Kirkland Signature Online Survey Tool

Team Costco

Team Members

Team Lead: Josh Pfeffer

Technical Lead: Jeremy Koletar

Secretary: Joe Thomas

Documentation: Tim Wong

QA: Patrick Cook

Vision & Scope

XYZ Corp. uses a variety of survey tools

Existing tools fall short

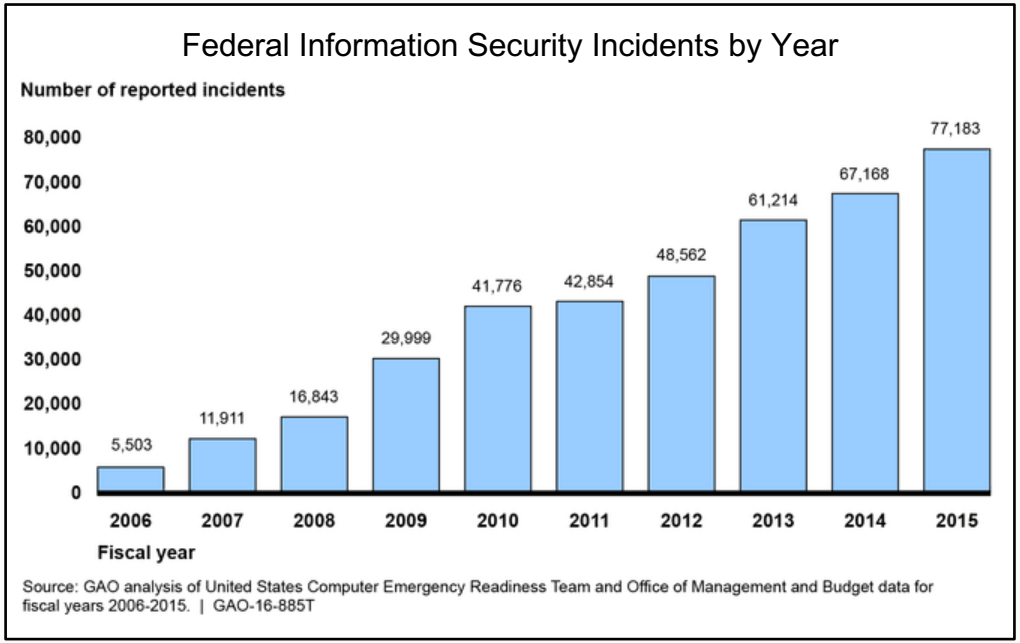
- Remotely Hosted - Security
- No Support for Conjoint Analysis

Vision & Scope

Information Security is a perpetually growing concern.

XYZ. Corp's end product is data.

Ensuring the security of that product is paramount.



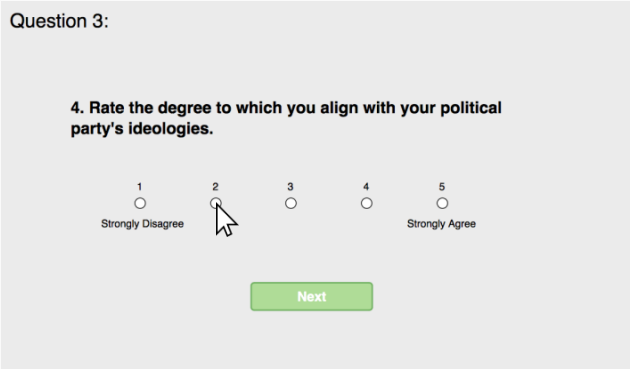
Major Requirements

Functional Requirements

Standard Survey Tool Features

- Easy creation of surveys
- Support for a variety of question types
- Manage lists of potential respondents
  - Control who may respond to a given survey
- Question Branching
- Analytics

Secure access to survey data by XYZ Corp. employees



Functional Requirements

Conjoint analysis is the pivotal functional requirement for this project.

- Generated conjoint trade-off questions
- Manually specified conjoint trade-off questions
- Computation of preference values

“Conjoint analysis is marketers' favorite methodology for finding out how buyers make trade-offs among competing products and suppliers.” [1]

1: Green, Paul E., Abba M. Krieger, and Yoram Wind. "Thirty Years of Conjoint Analysis: Reflections and Prospects." Interfaces 31.3 - Supplement (2001): 56-73. Web.

Non-Functional Requirements

Performant at scale:

- Up to 10 simultaneous open surveys
- Respondents tend to respond within the first 2 hours of solicitation
- System must be **performant** with **10,000** simultaneous users
- **Performant:** response times for HTTP requests *should* remain below 100 milliseconds, and *must* remain below 200 milliseconds.

Secure:

- Purpose of project is to keep data in the custody of XYZ Corp
- SEI CERT's *Top 10 Secure Coding Practices* will be taken into account during code reviews

Major Use Cases

UC-7: Set up branching questions

Actors: Survey Creator

Goals: Allow a survey creator to add branching logic to a question

Precondition: User editing a question

Survey Creator adds branching logic to a particular pivot question (the location from where the respondent will branch) by ticking a checkbox. Pivot questions cannot be located during the conjoint segment.	View is updated with parameter fields that will be filled and used for branching logic
Survey Creator adds a conditional expression row to the table	A new empty row is created
Survey Creator can reorder Conditional Expressions, and can chain expressions to require multiple conditional expressions to be satisfied before branching to a new question.	

UC-3.7: Add a manual conjoint trade-off question to the survey

Actors: Survey Creator

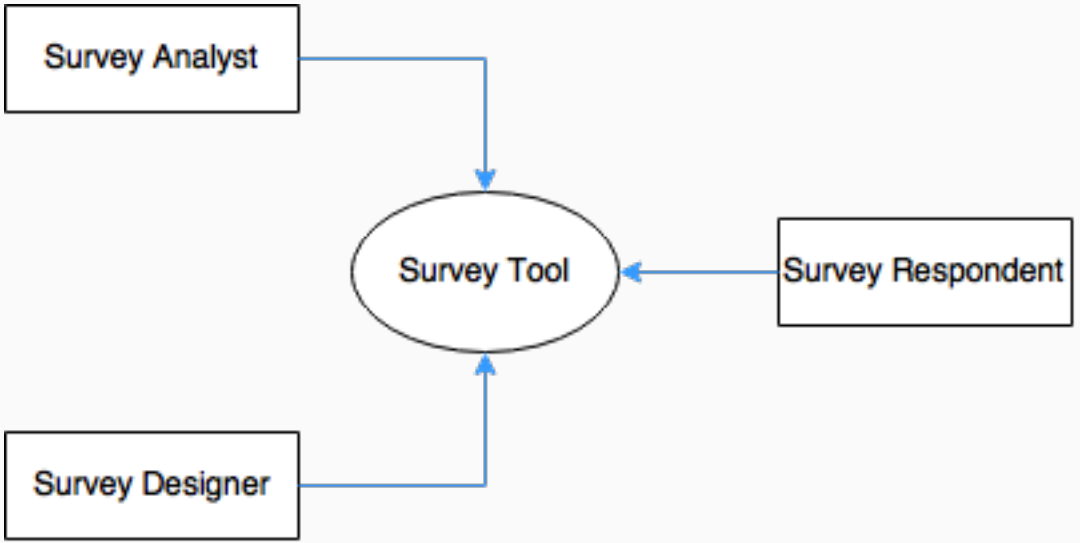
Goals: Add a manual conjoint trade off question to the survey.

