CMSC 447

Software Design Description (SDD)

[1 Scope 3](#_Toc432621786)

[1.1 Identification 3](#_Toc432621787)

[1.2 System overview 3](#_Toc432621788)

[1.3 Document overview 3](#_Toc432621789)

[2 Referenced documents 3](#_Toc432621790)

[3 CSCI-wide design decisions 3](#_Toc432621791)

[4 CSCI architectural design 4](#_Toc432621792)

[4.1 CSCI components 4](#_Toc432621793)

[4.2 Concept of execution 5](#_Toc432621794)

[4.3 Interface design 5](#_Toc432621795)

[4.3.1 Interface identification and diagrams 6](#_Toc432621796)

[4.3.2 (Project-unique identifier of interface) 6](#_Toc432621797)

[5 CSCI detailed design 7](#_Toc432621798)

[5.1 (Project-unique identifier of a software unit, or designator of a group of software units) 8](#_Toc432621799)

[6 Requirements traceability 9](#_Toc432621800)

[7 Notes 9](#_Toc432621801)

[A. Appendixes 10](#_Toc432621802)

# Scope

## Identification

This project is to be a web application which when implemented will allow a user to input preferences into a questionnaire and top locations will then be outputted onto a map. When first accessing the webpage, the user will begin with the “login page.” From the login page, a user can access a “create an account page” or after authentication access their “homepage.” The homepage will allow the user to modify their account information from a “modify account page,” as well as allow the user to see the “map” and access the “questionnaire,” which are defined below.

The “questionnaire” refers to the portion of the project where the user inputs these preferences. This will be accessed from the homepage into a “questionnaire page.” The “ratings” will refer to the numerical input from a user in which they determine which statistics have a higher preference. The “statistics” refer to the set of data that our project will be accessing.

The “map” refers to the portion of the project which is the visualization of data. The map is initially displayed from the homepage, but will initially be empty, and be modified as the user takes the questionnaire. After the user completes the questionnaire the map is referenced to be an “updated map” with the continuing definitions being components of the map. The map may also have markers which will be referred to as “pins.” The pins will allow the user to see an image of the location referred to as the “pin image.” In some cases where an image is not available, or if there is additional implementation, a pin may allow the user to see descriptive information on the location which will be referred to as the “pin information.”

## System overview

The goal of this project is to create a web application which will allow a user to create an account, and from such account be able to evaluate different statistical values such that a list of locations within the United States will be outputted to the user which correlate to the user input. With the list, a map should be presented with pins that correlate to the location and can then be further expanded to output a location image or description.

We expect to use a database for maintaining username and password data. Along with this we expect to be using third party software for our output. This would include the use of GoogleMaps for our map output, and possibly Flickr for our image outputs, and Wikipedia for a description output. Further third-party software will be expanded in this section as seems fit throughout the project.

Because the nature of this project is in the scope of a classroom project, there will not be long term maintenance of the project and will be run on a local machine. Throughout this document the term sponsor will refer to John Winder who is the group’s client throughout the project. Similarly, the group refers to the group of developers working on the project including Matthew Hearn, Aaron Lewis, Alex Rochford, Cathy Poore, Ben Kittner, and Steven Heckman. The project will refer to the software and documentation created for this assignment.

All project development will be done through GitHub and then developer preference for development environments, debuggers, ect. The web portion of the application will be ran using an Angular Framework on a localhost.

## Document overview

The goal of this document is to serve as a description of the system design as well as how the system will be split into different components. This document will all list any frameworks, APIs, or any other component which will need to be integrated into the system. For the sake of traceability and modifications to this document will be outlined below. Documentation version are below for client traceability,

Version 1.0 Original Document

This is the original document. Further changes will be outlined below.

Version 2.0 Update for 4/23 Meeting

There was a requirement update from our group meeting on 4/23. This included that the requirement of having a profile avatar was necessary. Design regarding this was laid out below. This will mostly be done through an added field in the login database which will hold an integer corresponding to the users desired avatar. Choices for avatar will be given by the user upon account creation. The user can modify the avatar from the modify account page. The avatar will be displayed next to the user’s ‘about me’ text on the homepage of the application.

Version 2.1 Updated some minor grammar changes. Update the requirements traceability to be more visually appealing and readable.

# Referenced documents

SDP - Software Development Plan

SRS - Software Requirement Specification

# CSCI-wide design decisions

This section shall be divided into paragraphs as needed to present CSCI-wide design decisions, that is, decisions about the CSCI’s behavioral design (how it will behave, from a user’s point of view, in meeting its requirements, ignoring internal implementation) and other decisions affecting the selection and design of the software units that make up the CSCI. If all such decisions are explicit in the CSCI requirements or are deferred to the design of the CSCI’s software units, this section shall so state. Design decisions that respond to requirements designated critical, such as those for safety, security, or privacy, shall be placed in separate subparagraphs. If a design decision depends upon system states or modes, this dependency shall be indicated. Design conventions needed to understand the design shall be presented or referenced. Examples of CSCI-wide design decisions are the following:

1. Design decisions regarding inputs the CSCI will accept and outputs it will produce, including interfaces with other systems, HWCIs, CSCIs, and users [(4.3.x](#_bookmark3) of this DID identifies topics to be considered in this description). If part or all of this information is given in Interface Design Descriptions (IDDs), they may be referenced.
2. Design decisions on CSCI behavior in response to each input or condition, including actions the CSCI will perform, response times and other performance characteristics, description of physical systems modeled, selected equations/algorithms/rules, and handling of unallowed inputs or conditions.
3. Design decisions on how databases/data files will appear to the user [(4.3.x](#_bookmark3) of this DID identifies topics to be considered in this description). If part or all of this information is given in Database Design Descriptions (DBDDs), they may be referenced.
4. Selected approach to meeting safety, security, and privacy requirements.
5. Other CSCI-wide design decisions made in response to requirements, such as selected approach to providing required flexibility, availability, and maintainability.

Currently our CSCI wide design decisions would be those relating to the setup of our databases and our algorithm implementation. To meet the requirement of verifying username and passwords, and store statistical data, the development team will be using databases. The se will be split between a login database and a data database. The login database will need to be accessible from all CSCI components, so user information can be queried. This will be most commonly associated with creating an account, modifying their account, storing their top locations, and displaying their top locations.

To meet the requirements regarding statistical variety in our results, the development team needed to create a method for scoring different locations. This was done through normalizing our data into percentages. This was normally done by dividing the given occurrences of a data item by the population of the location. This number could then be multiplied by the user scoring of a data item. This data score summed with the scores from the other locations would give the user an individual locations score. We will maintain a list of the top 10 choices while calculating the score. The final list will then be sent to the user login database which will maintain the list of user locations.

# CSCI architectural design

This section shall be divided into the following paragraphs to describe the CSCI architectural design. If part or all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced.

## CSCI components

This paragraph shall:

1. Identify the software units that make up the CSCI. Each software unit shall be assigned a project-unique identifier.

Note: A software unit is an element in the design of a CSCI; for example, a major subdivision of a CSCI, a component of that subdivision, a class, object, module, function, routine, or database. Software units may occur at different levels of a hierarchy and may consist of other software units. Software units in the design may or may not have a one- to-one relationship with the code and data entities (routines, procedures, databases, data files, etc.) that implement them or with the computer files containing those entities. A database may be treated as a CSCI or as a software unit. The SDD may refer to software units by any name(s) consistent with the design methodology being used.

