Cancer Stats App Product Requirements

Target release	v1.0
Epic	Week 4 Homework
Document status	1.0
Document owner	Max Fowler Tom Castonzo [Administrator]

Goals

- Establish requirements for the Cancer Stats App product
- Provide documentation for ACS560 class

Background and strategic fit

The requirements documentation compiled here will serve as our requirements for the ACS560 group project. It serves as a contract between us and the course, in that it provides exactly the steps we need to complete in order to have a finished project.

Assumptions

- We assume we have a working development server (this has been done, provided by Tom)
- We assume the server will support the back end language of our choice, whether it is C# or Golang. Tom has a server set up and it currently supports Golang (a planned back end language), Java, PHP, and Python (may also have some back end use)
- We assume the Android devices we have will be sufficient for application testing. If this is not true, we assume we can hopefully get one of the department's devices for testing. This should not be an issue, as Max's smart phone is an android smart phone.
- We assume some kind of mapping API can be used in conjunction with the code we write to provide the region support in the most organic wya possible. This is a fairly safe assumption, given the prevalence and usability of APIs like the Google Maps API.

Requirements - User Stories

#	Title	User Story	Importance	Criteria and Notes
1	User Registration	As a new user of the app, I want the ability to register a user identification and password with the application to make use of its services.	Must Have	• User identification may be username or password - this is a design step
2	User Login	As a user of the app, I want the ability to log into the application on my Android device to make use of the application.	Must Have	• The requirements assume an Android device. This is okay, as the app is targeted at Android explicitly
3	User Recovery	As a user of the app, I want the ability to reset a forgotten password using my email so that I can access the app even if I am forgetful.	Must Have	• This only works as long as the user remembers their email.
4	Region Selection	As a user of the app, I want the ability to select a region of the United States on a map to view the cancer statistics for so that I may view the stats.	Must Have	• The stats should be view-able with some sort of accompanying map. • The stats should be view-able in some form of chart or graph. The chart type will be determined during the design phase. • Regions should be defined by a reasonable name based on the location of the region. As a slightly specific example that may change during design, any area in Fort Wayne should be treated as 'Fort Wayne'.
5	Dynamic Region Data	As a traveling user of the app, I want the ability to get real-time data for the region I am currently passing through, that way I get a sense of the potential health risks of areas I am traveling through	Must Have	• The map should update its data as the user travels and enters a new region

6	Region Comparison	As a user of the app, I want the ability to select two regions of the United States on a map to view their cancer statistics so that I may compare the stats and see which region is healthier.	Must Have	• The stats should be view-able with some sort of accompanying map. • The stats should be view-able in some form of chart or graph. The chart type will be determined during the design phase. • The chart used must be able to compare the stats directly
7	Region Favorites	As a user of the app, I want the ability to select favorite regions so that I can view them later more easily in a list.	Optional, but would like to have	• The favorites should be stored in a list • The favorites should provide relevant region details to allow users to tell them apart • Duplicates should not be allowed
8	Region Marking	A a user of the app, I want the ability to mark regions on my favorites list as preferred. These regions should appear at the top of my list, so that I can access the more easily.	Optional, but required if we have a favorites list	• The mark should be reasonably visible on the list
9	Cancer Filtering - Region	As a user of the app, I want the ability to filter regions based upon incident rate percentages so that I can filter out regions of no interest to me based on their cancer prevalence.	Optional, based upon the way the map is implemented	• The goal of this region filtering is more research based and less practical.
10	Cancer Filtering - Favorites List	As a user of the app, I want the ability to filter my favorites list based on the percentage prevalence of cancer in the region's population to fine tune my interests on any given day.	Must have	• This is the more practical form of #8, as it allows the user to filter out regions they have previously seen based upon the cancer stat criteria
11	Data Set Updating	As an administrator of the app, I want the ability to update the data used by the application without directly accessing the database	Should have - for ease of use	• The back end code should support some interface, whether command line or GUI, to update the cancer stat data as needed
12	Data Set Exporting	As an administrator of the app, I want the ability to export the region stat information as raw data without directly accessing the database	Should have - for completeness	

Requirements - Functional, Server Side

This section addresses some functional requirements that did not fit nicely into user story format.

- The back end server application should handle all data processing between the database and the client.
- The client should only be concerned with asking for data, not performing calculations.
- The client needs to send query requests to the server. These requests will include the desired region and the cancer being looked at.
- The server is responsible for actually creating the queries from the client's requests.
- The server is responsible for receiving the data from the database and returning it to the client in a format the client can use.