Precondition: User editing a survey

User clicks “Add Question”	“Add Question” dialog opens, listing available types of questions
User selects question type	Format for respective question becomes available to be filled out
User selects number of options (2-3) offered to survey taker	The system renders 2-3 panels which contain input boxes where attributes and attribute levels can be defined by the survey creator
User adds attribute and defines attribute level	The system generates conjoint trade off questions with varying attribute values which are calculated based on the attribute level entered by the user upon question creation.

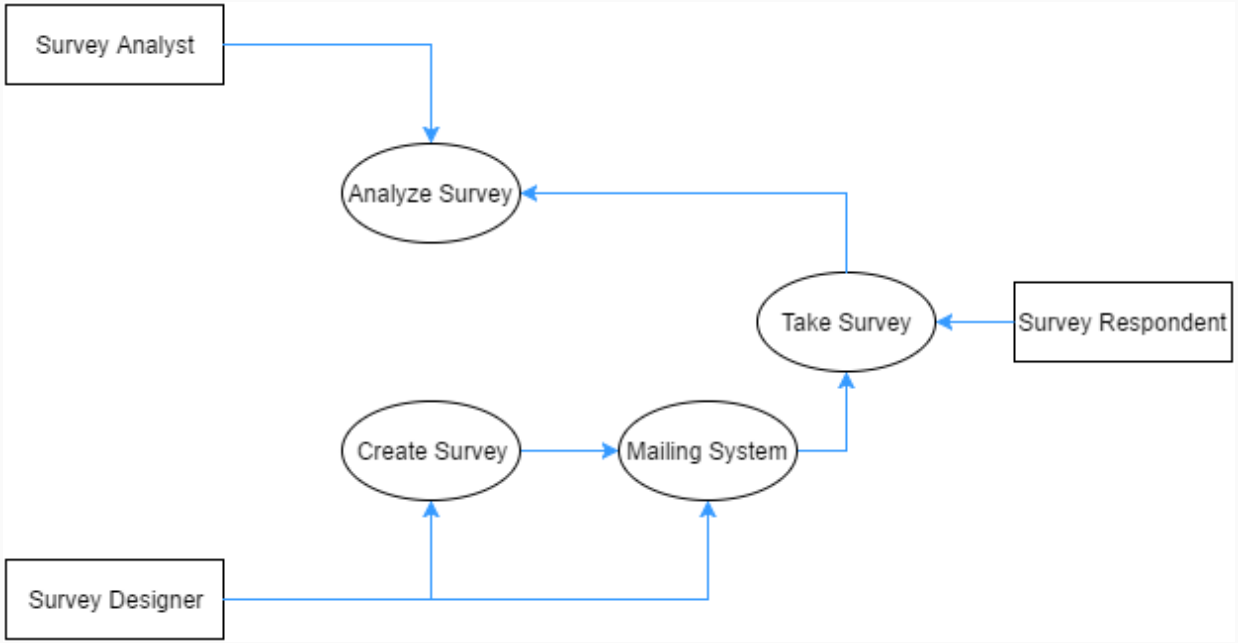
Data Flow Diagrams

Level 0 DFD:



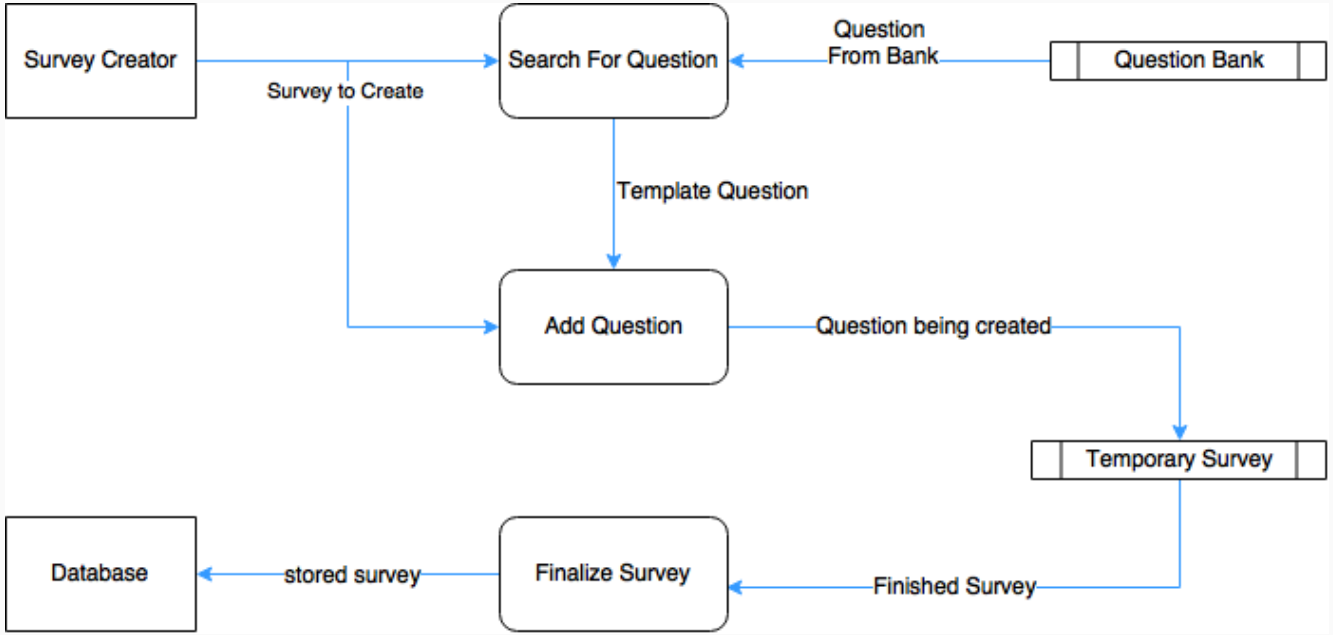
Data Flow Diagrams cont.

Level 1 DFD:



Data Flow Diagrams cont.