The overall system we are creating is a web application which will include a login, account modification, a questionnaire, and map output. Each of these are the components of the system, which upon completion will be a different version of the design. All these also work with one another, however can be tested on an individual basis.

CSCI component 1 will be the login portion of the application. This will include the original homepage of the website which will allow a user to create an account or login. Upon successful login the user would access the homepage for the application. For the login, the application will need to interface with the login database which will contain a user index, username, password, and “About Me” textfield. Creating account should increment the index, as well as verify that the username portion is unique, and neither the username or password are NULL. The user should also be able to optionally choose an avatar for their profile. For reference, the login portion of the application will be referenced as CSCI comp1.

CSCI component 2 will be the account modification portion of the application. This will be available from the homepage and allow the user to change their username, password, or “About Me” textfield. Modifying the account will also need to interface with the login database. Modifying the account should verify that the new username is unique, neither the username or password are NULL, as well as the username and password are different from the previously used username and password. The user should also be able to modify their profile avatar from the modify account page. For reference, the account modification portion of the application will be referenced as CSCI comp2.

CSCI component 3 will be the questionnaire portion of the application. This will need to present the user different statistics and allow the user to give a ranking to each statistic based on personal preference. The component will then need to interface with a dataset which holds statistical data to output locations which best correspond to the user preferences. The data we will be accessing will be from the Census Bureau which contains a data set which we will import as a database. This will contain allow us to associate a county with its state ID number, which will allow us to determine the counties state. The actual data for any county provided by the data set includes the number of violent crimes, percentage of population over 65, unemployment rate, population, housing cost, number of art institutions, percent of population with a bachelor’s degree, percent growth rate, median income, poverty rate, land area, and population density. Next, the Census Bureau also provides an environmental ranking to each state, which we will just allow each county within a state to have the same environmental ranking. This will allow us to rank counties based on an environmental tanking, art institutions per a unit area, and a crime rate of the number of crimes divided by the population. The final list of locations will be stored in the login database to save the user locations.These will satisfy our three-empirical statistics requirement. For reference, the questionnaire portion of the application will be refenced as CSCI comp3.

CSCI component 4 will be a map output which will visualize the data collected in the questionnaire. A map of the United States will be provided on the homepage of the application. However, the map will serve little purpose prior to the completion of the questionnaire in CSCI comp3. After the user completed the questionnaire we will have a list of the counties which are best associated with their preferences in locations. This will use a GoogleMaps API to printout a map of the United States, and then place pins on the top locations for the user associated with their individual list of locations. These pins will then interact with the GooglePlaces API which when inputted with the county ID, from the Census Bureau dataset, will be able to output an image and description of the county. For reference, the map output portion of the application will be refenced as CSCI comp4.

1. Show the static (such as "consists of") relationship(s) of the software units. Multiple relationships may be presented, depending on the selected software design methodology (for example, in an object-oriented design, this paragraph may present the class and object structures as well as the module and process architectures of the CSCI).

CSCI comp1 will consist of the login webpage of the application. This will direct the user to a create an account page or allow the user to login if they already have an account. The create account page will consist of textfields for the user to create an account username, password, and “About Me” text. After creating an account, the user will be redirected to the login webpage. After the user inputs their account username and password correctly, they will be redirected to the application homepage. The application homepage will consist of a map of the United States and have links to CSCI comp2 and CSCI comp3.

CSCI comp2, the modify account page will be accessed form the application homepage. This will consist of textfields for the user to modify their username, password, or “About Me” text. This will also interact with the user database to verify that the username and password have indeed been changed, as well as it being necessary to modify the account information for the next login. After modifying the account, the user will receive as confirmation message, and then be redirected to the homepage. Finally, CSCI comp2 will consist of a return to homepage button which will allow the user to return to the homepage if they do not modify their account information.

CSCI comp3 will consist of the questionnaire that the user will take to rank statistics on personal preference. The questionnaire webpage will consist of the individuals’ statistics as well as fields for a user to give a rating 0-10 for the statistic. This will then interact with our Census Bureau data database which will allow us to give an output of the users top ten locations. The user will then be redirected to the homepage.

CSCI comp4 will consist of the map located within the homepage. This will need to interact with both the homepage for its original popup, as well as the list output from CSCI comp3. From the list, the map will consist with pins corresponding to the user’s top locations. The user will then be able to click a pin, which will consist of both a description of the location, as well as an image provided by GooglePlaces API.

1. State the purpose of each software unit and identify the CSCI requirements and CSCI- wide design decisions allocated to it. (Alternatively, the allocation of requirements may be provided in [6.a.)](#_bookmark5)

This will be better summarized in the table found in 6.a. The following requirement numbers will be referencing the numbers assigned to requirements within the SRS. CSCI comp1 will satisfy requirements 3.1.a.1-3.1.a.4 ,3.1.b.1-5, 3.1.c.1-3,5,3.2.a.1-2 ,3.2.b.1-5,3.2.c.1-3,5 and 3.5.a-c. CSCI comp2 will satisfy requirements 3.1.d.1-3, and 3.2.d.1-4. CSCI comp3 will satisfy requirements 3.1.e.1-2, 3.1.e.3.a-c, and 3.2.e.1-4. CSCI comp3 will satisfy requirements 3.1.c.4, 3.1.e.a-c, and 3.2.e.5. Upon completion of these requirements, then the system will satisfy the following requirements. These requirements include 3.1.a, 3.1.b,3.1.c, 3.1.d, 3.1.e, 3.1.e.3, 3.2, 3.2.b, 3.2.c, 3.2.d, 3.2.e, and 3.5. This will satisfy the overall requirement listen in 3.1

CSCI comp1 will serve the purpose of being the initial webpage a user will interact with. This will allow them to create an account and proceed to the homepage. From the homepage the user can then proceed to interact with the other CSCI components. CSCI comp2 serves the purpose of allowing the user to change their username or password. This is done primarily for them to be able to change their “About Me” section but will also allow the user to modify their username or password. CSCI comp3 serves as the overall backend portion of the application. The user will be able to input their personal preferences, and these will be used to interact with CSCI comp4. CSCI comp3 serves as the data access and data manipulation portion of this project. Finally, CSCI comp4 serves as the data visualization portion of the project. This will output the list of locations from CSCI comp3 onto a GoogleMaps of the United States. It will also allow the user to gather more information on a given location through a pin of each location which will give the GooglePlaces information of that location including a description and image.

1. Identify each software unit’s development status/type (such as new development, existing design or software to be reused as is, existing design or software to be reengineered, software to be developed for reuse, software planned for Build N, etc.) For existing design or software, the description shall provide identifying information, such as name, version, documentation references, library, etc.