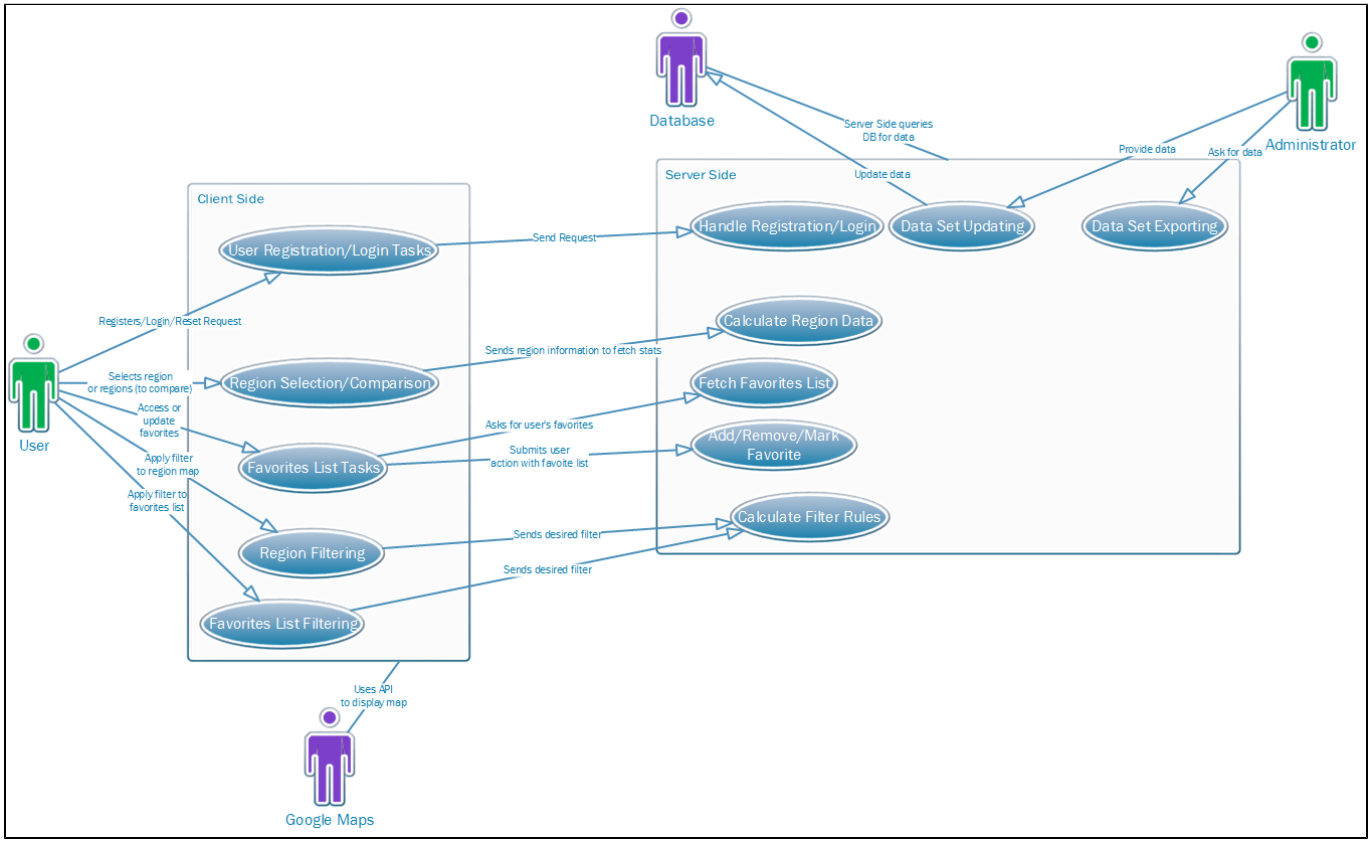
Requirements - Nonfunctional

This section addresses nonfunctional requirements, as they do not fit nicely into user story format.

- The application must use a client-server models
- The client and server must use different operating systems
    - The client uses the Android OS, with varying versions based on the device
        - The server uses a Debian Linux distribution
- The client and server require different programming languages
    - The client uses Java for Android.
    - The server-side may use Golang or C# and Python. The exact determination of the language used will follow in the design step, but Go is the most likely choice.
- The system must make use of a Database
    - The database used is MySql
- The system must respond to user requests within five seconds or report an error due to slow system response.

User interaction and design

Provided in this document is an informal use case diagram to provide a high level overview of the relations between actors and behaviors. Some of the actors used are not precisely actors in the most traditional sense. Traditional actors appear in green, while nontraditional actors appear in purple. For ease of construction and saving space, similar tasks were folded into the same use case.


Database
Server Side queries
DB for data
Update data
Administrator
Provide data
Ask for data
Server Side
Client Side
User Registration/Login Tasks
Send Request
Handle Registration/Login
Data Set Updating
Data Set Exporting
Registers/Login/Reset Request
Calculate Region Data
Sends region information to fetch stats
Selects region
or regions (to compare)
Region Selection/Comparison
Fetch Favorites List
Access or
update
favorites
Asks for user's favorites
Submits user
action with favoite list
Add/Remove/Mark
Favorite
User
Apply filter
to region map
Apply filter to
favorites list
Calculate Filter Rules
Region Filtering
Sends desired filter
Sends desired filter
Favorites List Filtering
Uses API
to display map
Google Maps