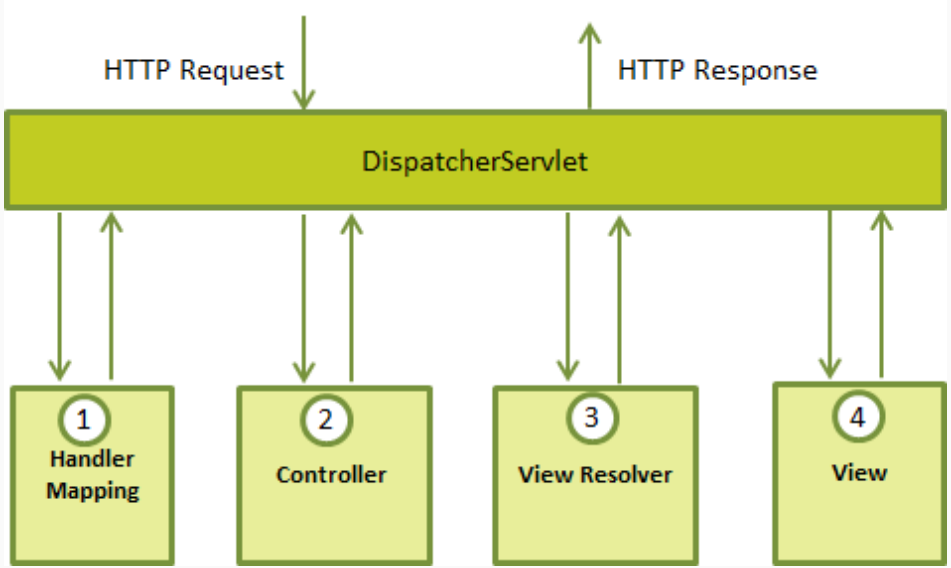
Level 2 DFD: process of creating survey



Design

Architecture Diagram

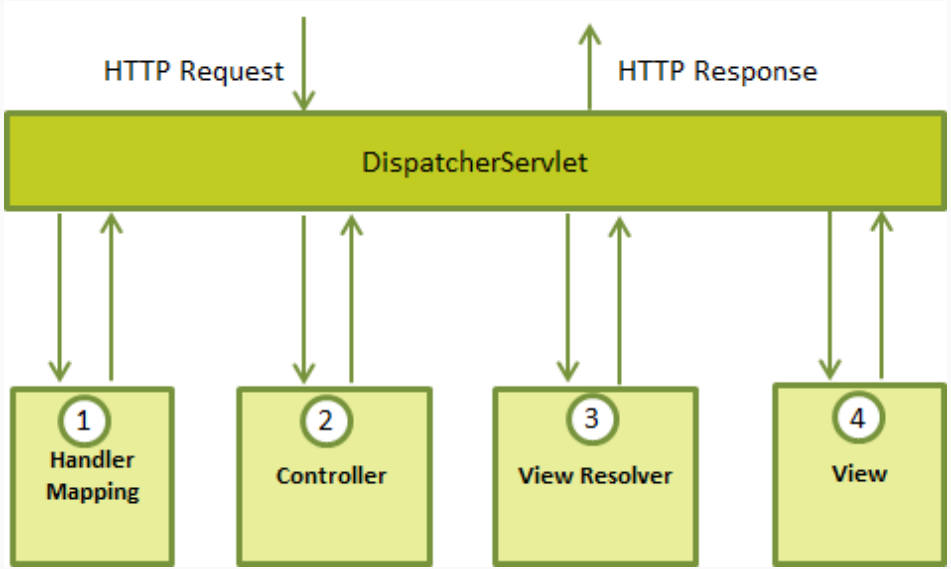
- The Spring Model-View-Controller (MVC) framework will be used
- The DispatcherServlet handles all HTTP requests and HTTP responses
- A controller is chosen using the HandlerMapping and the DispatcherServlet is notified



https://www.tutorialspoint.com/spring/images/spring_dispatcherservlet.png

Architecture Diagram cont.

- The controller calls service functions to set the model data and returns a view name
- The View Resolver uses the view name to determine the view to be used
- The model data is passed to the View which then renders it to the user



https://www.tutorialspoint.com/spring/images/spring_dispatcherservlet.png

Architecture Diagram cont.

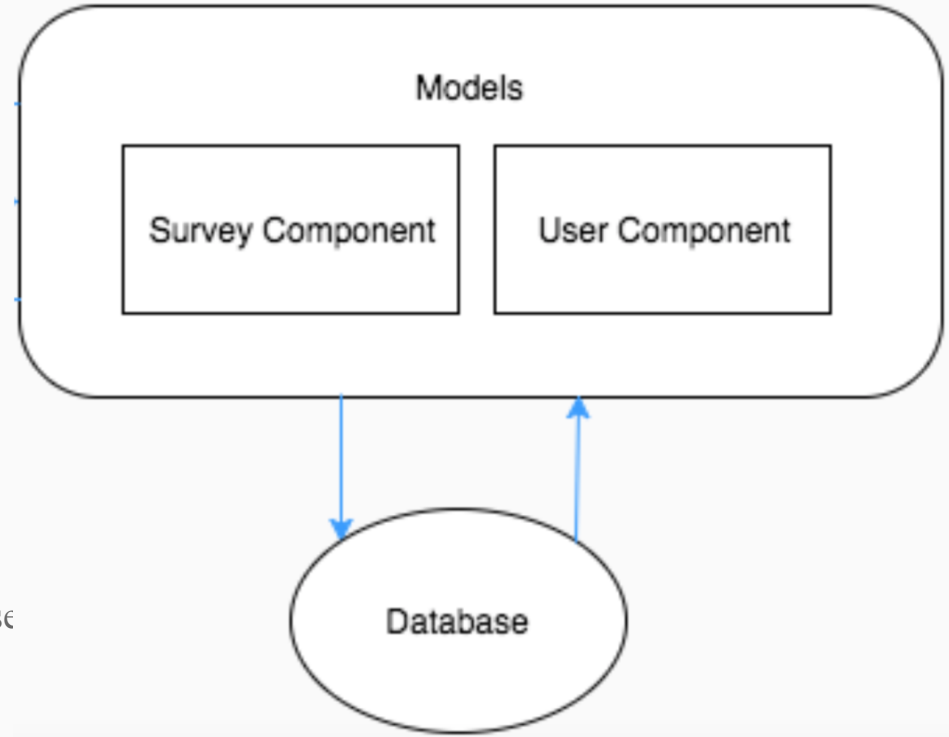
Models

The models are split into two groups:

