Use Cases for

Spatial

Version 1.0
Prepared by Rollin, Arjun, Li-Ke, and Austin
Group 8

Version 1.0 2/21/2016
Version 1.1 3/12/2016
Version 1.2 3/12/2016
Version 1.3 3/13/2016

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.

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 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

ID	Primary Actor	Use Case Title
1.0	App User	Display Map
2.0	App User	User Filters
3.0	App User	User Input
4.0	App User	User Authentication
5.0	App User	Setting
6.0	App User	Create Account
7.0	App User	Delete Account

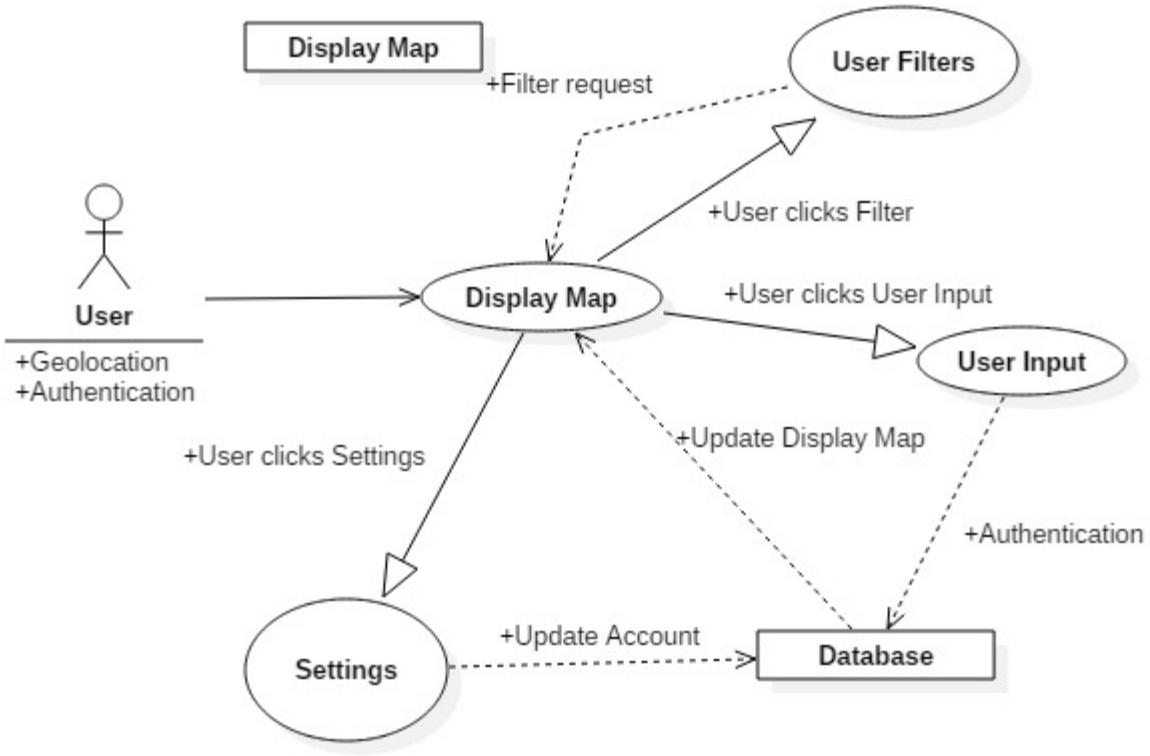
Use Case Template

Use Case ID:	1.0		
Use Case Name:	Display Map		
Created By:	Group 8	Last Updated By:	Rollin
Date Created:	2/21/16	Date Last Updated:	3/12/16

Actors:	App User
Description:	Display map with businesses and events open around the user. User may selects businesses/events that they are interested in and we provide more information about that business/event. If user creates an account, user may input real-time information about a location.