The website portion of the application will be built using the Angular framework. This will allow for the creation of the different webpages, as well as allow for movement within the different pages in the application. CSS will then be used for the editing of the visuals within the pages. The databases used within the project will be developed by us as the developers. The data database will be initialized from an excel document provided by the Census Bureau. As a result, the questionnaire will also be custom build. Finally, the map will be using the GoogleMaps API which can display a map of the United States within the page. The GoogleMaps API will also allow us to place the pins on the map when we input a location. Finally, we will use the GooglePlaces API for presenting the location image and description.

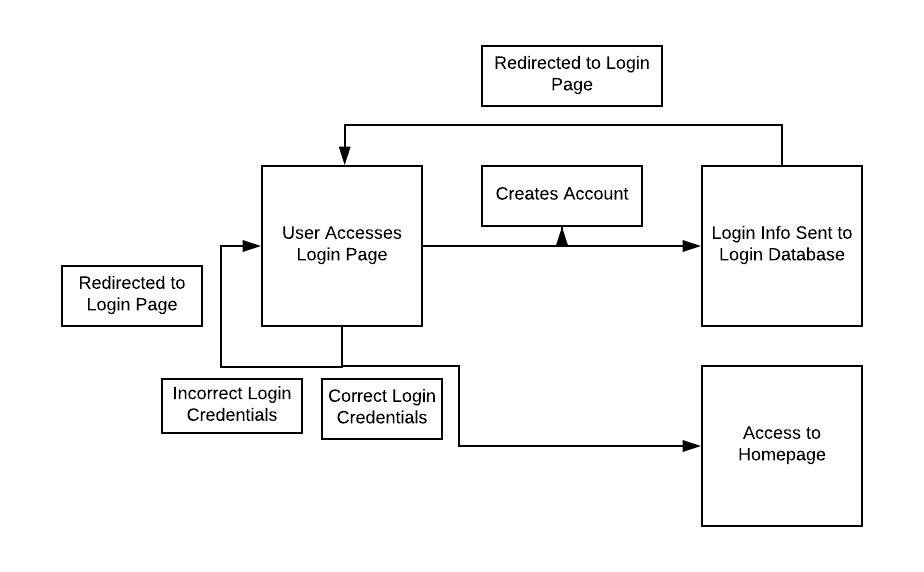
1. Describe the CSCI’s (and as applicable, each software unit’s) planned utilization of computer hardware resources (such as processor capacity, memory capacity, input/output device capacity, auxiliary storage capacity, and communications/network equipment capacity). The description shall cover all computer hardware resources included in resource utilization requirements for the CSCI, in system-level resource allocations affecting the CSCI, and in resource utilization measurement planning in the Software Development Plan (SDP). If all utilization data for a given computer hardware resource are presented in a single location, such as in one SDD, this paragraph may reference that source. Included for each computer hardware resource shall be:
   1. The CSCI requirements or system-level resource allocations being satisfied
   2. The assumptions and conditions on which the utilization data are based (for example, typical usage, worst-case usage, assumption of certain events)
   3. Any special considerations affecting the utilization (such as use of virtual memory, overlays, or multiprocessors or the impacts of operating system overhead, library software, or other implementation overhead)
   4. The units of measure used (such as percentage of processor capacity, cycles per second, bytes of memory, kilobytes per second)
   5. The level(s) at which the estimates or measures will be made (such as software unit, CSCI, or executable program)
2. Identify the program library in which the software that implements each software unit is to be placed.

There are no hardware components of this application and as a result this section will be left blank until the need for modification if such is made.

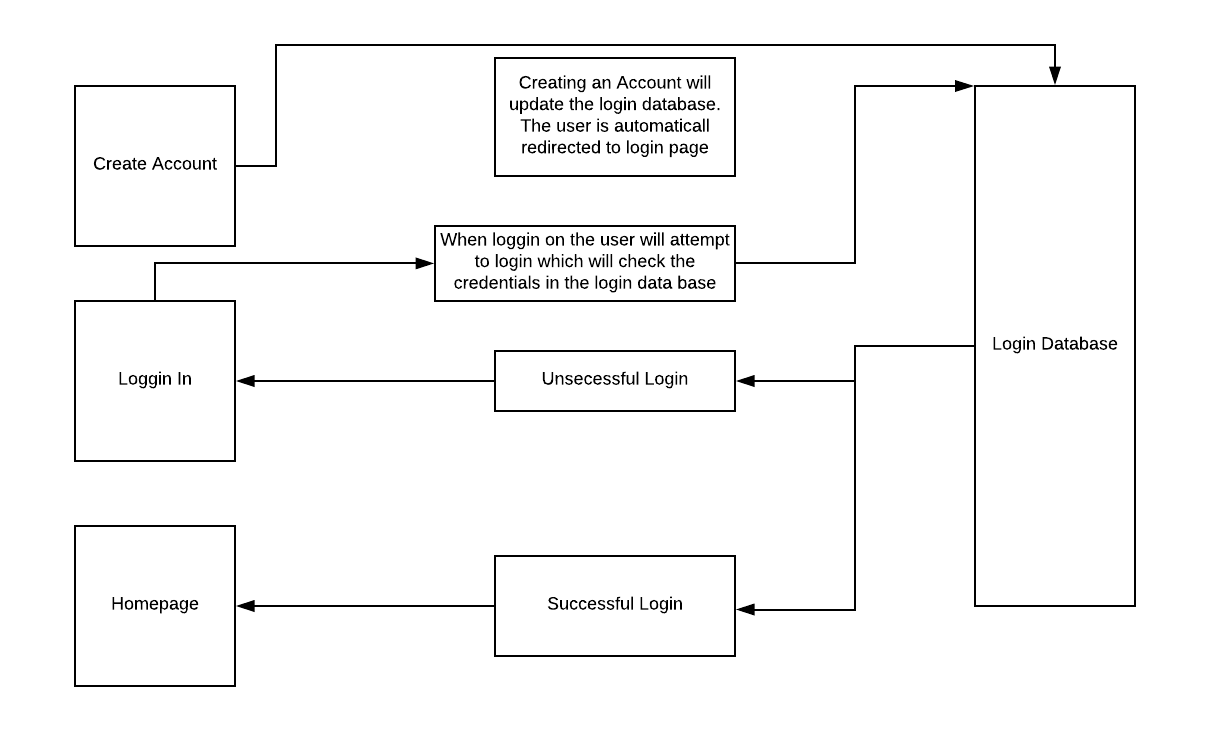
## Concept of execution

This paragraph shall describe the concept of execution among the software units. It shall include diagrams and descriptions showing the dynamic relationship of the software units, that is, how they will interact during CSCI operation, including, as applicable, flow of execution control, data flow, dynamically controlled sequencing, state transition diagrams, timing diagrams, priorities among units, handling of interrupts, timing/sequencing relationships, exception handling, concurrent execution, dynamic allocation/deallocation, dynamic creation/deletion of objects, processes, tasks, and other aspects of dynamic behavior.