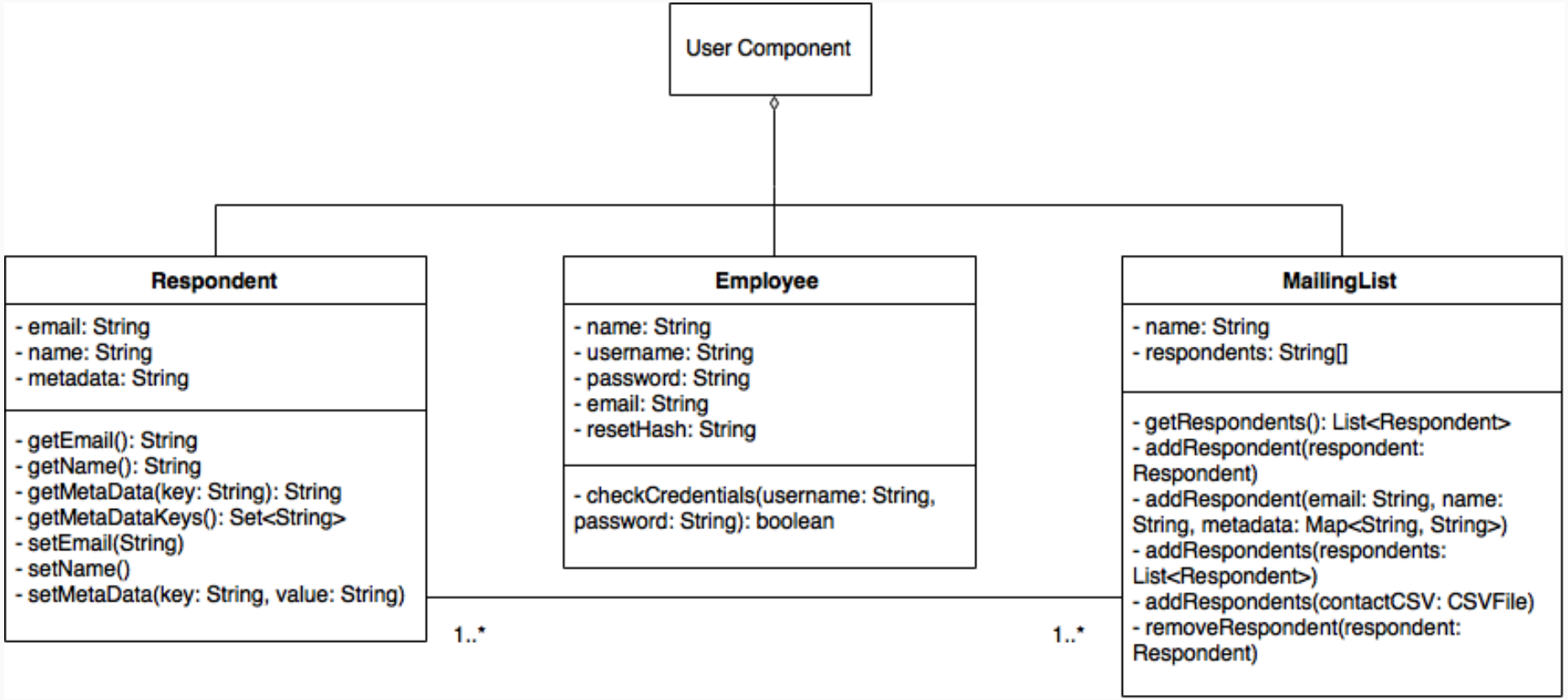
- The survey component:
  - Survey, questions, responses, question bank

- The user component:
  - Users, employees, respondents

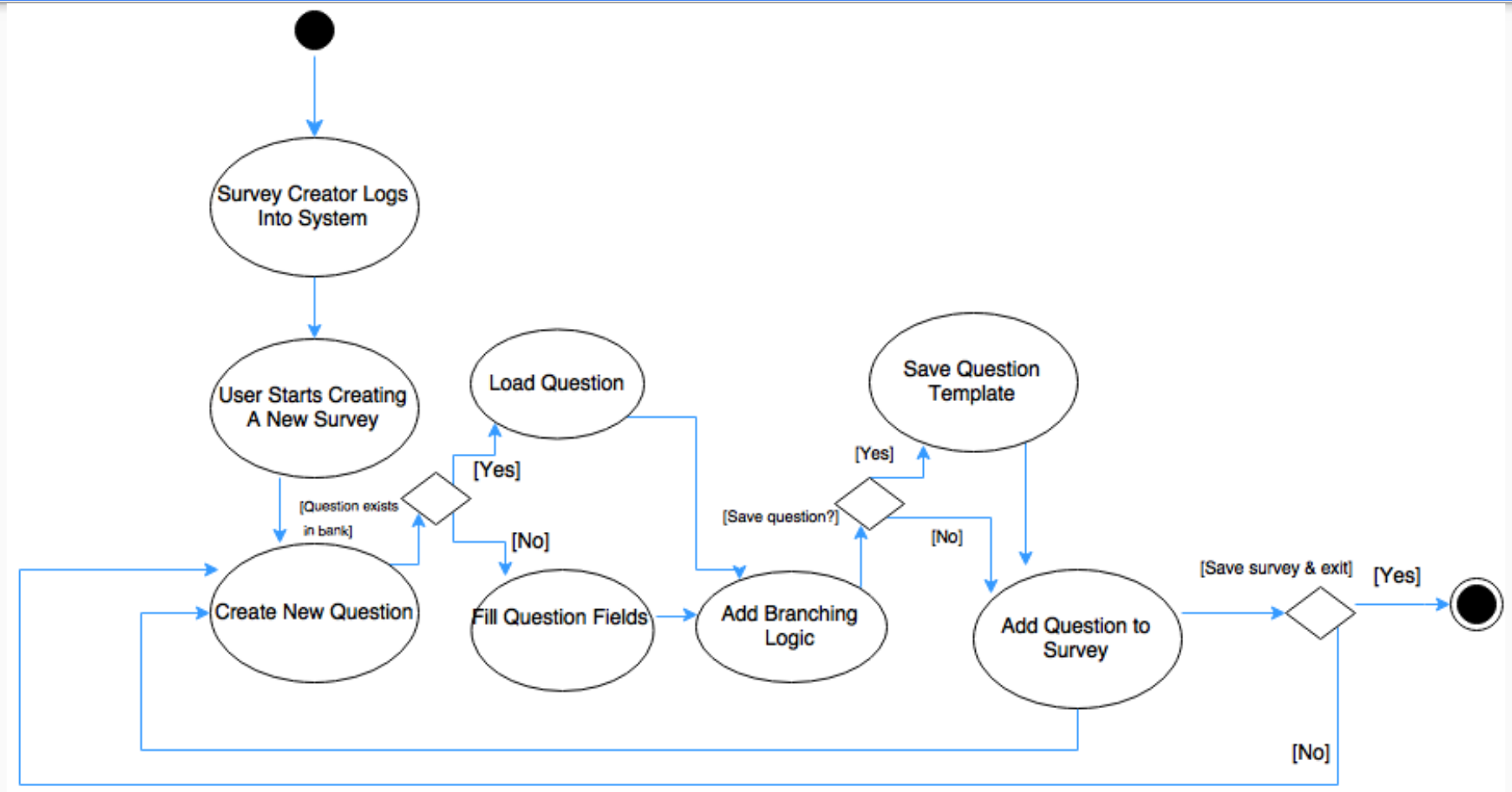
These models will interact with our mySQL database where all survey data is held