Trigger:	Trigger is the app user accessing website
Preconditions:	1. If user has an account we can authenticate them. 2. We can get geolocation of user 3. Information about businesses/events is available 4. User has internet access or capable device
Postconditions:	1. Map/events are constantly updated 2. Potential user comment/update on location
Normal Flow:	UC1.0: Treat all users the same (no one has an account to view display map). Get geolocation and show businesses/events around the user that are open/will be open soon. If user clicks on event we show them detailed information about business/event.
Alternative Flows:	UC1.1: If the user wants to update information about event/business (example: if the open/close times were wrong) then they will be asked to login in to their account. Upon authentication, user is allowed to update.
Exceptions:	1.0.E.1: If one of the preconditions is not met then website lets the user know which condition(s) they need to meet in order for the website/app to work properly. 1.0.E.2: If preconditions are met and there is an unexpected error then website/app rolls back to previous stable state (for example if the website/app can get geolocation of user but there is an error in the code and the app crashes then we can roll back and display information from previous stable state). 1.0.E.3: If an unaccounted error occurs then website/app gives a generic error message and asks user to reload the website. 1.1.E.1: If user tries to log in but inputs incorrect username and password then website/app tells the user their username/password is incorrect and to re-enter the information. 1.1.E.2: If the user tries to update but does not have an account then website/app will prompt user to make an account.
Includes:	User Input User Filters Settings Authentication
Priority:	Top priority
Frequency of Use:	Once (updates are continuous after user has visited the

	website/app)
Business Rules:	A business's/events decision to make keep their information public. If they decide not to provide information on our sources of data we can't display it on our website/app.
Special Requirements:	The map/event update feature must not lag. The UI is clean and user friendly.
Assumptions:	N/A
Notes and Issues:	Group Decisions: Open issues: basic UI layout of the website TBD: Implementation details.

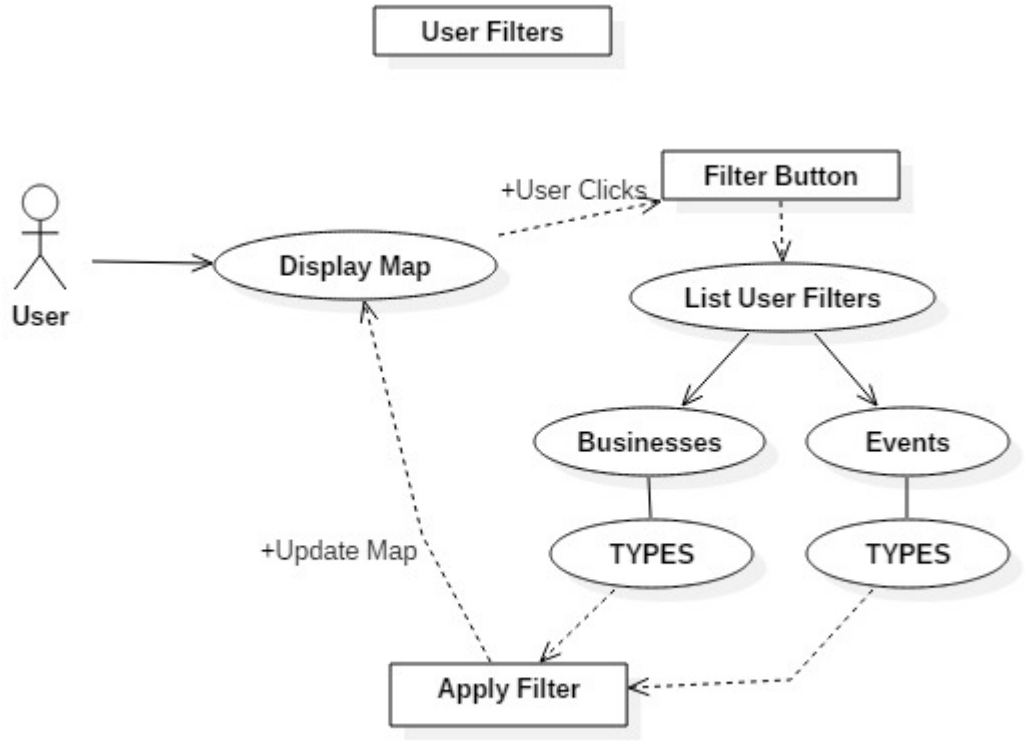


Use Case Template

Use Case ID:	2.0		
Use Case Name:	User Filters		
Created By:	Group 8	Last Updated By:	Rollin
Date Created:	3/12/16	Date Last Updated:	3/12/16