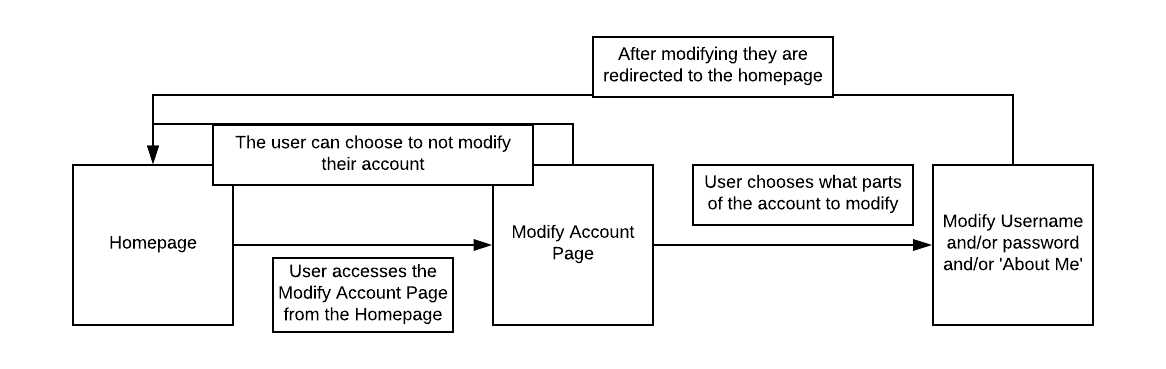
The following diagram demonstrates how the user will progress the login process. The user will start at the login page.



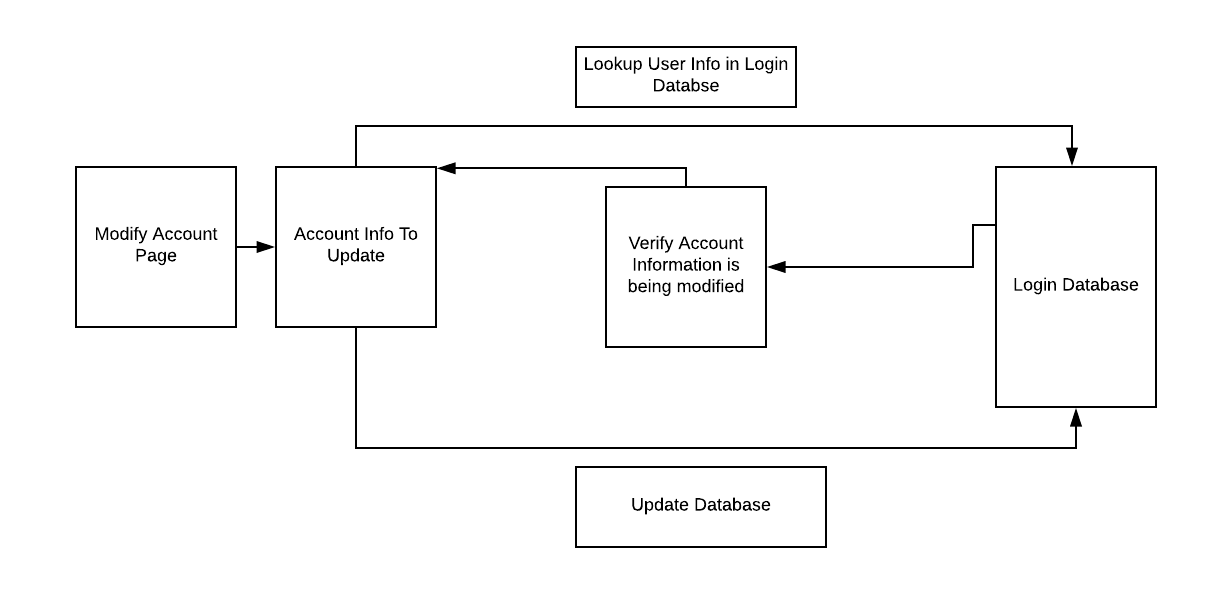
The following diagram demonstrates the flow of data within the login process.



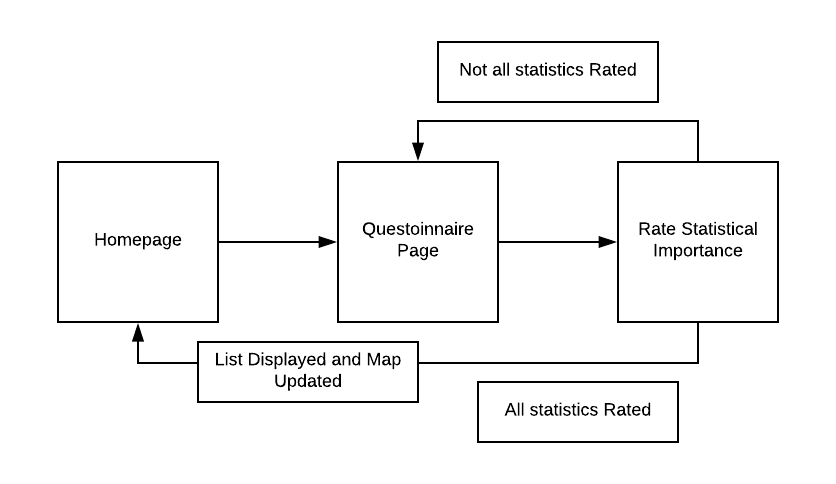
The following diagram demonstrates the flow for the user during the modify account process.



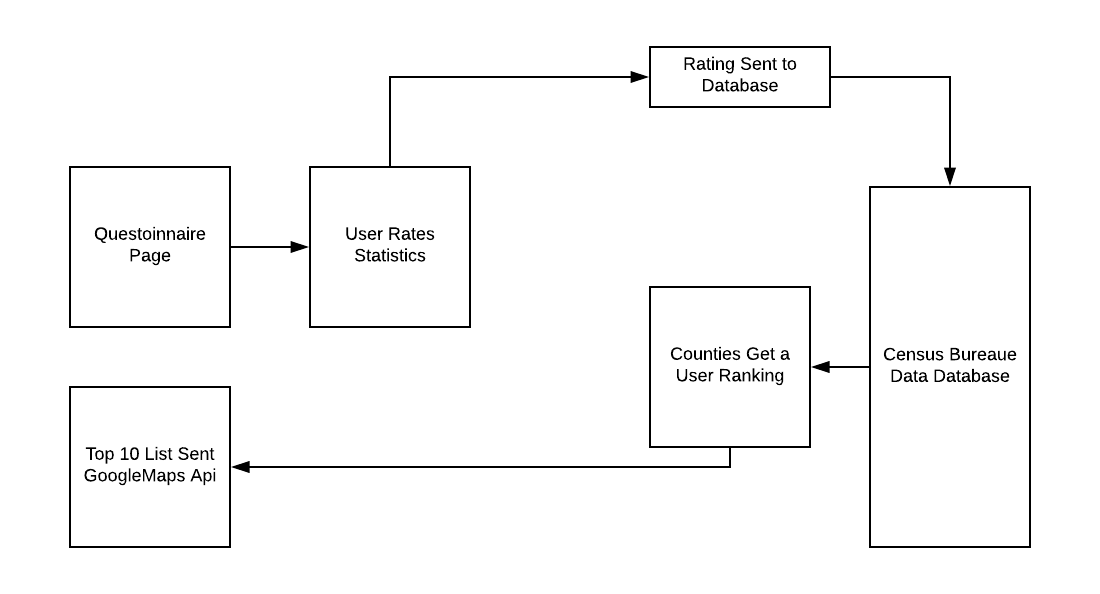
The following diagram demonstrates the flow of information during the modify account page.