Class Diagram - User Component

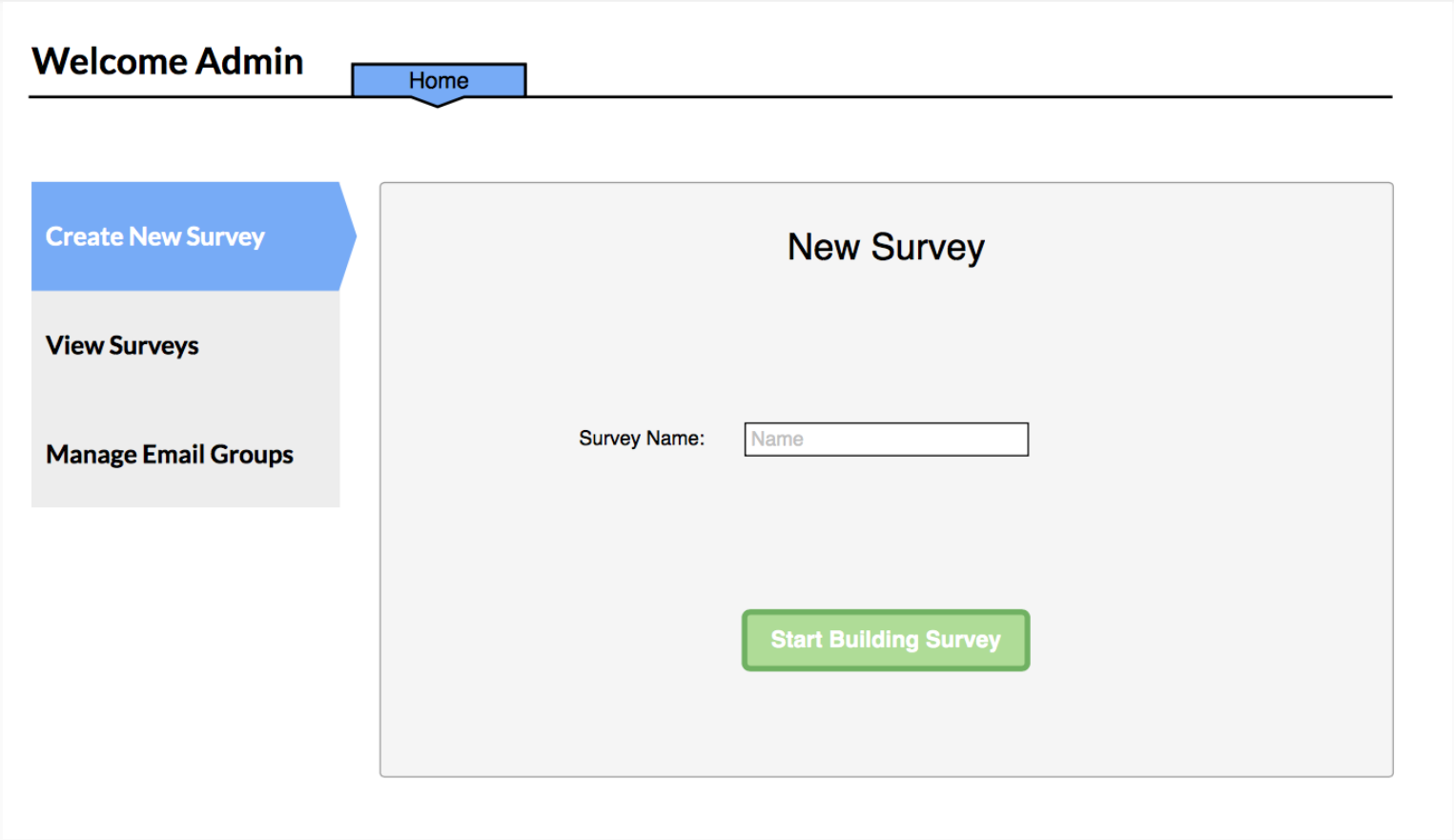


Activity Diagram - Creating a Survey

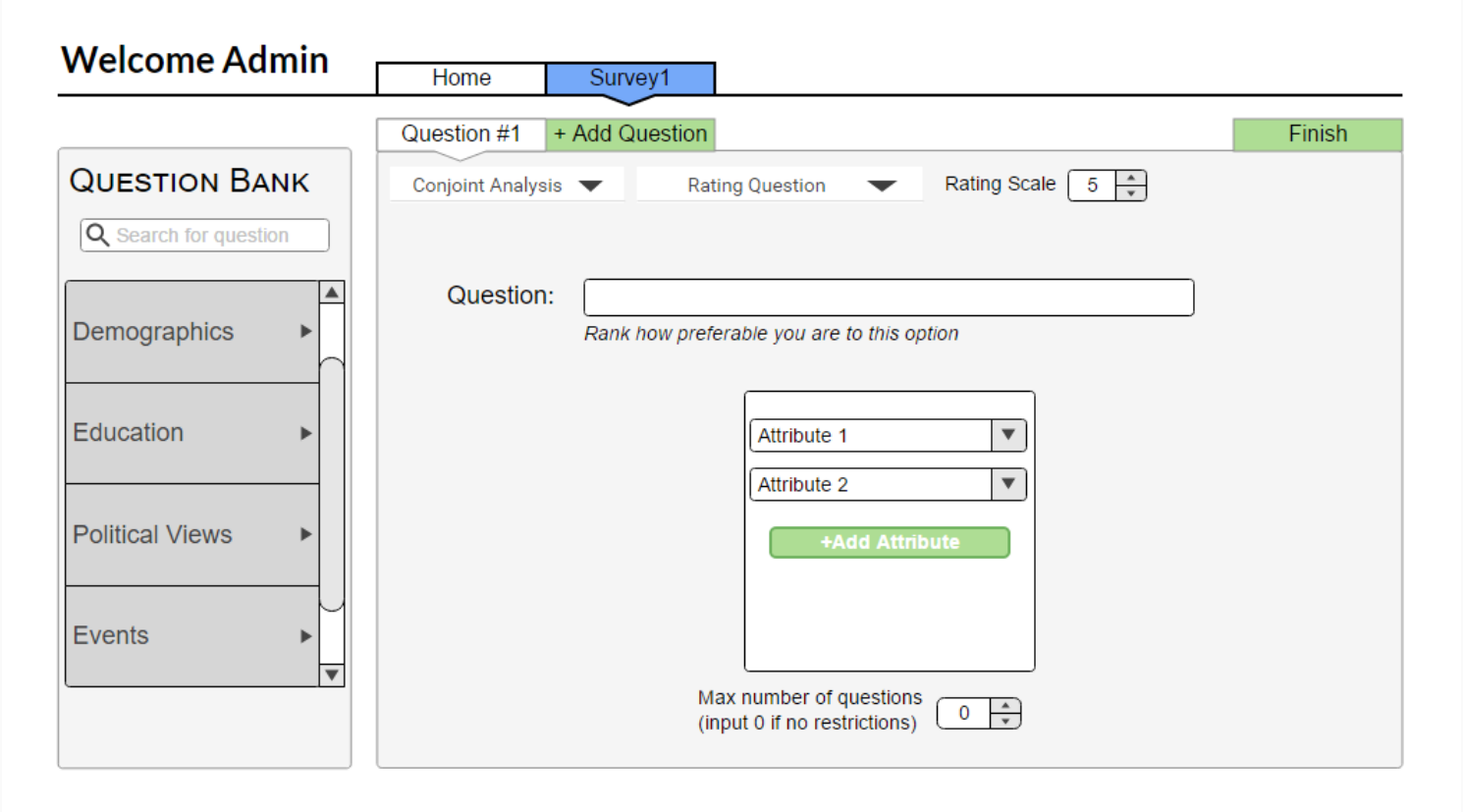


User Interface

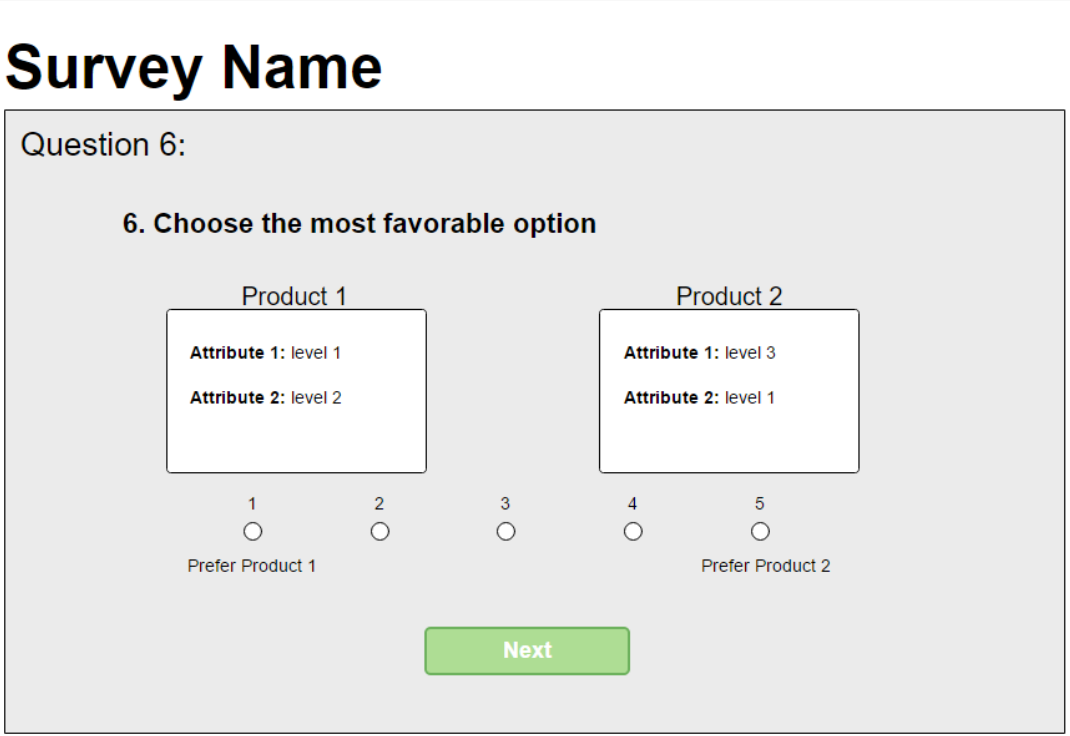
Home Screen


Welcome Admin
Home
Create New Survey
