

Use Cases **for** **<Whoogle>** **<version 2>**

**Prepared by Keyan Vakil, John Tokarowski, Omayr Abdelgany, Chang
Gao**

Group 5

10/5/16

1. Guidance for Use Case Template

Document each use case using the template shown in the Appendix. This section provides a description of each section in the use case template.

2. Use Case Identification

1.1. Use Case ID

Give each use case a unique integer sequence number identifier. Alternatively, use a hierarchical form: X.Y. Related use cases can be grouped in the hierarchy.

1.2. Use Case Name

State a concise, results-oriented name for the use case. These reflect the tasks the user needs to be able to accomplish using the system. Include an action verb and a noun. Some examples:

- View part number information.
- Manually mark hypertext source and establish link to target.
- Place an order for a CD with the updated software version.

1.3. Use Case History

1.1.1. Created By

Supply the name of the person who initially documented this use case.

1.1.2. Date Created

Enter the date on which the use case was initially documented.

1.1.3. Last Updated By

Supply the name of the person who performed the most recent update to the use case description.

1.1.4. Date Last Updated

Enter the date on which the use case was most recently updated.

3. Use Case Definition

1.1. Actors

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.

Use Cases for <Whoogle>

1.2. Trigger

Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.

1.3. Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

1.4. Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User's identity has been authenticated.
2. User's computer has sufficient free memory available to launch task.

1.5. Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

1.6. Normal Flow

Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description. This description may be written as an answer to the hypothetical question, "How do I <accomplish the task stated in the use case name>?" This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered "X.0", where "X" is the Use Case ID.

1.7. Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form "X.Y", where "X" is the Use Case ID and Y is a sequence number for the alternative flow. For example, "5.3" would indicate the third alternative flow for use case number 5.

1.8. Exceptions

Describe any anticipated error conditions that could occur during execution of the use case, and define how the system is to respond to those conditions. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form "X.Y.E.Z", where "X" is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, "E" indicates an exception, and "Z" is

Use Cases for <Whoogle>

a sequence number for the exceptions. For example “5.0.E.2” would indicate the second exception for the normal flow for use case number 5.

1.9. Includes

List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality.

1.10. Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

1.11. Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

1.12. Business Rules

List any business rules that influence this use case.

1.13. Special Requirements

Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.

1.14. Assumptions

List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.

1.15. Notes and Issues

List any additional comments about this use case or any remaining open issues or TBDs (To Be Determineds) that must be resolved. Identify who will resolve each issue, the due date, and what the resolution ultimately is.

Use Case List

<i>ID</i>	<i>Primary Actor</i>	<i>Use Case Title</i>
1	User	Search for another user
2	User	Choose a user from the results page
3	User	View profile and move to relationship or interest based maps
4	User	Interest & Friends Map(BU people only)
5	User	Map friend relationships based off of user filters

Use Case Template

Use Case ID:	3		
Use Case Name:	View profile and move to relationship or interest based maps		
Created By:	Everyone	Last Updated By:	Keyan
Date Created:	10/2/16	Date Last Updated:	10/4/16

Use Cases for <Whoogle>

Actors:	End user
Description:	The user will be shown a detailed description of the profile selected, including general profile information as well as hobbies and interests.
Trigger:	Selecting an option from the search results page
Preconditions:	User has selected a profile.
Postconditions:	Program generated profile which displays information on selected individual and provides links to the relationship and interest maps
Normal Flow:	<p>3.1 User views the profile page of the selected person which summarizes the selected individual's top social media discussion topics, analysis on the individual's grammar determined by past posts, a photo of the individual if available, several recent posts by the individual.</p> <p>3.2 User can choose to proceed to a relationship map showing connections between the social media friends of the selected individual</p> <p>3.3 User can choose to proceed to a map of BU users clustered by one of the top interests based on the selected individual's topics of discussion from social media</p>
Alternative Flows:	User can also access the profile page from the friends list of another user profile or from the relationship/interest map. User can also go back to the results page if they found the wrong person.
Exceptions:	Website API's may be down -> We will inform the user of the problem.
Includes:	Interest & Friends Map(BU people only), Map friend relationships based off of user filters,
Priority:	High
Frequency of Use:	This use case will be used every time a user clicks on a person