Actors:	App User
Description:	Filter for user manipulation. Will change the default map display to a map based on filters that user chooses. Branch of Display Map UC1.0
Trigger:	Trigger is the app user selecting the filters toolbar and picking a different filter than the default
Preconditions:	1. User submits request for filtered view 2. Receipt of request and response by dataserver
Postconditions:	1. Display Map updated with filter
Normal Flow:	UC2.0: User has access to map and clicks on Filter. Filter bar opens with list of available filters. User clicks on one of these filters and the display map is updated accordingly
Alternative Flows:	UC2.1: If the user did not want to use a Filter, user may click anywhere besides the Filter bar and the Filter bar will close.
Exceptions:	2.0.E.1: If one of the preconditions is not met then website lets the user know which condition(s) are causing no updated display(i.e. Loss of internet connection) 2.0.E.2: If preconditions are met but display map not correctly updated, website/app rolls back to previous stable state (default filter). 2.0.E.3: If an unaccounted error occurs then website/app gives a generic error message and asks user to reload the website. 2.1.E.1: If unaccounted error occurs while user tries to exit Filter, user prompted to reload website

Includes:	Display Map, Various Filter Types
Priority:	Medium Priority
Frequency of Use:	As many times as user selects filters
Business Rules:	A business's/events decision to make keep their information public. If they decide not to provide information on our sources of data we can't display it on our website/app. Thus we cannot then filter the data based on user input.
Special Requirements:	The filter update should not lag.
Assumptions:	List of available filter types are useful to actor
Notes and Issues:	Filter TYPES have yet TBD by team members Default Filter TBD

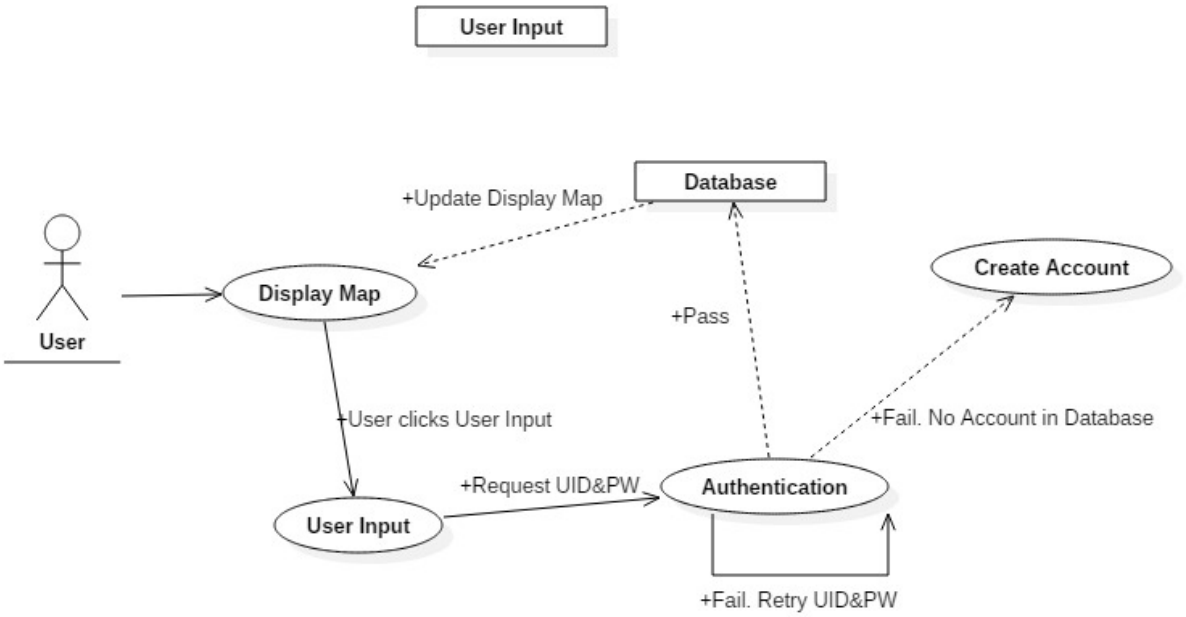


Use Case Template

Use Case ID:	3.0		
Use Case Name:	User Input		
Created By:	Group 8	Last Updated By:	Rollin
Date Created:	3/12/16	Date Last Updated:	3/12/16