View Surveys
Manage Email Groups
New Survey
Survey Name:
Name
Start Building Survey

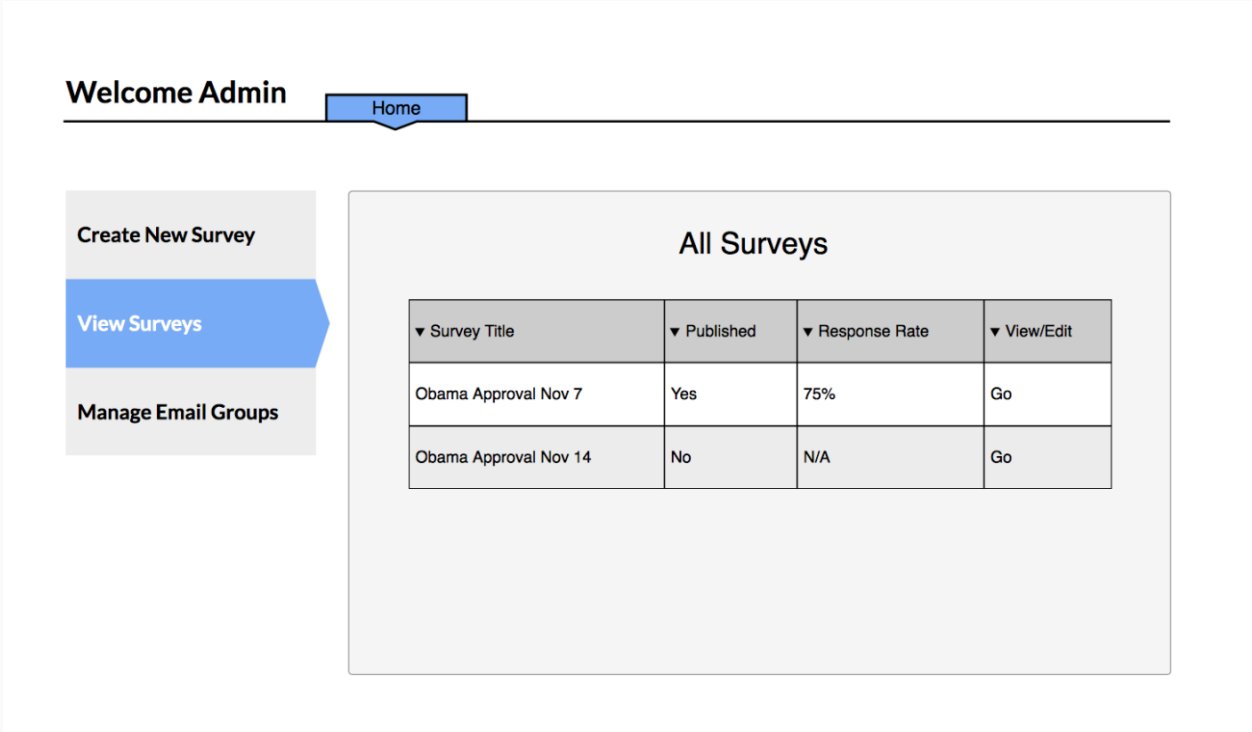
Adding Conjoint Trade off Questions (Engine Created)



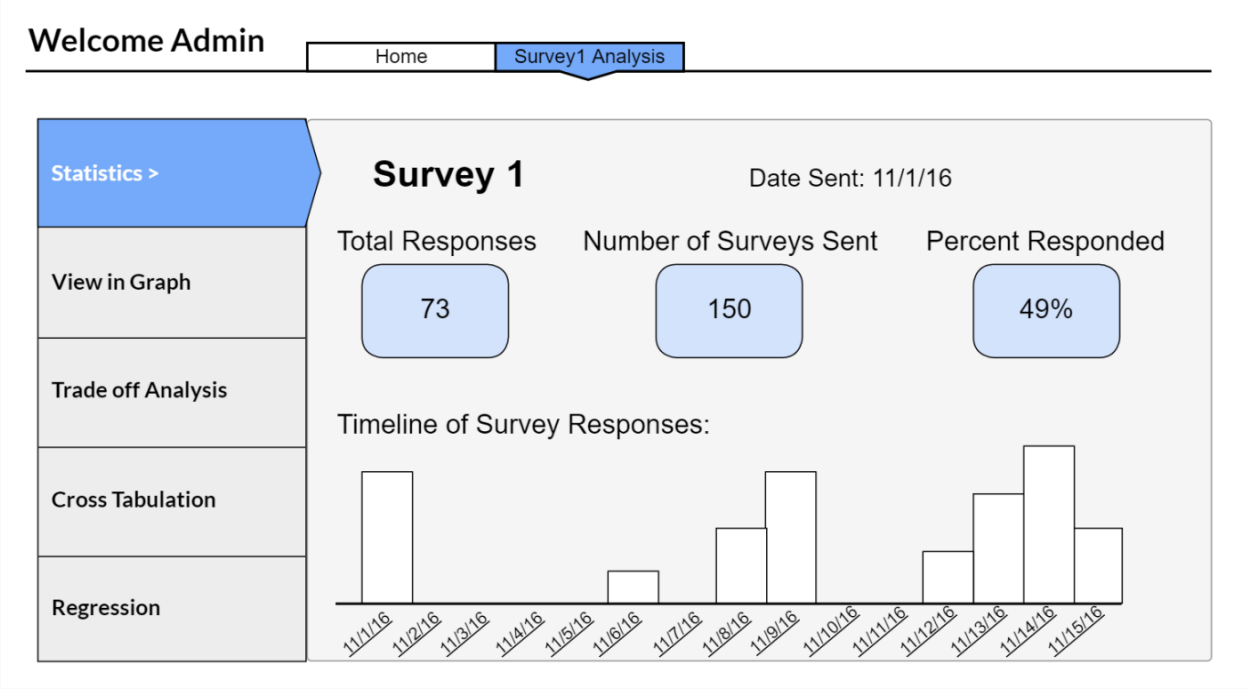
Example Engine Created Conjoint Analysis Question