Use Cases for <Whoogle>

	that shows up in the list on the previous page, and thus should theoretically be used almost every time a person reaches the page.
Business Rules:	Should not violate the rules accompanied by the usage of the APIs to be consumed.
Special Requirements:	<p>The additional requirements to ensure proper functionality of this use case entails making sure that the number of API calls aren't too high. The number of API calls made may depend on how many attributes of a person we want to analyse and how many APIs we want to use in order to determine these attributes vs how many attributes we want to determine using our own backend.</p> <p>Alternatively, if we decide to store all the data in our database, the information displayed will depend on if we can store an adequate amount of data in the database to provide enough information, and if we'll be able to access and process the data within a sufficient amount of time.</p>
Assumptions:	This usage of this use case depends on the assumption that the individual that the user is trying to find has a Facebook/Twitter account.
Notes and Issues:	<p>Overall, this is where we expect the user to benefit the most from the main functionality of our app, and a lot of work should be put into making the user experience as clean as possible. One possible issue that could arise is that the user can not figure out that they can click the interests to gain more functionality, so it should be made clear that this is possible.</p> <p>The same can be said for the back button and the relationship map.</p>

Revision History

Name	Date	Reason For Changes	Version

Use Case ID:	1		
Use Case Name:	Search for a user or interest		
Created By:	John	Last Updated By:	John
Date Created:	10/30/16	Date Last Updated:	10/30/16

Actors:	End user
Description:	In this use case, the user searches for a person or a hobby that they are interested in. Once the user or hobby is searched, the user is taken to a results page displaying options based on the search.
Trigger:	Opening the application or opting to return to search another user from the results end page
Preconditions:	User arrives at website
Postconditions:	List of results from the search are displayed and provide the user with options to choose from
Normal Flow:	1.1 User opens the application and sees the search screen 1.2 User selects either a person or topic search 1.3 User enters the name of the user or name of the topic they want to find into the search bar 1.4 User presses submit and is taken to the results page
Alternative Flows:	User can arrive at the search screen from the results page, the profile page, or either of the 'maps' by clicking the "Start another Search" button
Exceptions:	Server is unavailable or APIs unavailable

Use Cases for <Whoogle>

Includes:	Results page
Priority:	High
Frequency of Use:	This use case will be used every time a user arrives at the application
Business Rules:	Should not violate the rules accompanied by the usage of the APIs to be consumed.
Special Requirements:	N/A
Assumptions:	This usage of this use case depends on the assumption that the individual that the user is trying to find has a Facebook/Twitter account.
Notes and Issues:	This page should be very simple and direct the user to enter the required information then move thme to the next screen. This might be a good place to put a quick overview of the app's functionality.

Name	Date	Reason For Changes	Version

Use Case ID:	2		
Use Case Name:	Choose a user or interest from the results page		
Created By:	John	Last Updated By:	John
Date Created:	10/30/16	Date Last Updated:	10/30/16

Actors:	End user
Description:	In this use case, the user selects a user/hobby from the results page based on the keywords that they searched. Once they select the user hobby, they are taken to either the user profile or an interest-based map showing BU students interested in that hobby.
Trigger:	Submitting a username or interest into the search bar
Preconditions:	User has searched for a user or interest

Use Cases for <Whoogle>

Postconditions:	Program generated profile which displays information on selected individual or interest map
Normal Flow:	2.1 User views the results page to determine which of the profiles found is the person is who they are searching for 2.2 User can click on a username to be taken to their detailed profile
Alternative Flows:	User does not find the person searched for and selects the “start another search” button to try again
Exceptions:	Website API’s may be down -> We will inform the user of the problem.
Includes:	User profile, Interest Map
Priority:	High
Frequency of Use:	This use case will be used every time a user submits a search
Business Rules:	Should not violate the rules accompanied by the usage of the APIs to be consumed.
Special Requirements:	The additional requirements to ensure proper functionality of this use case entails making sure that the number of API calls aren’t too high. In this case, the search function should only send the text to search for and receive all the results in one response.
Assumptions:	This usage of this use case depends on the assumption that the individual that the user is trying to find has a Facebook/Twitter account.
Notes and Issues:	This case provides a link between the initial search and any other additional functionality, it has to be easy to navigate/