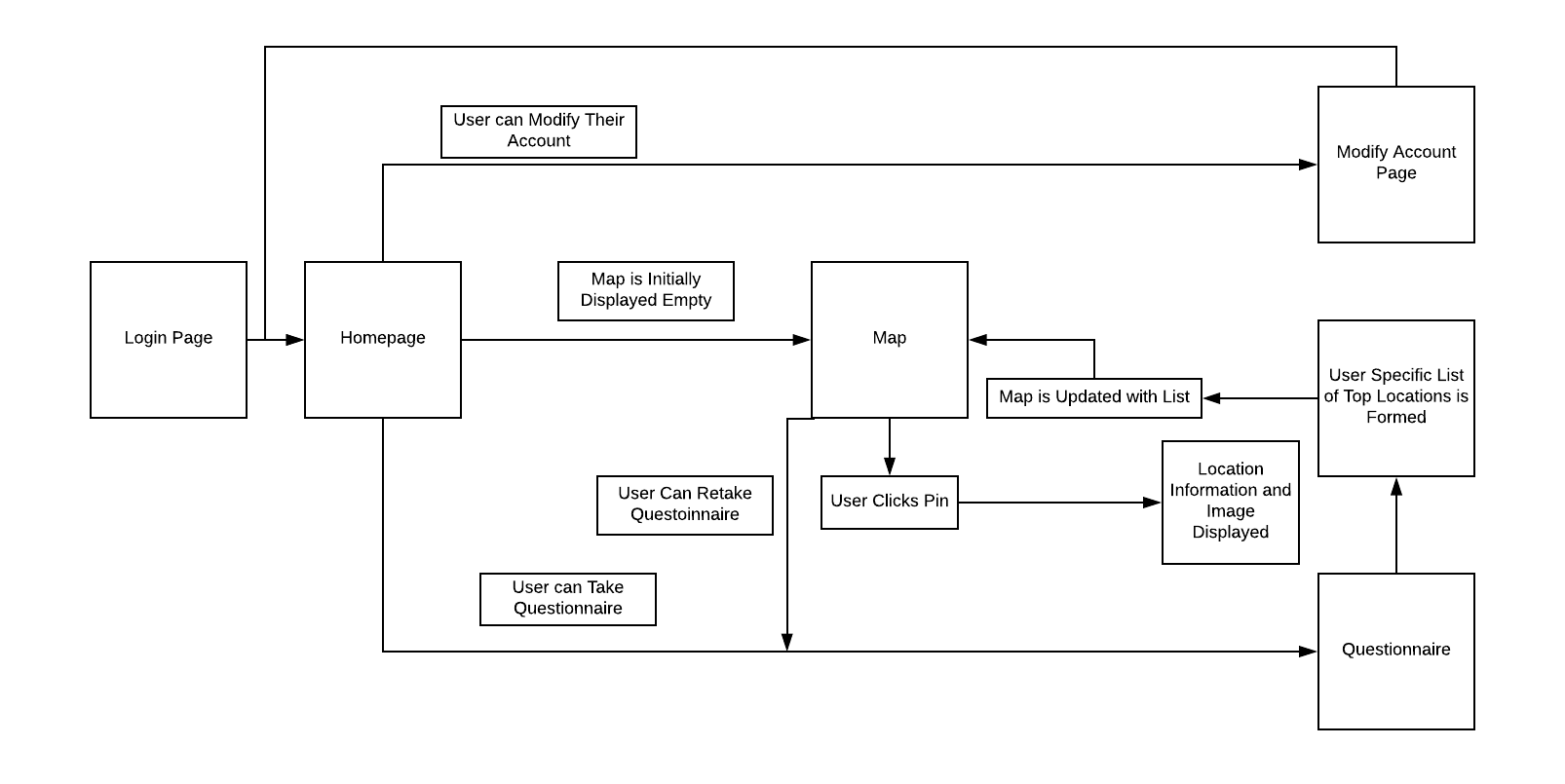
The following diagram demonstrates the user flow during the questionnaire process.



The following diagram demonstrates the flow of data within the questionnaire process.



The following diagram demonstrates the user flow form the homepage.



## Interface design

This paragraph shall be divided into the following subparagraphs to describe the interface characteristics of the software units. It shall include both interfaces among the software units and their interfaces with external entities such as systems, configuration items, and users. If part or all of this information is contained in Interface Design Descriptions (IDDs), in section [5](#_bookmark4) of the SDD, or elsewhere, these sources may be referenced.

### Interface identification and diagrams

This paragraph shall state the project-unique identifier assigned to each interface and shall identify the interfacing entities (software units, systems, configuration items, users, etc.) by name, number, version, and documentation references, as applicable. The identification shall state which entities have fixed interface characteristics (and therefore impose interface requirements on interfacing entities) and which are being developed or modified (thus having interface requirements imposed on them). One or more interface diagrams shall be provided, as appropriate, to depict the interfaces.

The first user interface will be found within the login page. In this webpage the user will input their username and password or access the create account page. This will be known as the login interface. If the user chooses to create an account, they will access the create account page. From here the user will have three textfields to enter a desired username, password, and “About Me” text. This interface will be known as the create account interface. The next interface will be on the user homepage which will allow them to move to the modify account page and the questionnaire page. This will be known as the homepage interface. After the questionnaire has been taken the homepage interface will also serve as the interface for the user to receive information on the locations from the pins placed on the map. The next user interface will be the modify account page which will have user input for the desired username, password, and “About Me” text. This will be known as the modify account interface. The final user interface will be found on the questionnaire page. From here the user will rate statistics from zero to ten. The user will then be able to submit once all statistics are given a rating. This will be known as the questionnaire interface.

### (Project-unique identifier of interface)

This paragraph (beginning with 4.3.2) shall identify an interface by project-unique identifier, shall briefly identify the interfacing entities, and shall be divided into subparagraphs as needed to describe the interface characteristics of one or both of the interfacing entities. If a given interfacing entity is not covered by this SDD (for example, an external system) but its interface characteristics need to be mentioned to describe interfacing entities that are, these characteristics shall be stated as assumptions or as "When [the entity not covered] does this, [the entity that is covered] will ...." This paragraph may reference other documents (such as data dictionaries, standards for protocols, and standards for user interfaces) in place of stating the information here. The design description shall include the following, as applicable, presented in any order suited to the information to be provided, and shall note any differences in these characteristics from the point of view of the interfacing entities (such as different expectations about the size, frequency, or other characteristics of data elements):

* + - 1. Priority assigned to the interface by the interfacing entity(ies)
      2. Type of interface (such as real-time data transfer, storage-and-retrieval of data, etc.) to be implemented
      3. Characteristics of individual data elements that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:
         1. Names/identifiers

Project-unique identifier

Non-technical (natural-language) name

DoD standard data element name

Technical name (e.g., variable or field name in code or database)

Abbreviation or synonymous names

* + - * 1. Data type (alphanumeric, integer, etc.)
        2. Size and format (such as length and punctuation of a character string)
        3. Units of measurement (such as meters, dollars, nanoseconds)
        4. Range or enumeration of possible values (such as 0-99)
        5. Accuracy (how correct) and precision (number of significant digits)
        6. Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated and whether business rules apply
        7. Security and privacy constraints
        8. Sources (setting/sending entities) and recipients (using/receiving entities)
      1. Characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:
         1. Names/identifiers