Actors:	App User
Description:	User Input available for users with accounts to comment/update information about display map locations Branch of Display Map UC1.0
Trigger:	Trigger is the app user authenticating and submitting a request to provide input to data map
Preconditions:	1. User submits request to input/comment 2. Receipt of request and authentication check by database 3. User is authenticated
Postconditions:	1. Database is updated with user input 2. Display Map is updated from Database
Normal Flow:	UC3.0: User has access to map and clicks on User Input. User is checked for authentication. User inputs/comments on location on map. Input is then received by Database and Display Map is subsequently updated. User input is logged with user account.
Alternative Flows:	UC3.1: User is not authenticated, prompted to create an account if none exists UC3.2: User does not want to provide user input and may cancel request
Exceptions:	3.0.E.1: If one of the preconditions is not met then website lets the user know which condition(s) are preventing user from providing input(i.e. Loss of internet connection, user not authorized) 3.0.E.2: If preconditions are met but display map not correctly updated, user prompted to check if comment is logged. If logged, user is prompted to reload website (display map may not be updated by database yet)

	3.0.E.3: If an unaccounted error occurs then website/app gives a generic error message and asks user to reload the website. 3.1.E.1: User is not authenticated. User prompted again for username and password. Otherwise directed to create an account 3.2.E.1: If unaccounted error occurs while user tries to exit User Input, user asked to reload the website
Includes:	Authentication
Priority:	Medium Priority
Frequency of Use:	TBD by group 8
Business Rules:	None
Special Requirements:	The update to display map should not take too long. Update to display map from Database should remain constant
Assumptions:	None
Notes and Issues:	Frequency of user inputs. Length of time that a user input remains within database.

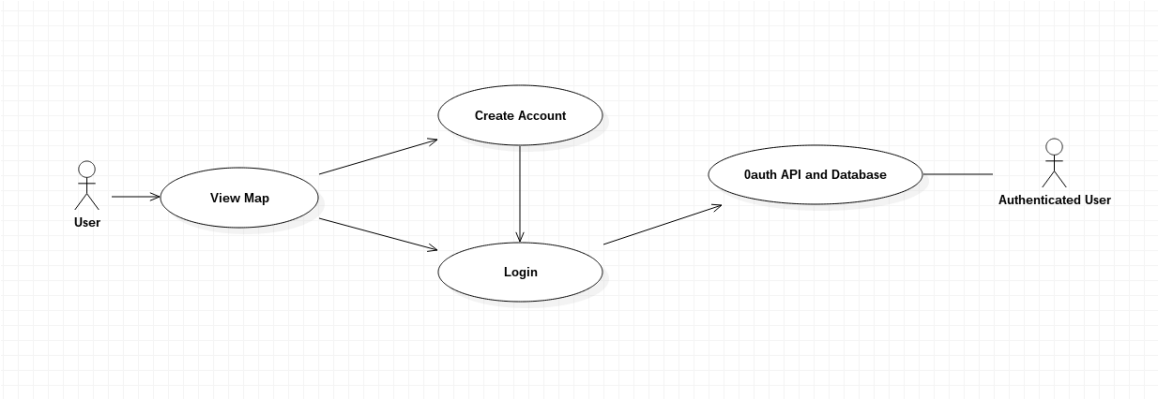


Use Case Template

Use Case ID:	4.0		
Use Case Name:	User Authentication		
Created By:	Group 8	Last Updated By:	Austin
Date Created:	3/12/16	Date Last Updated:	3/12/16

Actors:	App User
Description:	Authenticate user who tries to log in to the webapp (authenticated user is required for certain functionality).
Trigger:	User clicks on “login” or “signup” buttons.
Preconditions:	1. Username fits specified character requirements (validated with client-side logic). 2. Password fits specified character requirements (validated with client-side logic.
Postconditions:	1. If authentication succeeds, user is presented with a new view that allows access to authentication-required functions of the application. 2. If authentication fails, user is presented with a view that alerts them of the failure.
Normal Flow:	UC4.0: All site visitors can see the map/ basic functionality of the web app. Next, user clicks log in. After the user has authenticated, the user can then access more of the application's functionality.
Alternative Flows:	UC4.1: User has not yet created an account. User then clicks sign up, enters information, including an email address, and then confirms new account (from email). User then logs in (first flow).
Exceptions:	4.0.E.1: If the user's username or password do not meet the character requirements, the page notifies the user and has them input new ones. 4.0.E.2: If preconditions are met and there is an unexpected error then website/app rolls back to previous stable state (the original page with the map) and alerts the user that