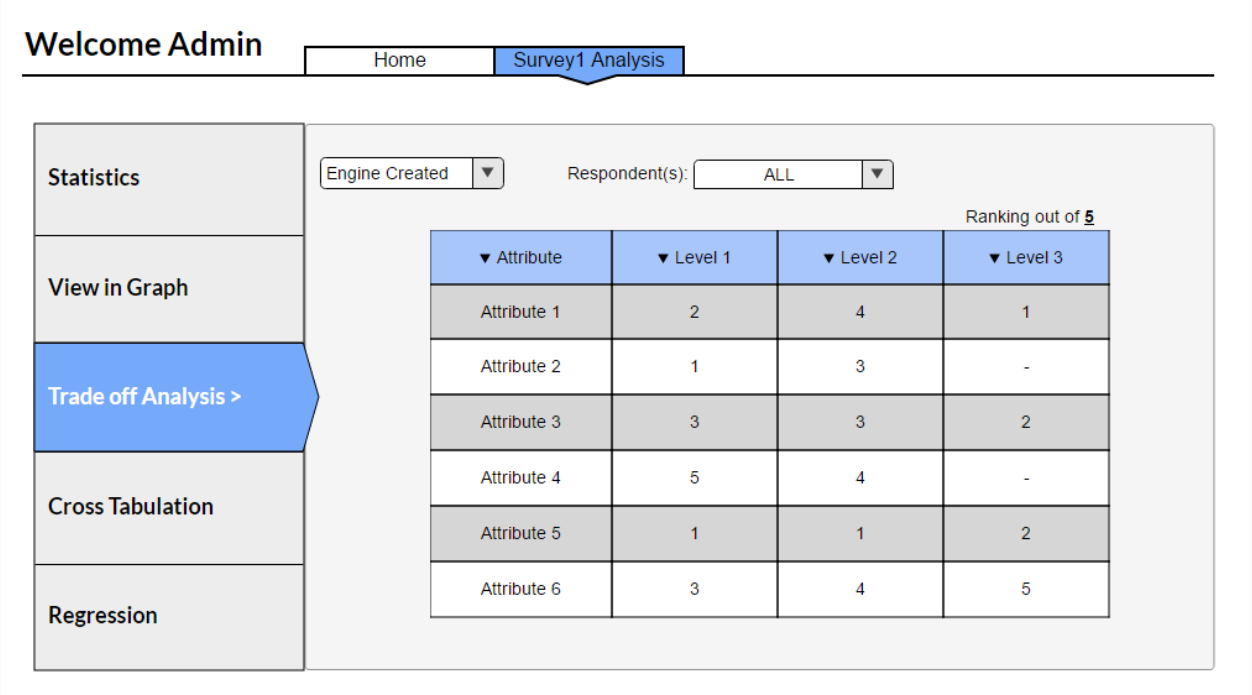
View Survey Analysis



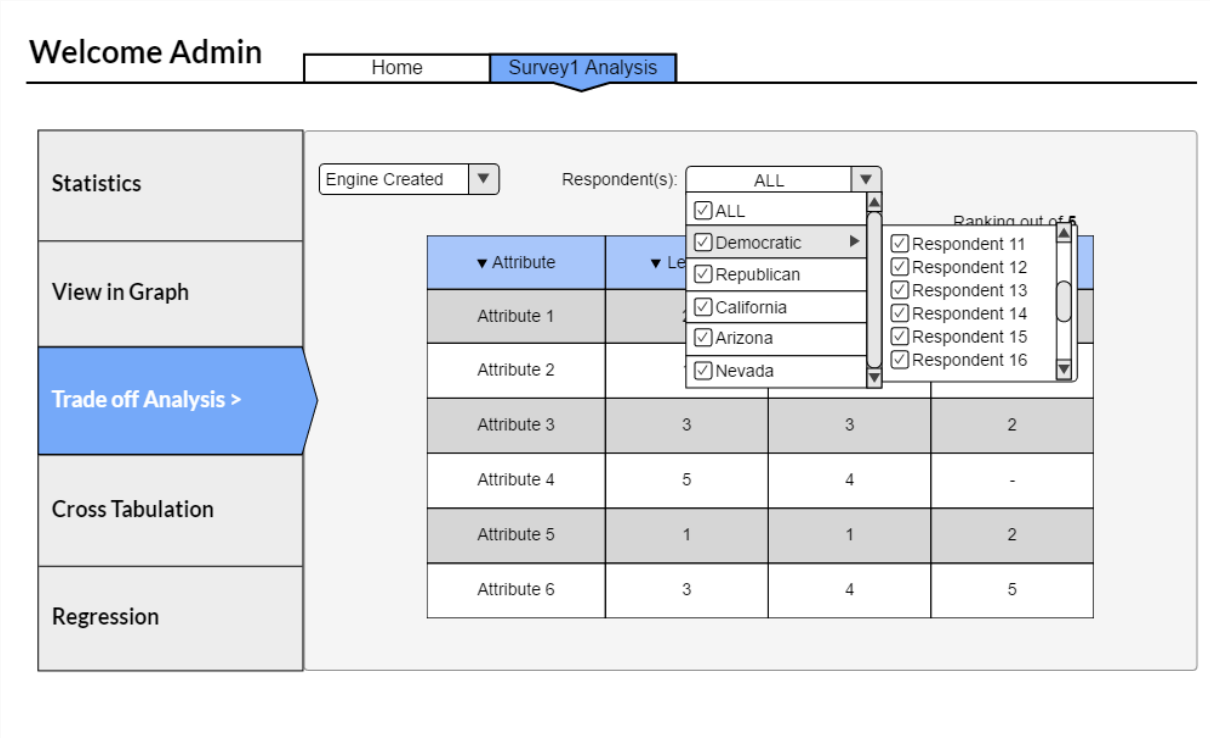
Analysis Main Page



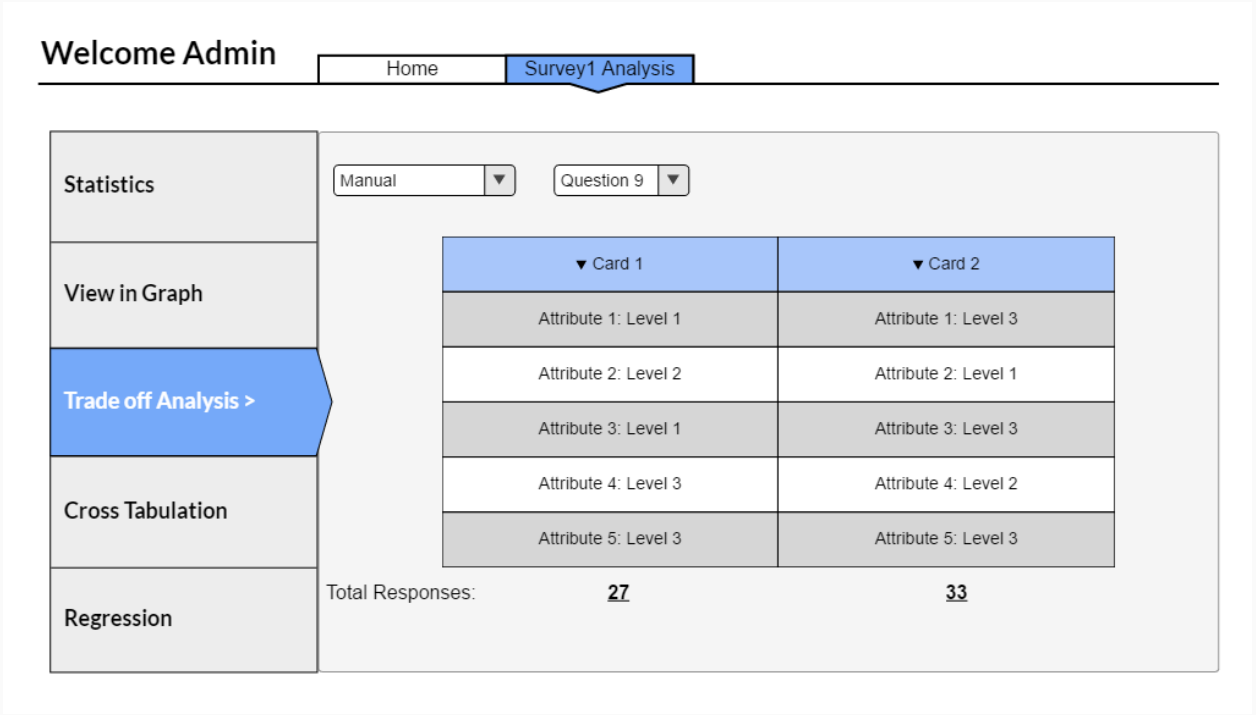
View Trade-Off Analysis



View Trade-Off Analysis (Cont.)



View Trade-Off Analysis (Cont.)



Testing Strategy

Approach

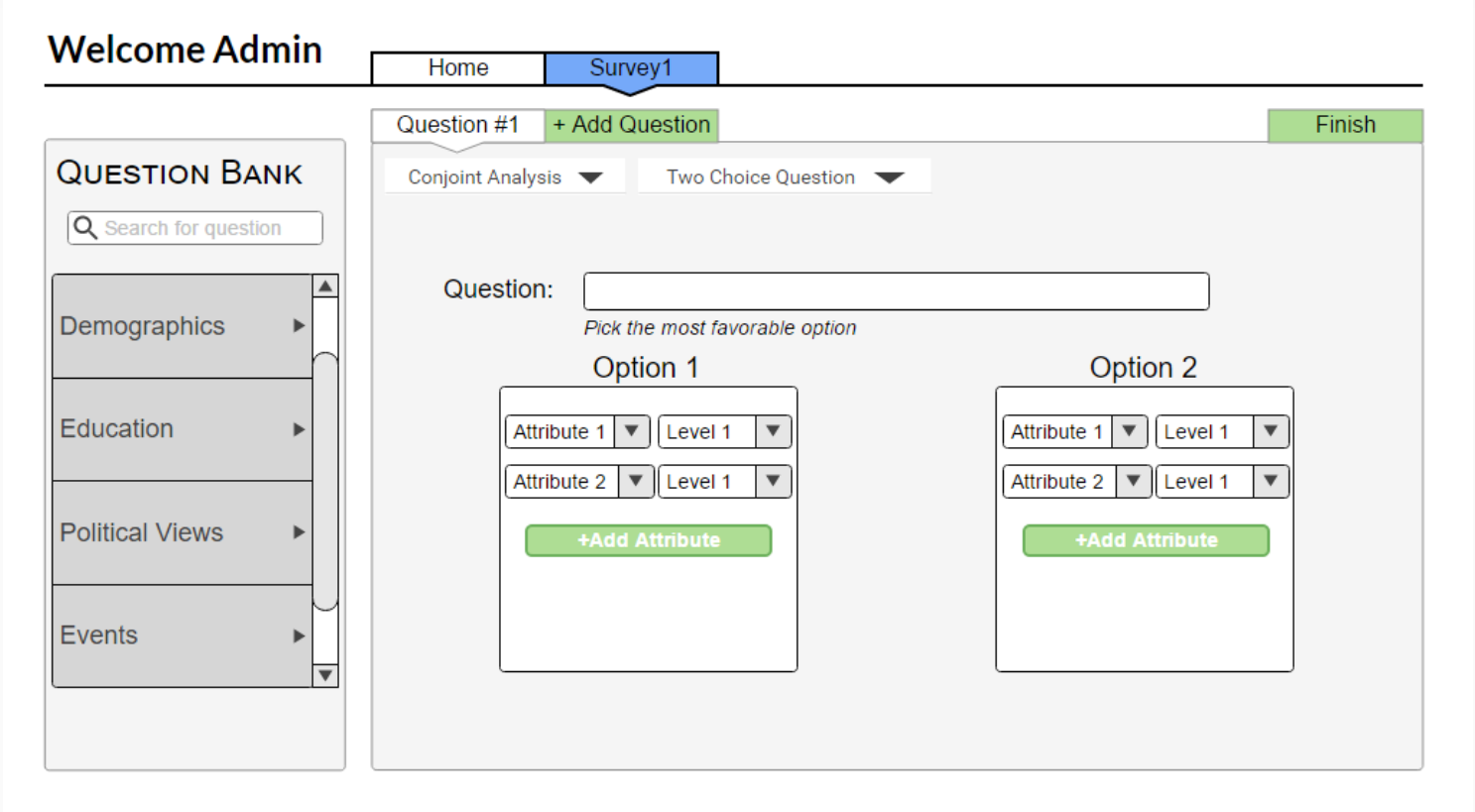
- Manual Testing
  - Manual Conjoint Analysis Question Creation
  - Branching Logic Creation and Behavior
  - Cross Tab Analysis
- At least 5 runs each for the manual tests
- Performance and Stress Testing
  - Use of scripts

Test Case - Manual Conjoint Analysis Question Creation

Test conditions:

- User inputs attributes correctly
- User does not specify attribute(s)
- User does not specify level(s)

Adding Conjoint Trade off Questions (Manual)



Wrap-Up

Vision & scope, major requirements, design, user interface, testing strategy

Our tool is different in that it will be hosted in-house and supports conjoint analysis

We hope to have the opportunity to implement and deploy this project

“Great things in business are never done by one person. They’re done by a team of people.” - Steve Jobs

Questions?