Project-unique identifier

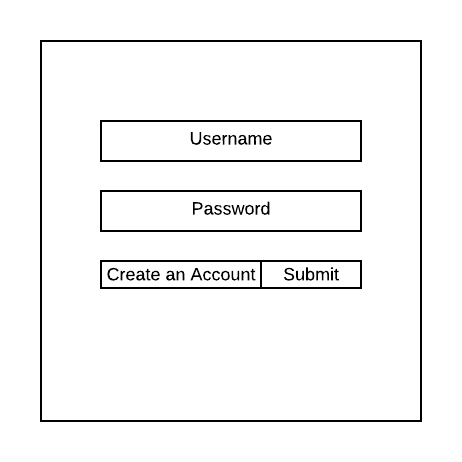
Non-technical (natural language) name

Technical name (e.g., record or data structure name in code or database)

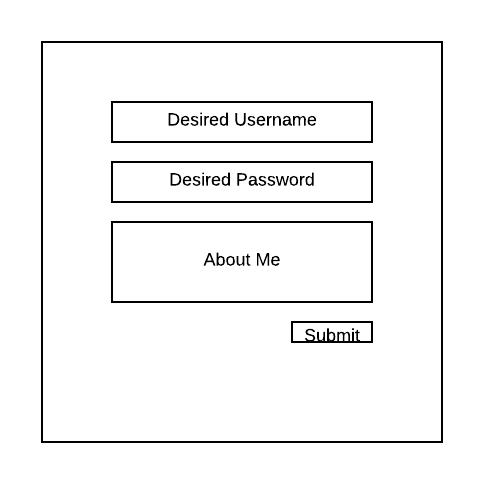
Abbreviations or synonymous names

* + - * 1. Data elements in the assembly and their structure (number, order, grouping)
        2. Medium (such as disk) and structure of data elements/assemblies on the medium
        3. Visual and auditory characteristics of displays and other outputs (such as colors, layouts, fonts, icons and other display elements, beeps, lights)
        4. Relationships among assemblies, such as sorting/access characteristics
        5. Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated and whether business rules apply
        6. Security and privacy constraints
        7. Sources (setting/sending entities) and recipients (using/receiving entities)
      1. Characteristics of communication methods that the interfacing entity(ies) will use for the interface, such as:
         1. Project-unique identifier(s)
         2. Communication links/bands/frequencies/media and their characteristics
         3. Message formatting
         4. Flow control (such as sequence numbering and buffer allocation)
         5. Data transfer rate, whether periodic/aperiodic, and interval between transfers
         6. Routing, addressing, and naming conventions
         7. Transmission services, including priority and grade
         8. Safety/security/privacy considerations, such as encryption, user authentication, compartmentalization, and auditing
      2. Characteristics of protocols the interfacing entity(ies) will use for the interface, such as:
         1. Project-unique identifier(s)
         2. Priority/layer of the protocol
         3. Packeting, including fragmentation and reassembly, routing, and addressing
         4. Legality checks, error control, and recovery procedures
         5. Synchronization, including connection establishment, maintenance, termination
         6. Status, identification, and any other reporting features
      3. Other characteristics, such as physical compatibility of the interfacing entity(ies) (dimensions, tolerances, loads, voltages, plug compatibility, etc.)

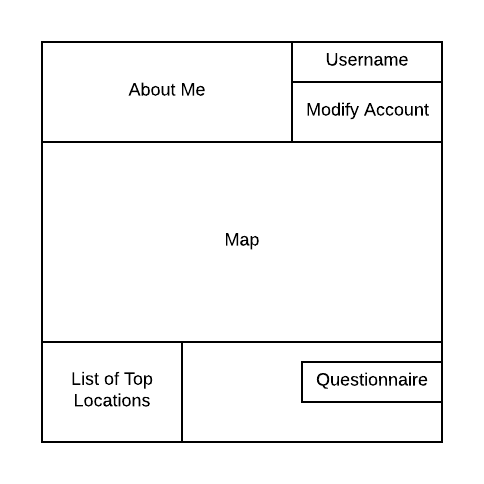
The login interface will serve as the users first exposure to the application. From here the user will be able to enter their username and password if they already have an account or be directed to create an account. This interface serves to retrieve data stored in the user login database, as well as authenticate the user. The username and password will need to be an exact match with the information stored within the login database. All data entered here will be treated as strings. To access the create an account page the user will need to hit a button labeled “create an account.” Below is the layout for the login interface.



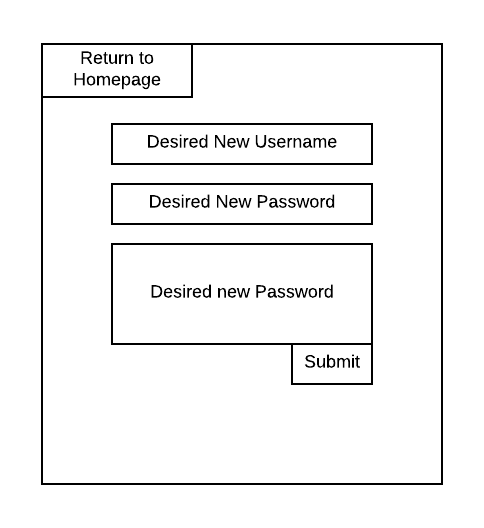
The create account interface will contain three textfields for the user to enter their desired username, password, and a “About Me” text. These values will then be sent to the login database for account validation when the user logs in. The username, password, and “About Me” fields will be treated as strings. These fields will then be stored in the login database. Below is a layout for the create account interface.



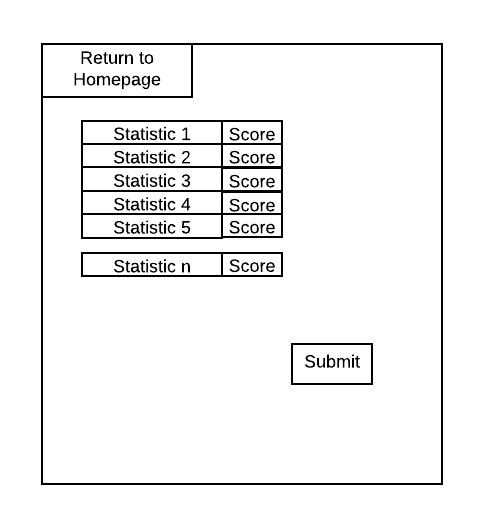
The homepage interface will serve as the main page for the user to go to the other aspects of the application. It will also serve as the location for the final data visualization. Within the page the username should be displayed in the top right of the screen. Below, the user will have the ability to access the modify account page through a button. To the side of the account information will be the users “About Me” text. Next, the main portion of the page should be the visual of the map. Once the map has been updated from the questionnaire there will also be pins for the location which the user can click as buttons to be displayed with an image and description of the location. Also, after the questionnaire there will be a list of the user’s top locations found below the map. Finally, there should be a button allowing the user to move to the questionnaire page. Below is a layout for the homepage interface.



