**Software Requirements and Design Document**

**For**

**Business Survey Analytics**

Version 1.0

**Authors**:

Samuel Anderson

Michael Fogarty

Joseph Fields

Trey Hamilton

Luis Corps

# Overview (5 points)

*Give a general overview of the system in 1-2 paragraphs (similar to the one in the project proposal).*

We want to provide a business, or any user, with a platform to generate their own survey. For example, the business can create their own personalized survey with specific questions tailored to their business. The platform would then store these results in a database and be able to supply the business with analytics so they can decide what changes should be made to better serve their clients.

# Functional Requirements (10 points)

*List the* ***functional requirements*** *in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

* When entering the app both customers and survey takers will be prompted to login or take a survey, this is our home page (High Priority). *This is the first thing that customers see when entering the web app it is important that the home page is clear and concise.*

* Each page will have a link back to the home page (Medium Priority). *It is Important that customers can easily navigate through the web app without having to relaunch the app to get back to the home page.*

* If users choose to take a survey they will be asked to enter a survey code (Medium Priority). *For this increment our main priority is to have the login menu working properly.*

* When the code is submitted, and if it’s correct, it’ll take them to the survey to complete. (High Priority) *This builds off the last requirement*

* An invalid code will reload the page with a message displayed “invalid code”. (Low Priority) *This builds off the last requirement would be helpful but not necessary*

* If users choose to login they will be prompted to enter a username(email) however if they don't have an account they will be asked to sign up. Once email is entered they will be sent a code to confirm their account (High Priority) *This is a crucial aspect of the login page given that new customers will not have an account yet need one in order to start creating surveys.*

* New users will link them to a registration page where they will be prompted for certain information that will create their account. (High Priority) *This builds off the last requirement and is where the customer's will establish their email, password, and company name.*

* *Users who will chose to create a survey will have the ability to create unique multiple choice question.*(High Priority)

* Once they have finished creating the survey they be given the option to save and publish after this they will be given their survey code.(High Priority) *this is a core functionality of the web app.*

* If a company/user has pre-existing surveys they can view the results for their surveys and see what the most popular answer choice for each question is. (High Priority) *without this feature companies would have no way of interpreting the results of their surveys which would make the creation of a survey borderline useless*

* No two Accounts can have access to the same survey, all costumes accounts are independent of each other.

***\*NOTE***

*Survey Takers do not need an account to take a survey, only a survey code.*

*Also if Auth0 cannot be successfully implemented then users will instead be*

*prompted to create a password instead of entering code, this is the only thing*

*that would change.*

# Non-functional Requirements (10 points)

*List the* ***non-functional requirements*** *of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

* Home page will take no longer than 2 seconds to load.

* To move from one page back to another ex. sign up page **-->** home page should also take no longer than 2 seconds to load.

* Database will store a unique token from the user to determine if they have an account. *We plan to use Auth0 to ensure a secure login for customers instead of rolling out our own, which due to our lack of experience would likely be much less secure.*

* Login Information will take no longer than 3 seconds to be sent to the database

* Every user will only have access to their own information.

* Login page is secured using Auth0.

# Use Case Diagram (10 points)

*This section presents the* ***use case diagram*** *and the* ***textual descriptions*** *of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*

***Textual descriptions of use cases****: For the first increment, the textual descriptions for the use cases are not required. However, the textual descriptions for all use cases discovered for your system are required for the second and third iterations.*

*A black sign with white text

Description automatically generated*

# Class Diagram and/or Sequence Diagrams (15 points)

*This section presents a high-level overview of the anticipated system architecture using a* ***class******diagram*** *and/or* ***sequence diagrams****.*

*If the main* ***paradigm*** *used in your project is* ***Object Oriented*** *(i.e., you have classes or something that acts similar to classes in your system), then draw the* ***Class Diagram******of the entire system and Sequence Diagrams for the three (3) most important use cases in your system.***

*If the main* ***paradigm*** *in your system is* ***not Object Oriented*** *(i.e., you* ***do not*** *have classes**or anything similar to classes in your system) then only draw* ***Sequence Diagrams****,* ***but for all the use cases of your system.*** *In this case, we will use a modified version of Sequence Diagrams, where instead of objects, the lifelines will represent the functions in the system involved in the action sequence.*

***Class Diagrams*** *show the* ***fundamental objects/classes*** *that must be modeled with the system to satisfy its requirements and* ***the relationships*** *between them. Each class rectangle on the diagram* ***must also include the attributes and the methods of the class*** *(they can be refined between increments). All the* ***relationships between classes and their multiplicity*** *must be shown on the class diagram.*

*A* ***Sequence Diagram*** *simply depicts* ***interaction******between objects*** *(or* ***functions -*** *in our case - for non-OOP systems) in a sequential order, i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.*

*A screenshot of a cell phone

Description automatically generated*

# Operating Environment (5 points)

*Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.*

BSA is a web-based application, so it is necessary for the user to have access to the internet during its use. Subsequently, the user must have a web browser that supports HTML5, CSS3, and JavaScript. BSA can operate on any operating system that can meet the web browsing requirements.

# Assumptions and Dependencies (5 points)

*List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.*

Assumptions

* We are using Auth0 for our passwordless setup, so the ability of our users to gain access to our service is dependent on that.

Dependencies

* Currently hosting database on a personal machine, which must be functioning properly to send and receive data.