	there was an authentication problem. 4.0.E.3: If an unaccounted error occurs then website/app gives a generic error message and asks user to reload the website. 4.0.E.4: If user inputs incorrect username and password then website/app tells the user their username/password is incorrect and to re-enter the information. 4.1.E.1: If user inputs insufficient information for account creation, app alerts user how to correct it.
Includes:	Account Creation/Deletion
Priority:	High Priority
Frequency of Use:	Each time a user logs in or signs up.
Business Rules:	None
Special Requirements:	The update to display map should not take too long. Update to display map from Database should remain constant
Assumptions:	None
Notes and Issues:	Webapp should allow some functionality without login.

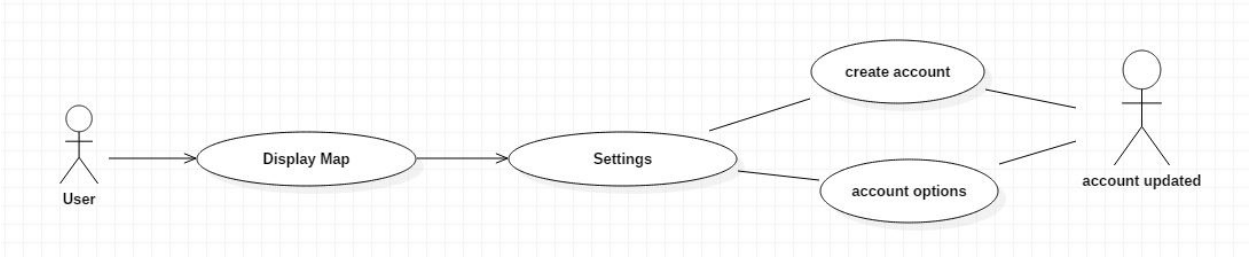


Use Case Template

Use Case ID:	5.0		
Use Case Name:	Settings		
Created By:	Group 8	Last Updated By:	Li Ke
Date Created:	3/12/16	Date Last Updated:	3/12/16

Actors:	App User
Description:	Setting allows users to manage their accounts
Trigger:	Users click on “Setting” button.
Preconditions:	1. The server is online. 2. The page is displayed completely 3. User can see the “Setting” button
Postconditions:	1. User can select “Account options” 2. User can select “Create Account”
Normal Flow:	UC5.0: User can create account, and will be able to use all functions after user authentication.
Alternative Flows:	UC5.1: User goes account options to change password. UC5.2: User goes account options to delete account UC5.3: User does not want to manage account, and may cancel request
Exceptions:	5.0.E.1: If the web app is down or in maintenance, the page will only display “in maintenance”. 5.1.E.1: if the new password is the same, the page notifies the user and has them input new ones.
Includes:	Create Account, Account options
Priority:	Medium Priority
Frequency of Use:	TBD by group 8
Business Rules:	None
Special Requirements:	Focus on simplicity/ user friendly design because “Settings” is the only way to manage account.
Assumptions:	None

Notes and Issues:	There can be more functions added in the future, for example, user's preference, notification...etc.

