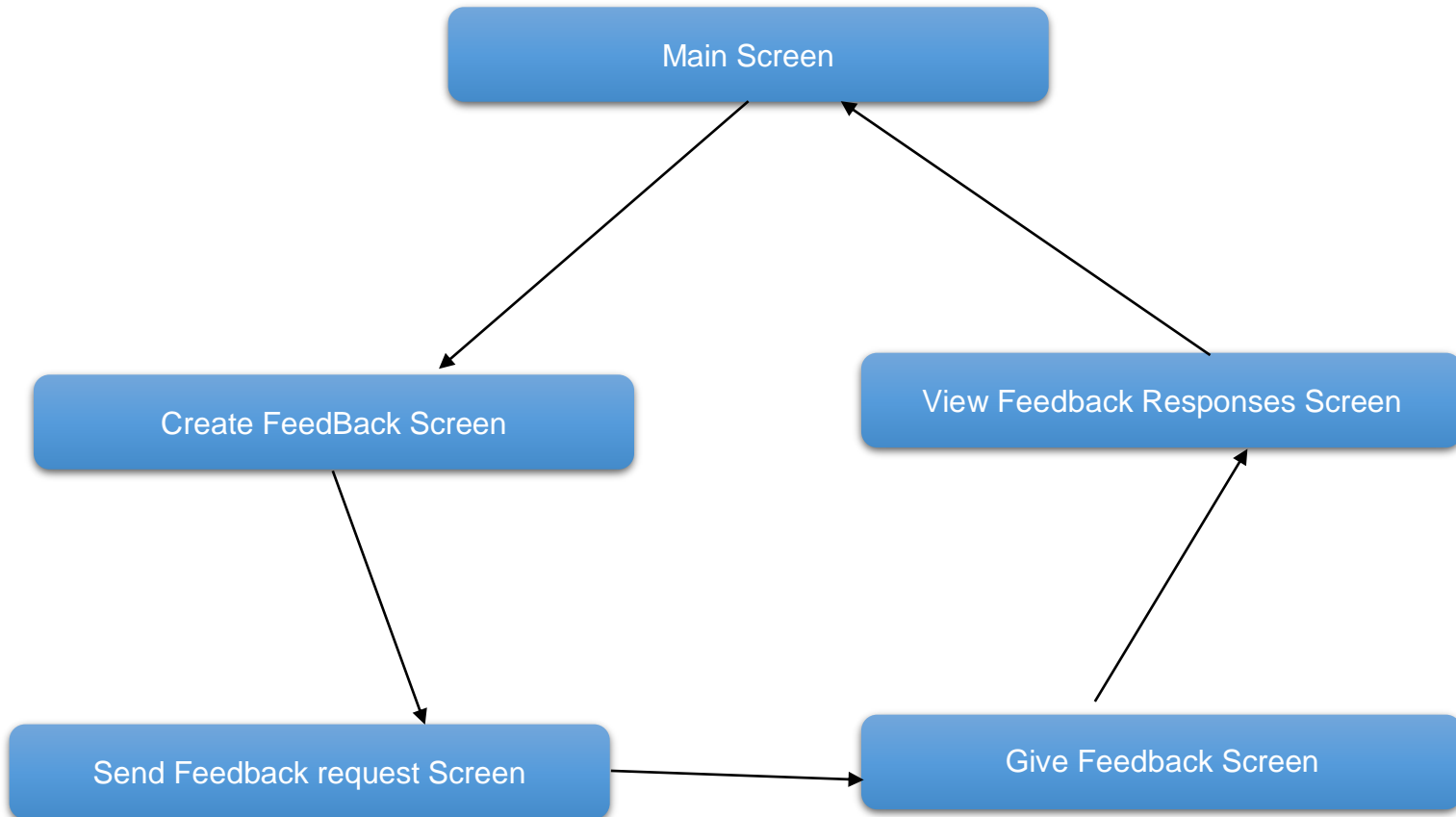| > Anonymous | | | Last Submitted on:11/04/2016 |

View Responses Screen

Case Study I
Date: 11/04/2016
Name: Rahulkumar Gaddam
Email ID: rgaddam@syr.edu

Application Flow: This is ideal work flow but Main Screen is accessible form any screen so that we can access all other three screens from the current screen.

```
                        ┌─────────────────────┐
                        │    Main Screen      │
                        └─────────────────────┘
                         ↙                 ↖
        ┌──────────────────────┐      ┌──────────────────────────────┐
        │ Create FeedBack Screen│      │ View Feedback Responses Screen│
        └──────────────────────┘      └──────────────────────────────┘
                  ↓                                    ↑
   ┌────────────────────────────┐      ┌──────────────────────┐
   │ Send Feedback request Screen│ ──→ │  Give Feedback Screen │
   └────────────────────────────┘      └──────────────────────┘
```

Main Screen

Create FeedBack Screen

View Feedback Responses Screen

Send Feedback request Screen

Give Feedback Screen

Case Study I
Date: 11/04/2016
Name: Rahulkumar Gaddam
Email ID: rgaddam@syr.edu

Third Question:

SELECT FD .Questions, FD .Response,
IsNull(Convert(varchar(50),submitter.SubmitterName),"Anonymous") AS
submitter.SubmitterName FROM Feedbackformdata FD LEFT JOIN submitter ON
FD.SubmitterID = submitter.SubmitterID AND FD.Feedbackdataid = SURVEYID ;

Else we can have custom entry in the submitter database for submitter ID in case of anonymous entry