Name	Date	Reason For Changes	Version

Use Case ID:	4		
Use Case Name:	Interest & Friends Map		
Created By:	John	Last Updated By:	John
Date Created:	10/30/16	Date Last Updated:	10/30/16

Use Cases for <Whoogle>

Actors:	End user
Description:	When a user clicks on one of the “Top Three Interests”, a map will be shown to the user, representing the relationships of people who are interested in such topic in BU. The nodes in the graph represent people in BU who are interested in the selected topic (the selected topic is one of their “Top Three Interests”), the edges are added to the nodes which are friends on Facebook. The graph/map should be updated once per month, including adding or removing nodes and edges, and it will show the last updated date to the user. Using this map, the user should be able to determine which persons/groups should him/her to stalk next.
Trigger:	Selecting the interest & friends map option from a user profile
Preconditions:	User has viewed a detailed profile and clicked on a specific interest
Postconditions:	Click on a specific node and get the profile of that person
Normal Flow:	4.1 User can see the visualized relationships between the B students and their common top interests 4.2 User can go back to the detailed profile page
Alternative Flows:	User chooses to start a new search from the beginning
Exceptions:	Website API’s may be down -> We will inform the user of the problem.
Includes:	Search
Priority:	Medium / High
Frequency of Use:	This use case will only come into play if the user wants to visualize the connections between BU students related to a selected interest
Business Rules:	Should not violate the rules accompanied by the usage of the APIs to be consumed.
Special Requirements:	N/A
Assumptions:	This usage of this use case assumes the API was able to find top interests for the selected user based on their social media.
Notes and Issues:	This will be the most complex portion of the application, and may get less use than the user profile segment.

Use Cases for <Whoogle>

Name	Date	Reason For Changes	Version

Use Case ID:	5		
Use Case Name:	Map friend relationships based off of user filters		
Created By:	John	Last Updated By:	John
Date Created:	10/30/16	Date Last Updated:	10/30/16

Actors:	End user
Description:	<p>The fifth use case of our project entails a relationship map. Once the user has reached Use Case 3, the user can access this use case. The primary actor would be the user. The server would initially provide a "star map" where each node in the map would represent a person, each connection would represent a Facebook friendship, and each color of each node would represent an attribute they have in common. By default, the degrees of friendship should be 2 (meaning that we map all of the friends of the person in question and all of their friends and connections), unless we determine that it would take too long/too many api calls to provide that information. Also, one attribute should be selected by default, which, (for now) represents the university each person attends if they attend a university. Additionally, a user should be able to adjust the degrees of friendship displayed, and other attributes to visualize the data as they please. On this graph, ideally, the user can also click on any individual node in order to provide the profile page of the person clicked.</p>
Trigger:	User selects the relationship map from the profile page
Preconditions:	User is on another user's profile page
Postconditions:	Click on node to be redirected to the person's profile
Normal Flow:	<p>5.1 User views the relationship 'starmap' to visualize connections and gain information about the network</p> <p>5.2 User moves to a user profile by clicking the node</p>
Alternative Flows:	User changes options on the map to display more degrees of friendship

Use Cases for <Whoogle>

Exceptions:	Website API's may be down -> We will inform the user of the problem.
Includes:	Search, API calls
Priority:	Medium
Frequency of Use:	This is essentially the final step of the application, it will only be used once the user completed a search, and moved on to the relationship map
Business Rules:	Should not violate the rules accompanied by the usage of the APIs to be consumed.
Special Requirements:	Avoid sending excessive API calls to improve performance of application
Assumptions:	Assumes the user is a BU student and the application can successfully retrieve the connections of that user and all of the user's friends
Notes and Issues:	N/A

Name	Date	Reason For Changes	Version