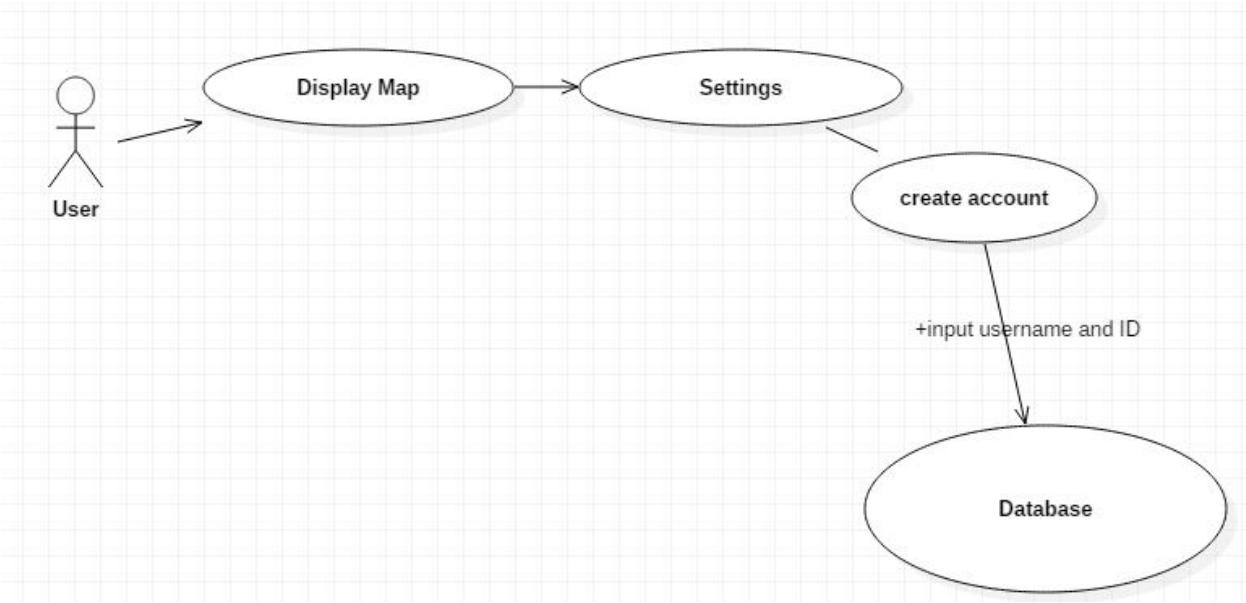


Use Case Template

Use Case ID:	6.0		
Use Case Name:	Create Account		
Created By:	Group 8	Last Updated By:	Li Ke
Date Created:	3/12/16	Date Last Updated:	3/12/16

Actors:	App User
Description:	Users can create accounts so they will be authenticated and can access more of the app.
Trigger:	User clicks on "Create account" button
Preconditions:	1. User has an email address to create an account. 2. The database is configured to store account information. 3. The 0auth API is used properly for user authentication.
Postconditions:	1. The user succeeded to make the account 2. If the user failed to make the account, the page will display "username or password does not fit the requirement".
Normal Flow:	UC6.0: User creates an account, fills in the information that fits the requirement. After clicking "I agree" for the terms and condition, a validate email will be sent to user's email address.
Alternative Flows:	UC6.1: User does not want to manage account, and may cancel request

Exceptions:	6.0.E.1: Username is taken 6.0.E.2: Password is too easy the page notifies the user and has them input new ones.
Includes:	Settings
Priority:	Medium Priority
Frequency of Use:	TBD by group 8
Business Rules:	None
Special Requirements:	Need special requirements for username and password, to protect user's account.
Assumptions:	None
Notes and Issues:	Each email is limited to making one account only, to prevent someone making numerous accounts



Use Case Template

Use Case ID:	7.0		
Use Case Name:	Delete Account		
Created By:	Group 8	Last Updated By:	Li Ke

Date Created:	3/12/16	Date Last Updated:	3/12/16

Actors:	App User
Description:	User can delete their account from the database, and will no longer be authenticated.
Trigger:	User clicks on “Delete account” button
Preconditions:	1. User has an account. 2. User has logged in to the account and authenticated.
Postconditions:	1. The user succeed to delete the account 2. The user now is not authenticated, and can no longer use full function anymore
Normal Flow:	UC7.0: User logs in to the account, clicks setting, account options, modify account, deleted account. Re-enter the password, and click “yes” to delete the account”
Alternative Flows:	UC7.1: User does not want to delete account, and may click “no” in the last step
Exceptions:	7.0.E.1: User re-enters the wrong password the page notifies the user and has them input new ones.
Includes:	Settings, user authentication
Priority:	Medium Priority
Frequency of Use:	TBD by group 8
Business Rules:	None
Special Requirements:	Send an email to the user, with a survey about the using experience.
Assumptions:	None
Notes and Issues:	All information of the user, will be removed from the database.

Revision History

Name	Date	Reason For Changes	Version
Rollin	2/21/16	Group Deliverable. Detail primary use case	1.0

Rollin	3/12/16	Added User Filter and User Input use cases + diagrams Changed “Filter Business/Events” to “Display Map” on template, new diagram, changed trigger, changed Flows	1.1
Austin	3/12/16	Added User Authentication use case	1.2
Li Ke	3/12/16	Added Settings, Account Creation/Deletion use cases	1.3