From the homepage the user can then access the modify account interface. From here the user can change their username, password, or “About Me” text. All these variables will be stored as strings. After submission these values will be validated and then update the login database. The username and passwords will be validated with current login database information to validate that the username or password has been changed, and that the new username and password are not NULL. Finally, the modify account interface should allow the user to return to the homepage through a button. Below is a layout of the modify account interface.



The final interface is the questionnaire interface. This is accessed through the questionnaire button found on the homepage. This will allow the user to rank multiple statistics which will then be used in the calculations of the users top 10 locations. The numerical values for the ranking of statistics will be kept as integers. The values stored within the actual dataset will be stored as floats with the following limitations. All names within the data will be stored as strings. All county identifiers will be stored as integers. The output list of top user locations will be an array of strings. Finally, there will be a button for the user to submit their rankings which will redirect them to the homepage with an updated map. The user will be able to optionally return to the homepage without completing the questionnaire through a return to homepage button. Below is a layout of the questionnaire interface.



# CSCI detailed design

This section shall be divided into the following paragraphs to describe each software unit of the CSCI. If part of all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced. Interface characteristics of software units may be described here, in Section [4,](#_bookmark0) or in Interface Design Descriptions (IDDs). Software units that are databases, or that are used to access or manipulate databases, may be described here or in Database Design Descriptions (DBDDs).

## (Project-unique identifier of a software unit, or designator of a group of software units)

This paragraph shall identify a software unit by project-unique identifier and shall describe the unit. The description shall include the following information, as applicable. Alternatively, this paragraph may designate a group of software units and identify and describe the software units in subparagraphs. Software units that contain other software units may reference the descriptions of those units rather than repeating information.

* + 1. Unit design decisions, if any, such as algorithms to be used, if not previously selected
    2. Any constraints, limitations, or unusual features in the design of the software unit
    3. The programming language to be used and rationale for its use if other than the specified CSCI language
    4. If the software unit consists of or contains procedural commands (such as menu selections in a database management system (DBMS) for defining forms and reports, on- line DBMS queries for database access and manipulation, input to a graphical user interface (GUI) builder for automated code generation, commands to the operating system, or shell scripts), a list of the procedural commands and reference to user manuals or other documents that explain them
    5. If the software unit contains, receives, or outputs data, a description of its inputs, outputs, and other data elements and data element assemblies, as applicable. Paragraph [4.3.x](#_bookmark3) of this DID provides a list of topics to be covered, as applicable. Data local to the software unit shall be described separately from data input to or output from the software unit. If the software unit is a database, a corresponding Database Design Description (DBDD) shall be referenced; interface characteristics may be provided here or by referencing section [4](#_bookmark0) or the corresponding Interface Design Descriptions (IDDs).
    6. If the software unit contains logic, the logic to be used by the software unit, including, as applicable:
       1. Conditions in effect within the software unit when its execution is initiated
       2. Conditions under which control is passed to other software units
       3. Response and response time to each input, including data conversion, renaming, and data transfer operations
       4. Sequence of operations and dynamically controlled sequencing during the software unit’s operation, including:
          1. The method for sequence control
          2. The logic and input conditions of that method, such as timing variations, priority assignments
          3. Data transfer in and out of memory
          4. The sensing of discrete input signals, and timing relationships between interrupt operations within the software unit
       5. Exception and error handling

The first CSCI comp1 consist of the login portion of the application. This includes the login webpage, the create account webpage, the login interface, and the create account interface. The login information will interact with the login database. A typical user will begin in the user login page and login interface. A new user will need to create an account by clicking the ‘create an account’ button. From here they will have textfields for enter a username, password, and ‘About Me’ text. Upon successful login the user will be directed to the homepage. Finally, the user will be able to choose a preferred avatar when they create their account. This will be displayed with their about me sections.

Most of this portion of the application will be written using the Angular Framework and CSS for the webpage design. For the manipulation of variables within the webpage Typescript, HTML, and JavaScript will be used. For the database, Angular provides an interaction with data that is stored with in a JSON format through a PHP server running MYSQL. This will be how we will store and extra information from our login database.

The application will verify that the username is unique within the login database. This is to prevent multiple users with the same username. If a username is not unique, then the error will be handled by redirecting the user to the create account page after a message is displayed stating that the username is not unique. Next, there will be verification that neither the username or password are NULL. If either of these fields are NULL then the user will be redirected to the create an account page after a message is displayed stating that the username and password may not be empty.

CSCI comp2 consist of the modify account portion of the application. This will mirror CSCI comp1 since it also interacts with the login database. This portion consist of the modify account webpage and modify account interface. The user will be able to move back to the homepage from the modify account page or submit the account changes. When accessing the page, the user will be able to modify their username, password, or “About Me” text. The user will then be able to submit these changes.

This portion of the application will be written using the Angular Framework and CSS for the webpage design. For the manipulation of variables within the webpage Typescript, HTML, and JavaScript will be used. For the database, Angular provides an interaction with data that is stored with in a JSON format through a PHP server running MYSQL. This will be how we will store and extra information from our login database.

The following error checks will be done during the modify account portion of the application. The username will first verify that the new username is unique. If the username is not unique then they user will be redirected to the modify account page after a message displays stating that the username must be unique. Next, the username and password will be verified to not be NULL. If either the username or password are NULL then the user will be redirected to the modify account page after a message displaying that they must enter a username and password is displayed. Finally, if the user enters a username or password the login database will be checked to verify that a new username or password was entered. If both the username and password are the same, then the user will be redirected to the modify account page after a message is displayed stating that there were no changes to the account login information. Finally, the user will be able to modify their account avatar from the modify account page.

CSCI comp3 consist of the questionnaire portion of the application. The questionnaire will be accessed from the homepage. The portion consist of the questionnaire webpage, the questionnaire interface, and interacts with the data collected from the Census Bureau in the data database. Within this portion of the application the user can give a ranking to a list of statistics depending on how important that statistic matters to them. These statistics will then be used to give a user score to each county in the data database. The list output will consist of the top 10 counties based on the scoring the user provides.

This portion of the application will be written using the Angular Framework and CSS for the webpage design. For the manipulation of variables within the webpage Typescript, HTML, and JavaScript will be used. For the database, Angular provides an interaction with data that is stored with in a JSON format through a PHP server running MYSQL. This will be how we will extra information from our database. Python will also be used for the mathematics for calculating the scoring of each county within the database.

The following error checks will be done within the questionnaire portion of the application. The user must enter a numerical value for each statistic in the range of zero to ten. If there is a statistic without an input, then the user will be redirected to the questionnaire page after a message is displayed stating that all statistics require an input. An error check will also be done to verify that the user enters integer values. If the user enters a decimal value, then the user will be redirected to the questionnaire page after a message is displayed stating that only integer values may be used.

CSCI comp4 consist of the map output within the homepage. Upon initial access to the homepage the map will be empty besides the output of the United States. After successful completion of the questionnaire the map will interact with the top ten location list provided from the questionnaire output. The top ten locations will have pins on the map corresponding to their location. A user can then click on the pin which will display an image and description of the location.

The map will be embedded into the homepage which will be using the same Angular, CSS, Typescript, HTML, and Javascript format as the other pages. However, the map itself will be displayed using the GoogleMaps API. Within the GoogleMaps API there is the means to creating the pins for each location provided submission of the county unique identifier, or a longitude and latitude corresponding to the location of the pin. For the information on each pin, there will be the integration of the GooglePlaces API. From here, we can receive both an image and description of a location provided submission of the county unique identifier, or a longitude and latitude corresponding to the location.

The map visualization will be done as a complete backend portion of the project. There will be no need for user input verification or error checking.

# Requirements traceability

This section shall contain:

1. Traceability from each software unit identified in this SDD to the CSCI requirements allocated to it. (Alternatively, this traceability may be provided in [4.1.)](#_bookmark1)
2. Traceability from each CSCI requirement to the software units to which it is allocated.

The following table will show the interaction between the requirements within the SRS requirements tables, and the CSCI components within this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Number | Requirement Summary | Test | CSCI component # |
| 3.1 | State Requirements | Meets 3.1.a-e | System |
| 3.1.a | Login page | Meets 3.1.a.1-4 | Comp1 |
| 3.1.a.1 | Access account creation |  | Comp1 |
| 3.1.a.2 | Username input |  | Comp1 |
| 3.1.a.3 | Password input |  | Comp1 |
| 3.1.a.4 | Authentication |  | Comp1 |
| 3.1.a.5 | Admin\*optional |  | Comp1 |
| 3.1.b | CAP | Meets 3.1.b.1-5 | Comp2 |
| 3.1.b.1 | Create username |  | Comp1 |
| 3.1.b.2 | Unique username |  | Comp1 |
| 3.1.b.3 | Create Password |  | Comp1 |
| 3.1.b.4 | About me |  | Comp2 |
| 3.1.b.5 | Avatar |  | Comp1 |
| 3.1.c | Homepage | Meets 3.1.c.1-4 | Comps 1,4 |
| 3.1.c.1 | Original Map |  | Comp1 |
| 3.1.c.2 | Access modify account page |  | Comp1 |
| 3.1.c.3 | Access questionnaire |  | Comp1 |
| 3.1.c.4 | Updated Map |  | Comp4 |
| 3.1.c.5 | Avatar Display |  | Comp2 |
| 3.1.d | Modify Account Page | Meets 3.1.d.1-4 | Comp2 |
| 3.1.d.1 | Change username |  | Comp2 |
| 3.1.d.2 | Change password |  | Comp2 |
| 3.1.d.3 | Change ‘About Me’ |  | Comp2 |
| 3.1.d.4 | Change Avatar |  | Comp2 |
| 3.1.e | Questionnaire | Meets 3.1.e.1-3 | Comp3 |
| 3.1.e.1 | Display Statistics |  | Comp3 |
| 3.1.e.2 | Rate statistics |  | Comp3 |
| 3.1.e.3 | Updated Map Algorithm | Meets 3.1.e.3.a-c | Comp3 |
| 3.1.e.3.a | List top 10 |  | Comp4 |
| 3.1.e.3.b | Pin Information |  | Comp4 |
| 3.1.e.3.c | Multiple Questionnaires |  | Comp4 |
| 3.2 | Capability Requirements | Meets 3.2.a-e | Comp1 |
| 3.2.a | Login Page | Meets 3.2.a.1-2 | Comp1 |
| 3.2.a.1 | Successful login |  | Comp1 |
| 3.2.a.2 | Unsuccessful login |  | Comp1 |
| 3.2.b | Create Account | Meets 3.2.b.1-4 | Comp1 |
| 3.2.b.1 | Unique username |  | Comp1 |
| 3.2.b.2 | Non-unique username |  | Comp1 |
| 3.2.b.3 | Empty Username |  | Comp1 |
| 3.2.b.4 | Empty Password |  | Comp1 |
| 3.2.b.5 | Choose Avatar |  | Comp1 |
| 3.2.c | Homepage | Meets 3.2.c.1-3 | Comp4 |
| 3.2.c.1 | Empty Map |  | Comp4 |
| 3.2.c.2 | Top 10 |  | Comp4 |
| 3.2.c.3 | Pin Info |  | Comp4 |
| 3.2.d | Modify Account | Meets 3.2.d.1-3 | Comp2 |
| 3.2.d.1 | Empty Username |  | Comp2 |
| 3.2.d.2 | Unique Username |  | Comp2 |
| 3.2.d.3 | Empty Password |  | Comp2 |
| 3.2.e | Questionnaire | Meets 3.2.e.1-5 | Comp3,4 |
| 3.2.e.1 | Display Statistics |  | Comp4 |
| 3.2.e.2 | Empirical Statistics |  | Comp3 |
| 3.2.e.3 | Statistical Priority |  | Comp3 |
| 3.2.e.4 | Varied Output |  | Comp3 |
| 3.2.e.5 | Updated Map |  | Comp4 |
| 3.5 | Data Requirements | Meets 3.5.a-c | Comp1,2 |
| 3.5.a | Database usernames |  | Comp1,2 |
| 3.5.b | Database passwords |  | Comp1,2 |
| 3.5.c | Database Avatar |  | Comp1,2 |

# Notes

This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

# Appendixes

Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

DESCRIPTION/PURPOSE

The Software Design Description (SDD) describes the design of a Computer Software Configuration Item (CSCI). It describes the CSCI-wide design decisions, the CSCI architectural design, and the detailed design needed to implement the software. The SDD may be supplemented by Interface Design Descriptions (IDDs) and Database Design Descriptions (DBDDs).

APPLICATION/INTERRELATIONSHIP

Portions of this plan may be bound separately if this approach enhances their usability. Examples include plans for software configuration management and software quality assurance.

The Contract Data Requirements List (CDRL) should specify whether deliverable data are to be delivered on paper or electronic media; are to be in a given electronic form (such as ASCII, CALS, or compatible with a specified word processor or other support software); may be delivered in developer format rather than in the format specified herein; and may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document.

PREPARATION INSTRUCTIONS

General instructions.

a. Automated techniques. Use of automated techniques is encouraged. The term "document" in this means a collection of data regardless of its medium.

b. Alternate presentation styles. Diagrams, tables, matrices, and other presentation styles are acceptable substitutes for text when data required can be made more readable using these styles.

c. Title page or identifier. The document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem, or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of the preparing organization; and distribution statement. For data in a database or other alternative form, this information shall be included on external and internal labels or by equivalent identification methods.

d. Table of contents. The document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data in a database or other alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.

e. Page numbering/labeling. Each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data in a database or other alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.

f. Response to tailoring instructions. If a paragraph is tailored out of this document, the resulting document shall contain the corresponding paragraph number and title, followed by "This paragraph has been tailored out." For data in a database or other alternative form, this representation need occur only in the table of contents or equivalent.

g. Multiple paragraphs and subparagraphs. Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability.

h. Standard data descriptions. If a data description required by this document has been published in a standard data element dictionary specified in the contract, reference to an entry in that dictionary is preferred over including the description itself.

i. Substitution of existing documents. Commercial or other existing documents, including other project plans, may be substituted for all or part of the document if they contain the required data.