## Design Documents

### **You will create two documents describing the design of your project. These documents should be in your project docs folder.**

### **Functional Specification. The document should have the following sections:**

### **Background. The problem being addressed.**

Urban health indicators are metrics that help assess and monitor the health and quality of life of people in urban areas. They cover a wide range of topics, including indicators that are health “determinants”, which include environmental, socioeconomic and health infrastructure factors that can influence population health, and indicators that represent health “outcomes”, such as life expectancy.

Multiple health indicators can be aggregated to generate one single urban health index (cite: <https://iris.who.int/bitstream/handle/10665/136839/9789241507806_eng.pdf>). The selection of indices and methods of aggregation are driven by the research goal and intended use. Based on prior knowledge in this subject area, we selected the following indicators - transportation accessibility, population density, street network density, unemployment density, health insurance coverage, economic diversity, walkability, and physical inactivity, which can be meaningfully aggregated to reflect the overall urban design infrastructure that influence health of the residents. Life expectancy is our primary outcome as it reflects the overall health status in a population. We will use census tract as our unit of analysis.

Although there are tools to visualize and analyze each indicator, no tool has been built to examine the aggregated effect of these specific indicators on life expectancy. Furthermore, no tool has been built to visualize how changes in one or more indicators can affect life expectancy on the census tract level. For this project, we will first train a model to aggregate these indicators, and build an interactive tool that can visualize the impact (derived from the trained model) of these indicators on life expectancy in each census tract in 10 US metropolitan areas.

### **User profile. Who uses the system. What they know about the domain and computing (e.g., can browse the web, can program in Python)**

Our intended users are local policy makers and invested individuals who are interested in learning about their city and neighborhood’s overall health status (life expectancy) and how it is influenced by certain urban indicators. The intended users are expected to have some intuition on what each urban indicator represents but may not understand the details of how each indicator is collected. They likely are not experts in computing but can browse the web with minimal instruction. They want the web interface to be clear and easy to follow, with adequate explanation when it comes to domain specific knowledge.

### **Data sources. What data you will use and how it is structured.**

The urban indicators and life expectancy for each census tract are publicly available and downloaded from xxxxx. For each metropolitan area and indicator, there is a tabular data file with the fields xxxx

The shapefile of each census tract is downloaded from xxx

### **Use cases. Describing at least two use cases. For each, describe: (a) the objective of the user interaction (e.g., withdraw money from an ATM); and (b) the expected interactions between the user and your system.**

**Use case 1:** A city-level policy maker may want to get an overview of the life expectancy across each census tract in its jurisdiction. They want to quickly get an idea of how each part of the city is performing (in terms of life expectancy) so they can better understand the geographical health disparities in the city. In addition, they may want to know the census tracts that have the poorest life expectancy. The expected interaction between this user will be:

**Input:** user selects 1 out the 10 metropolitan areas

**Output:** the webpage displays a choropleth map containing the census tracts in this metropolitan area, shaded according to the life expectancy, and a list of the 5 census tracts that have the lowest life expectancy.

**Use case 2:** The policy maker may want to zoom in on a certain census tract that they are interested in and look at the latest values of the urban health indicators.

**Input:** user select a census tract from the choropleth map by clicking the shape of the census tract

**Output:** a panel displaying the life expectancy in the census tract, its percentile in the distribution of life expectancy across all census tracts, and values of the urban indicator measurements.

**Use case 3:** A local policy maker or a policy researcher may be interested in which urban indicators influence life expectancy the most and want to see how future policy changes would influence life expectancy.

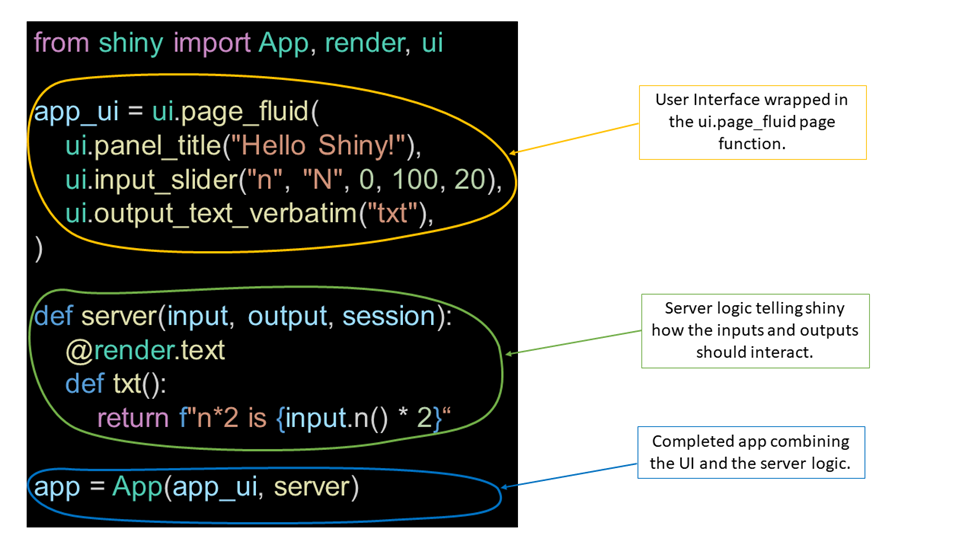
**Input:** slide bar to change the “future” urban indicators

**Output:** the predicted change in life expectancy in a given census tract when one or more areas of the urban infrastructure changes.

### **Component Specification. The document should have sections for.**

### **Software components. High level description of the software components such as: *data manager*, which provides a simplified interface to your data and provides application specific features (e.g., querying data subsets); and *visualization manager*, which displays data frames as a plot. Describe at least 3 components specifying: what it does, inputs it requires, and outputs it provides.**

* + Drop down menu that allows the user to select a metropolitan area
  + A database that stores the census level data
  + Interactive map to display the shaded census tract shapes on top of a street map
  + A display panel that shows the average life expectancy in a metropolitan area
  + A display panel that pops up to display the life expectancy, percentile, and urban indicator values when a census tract is selected
  + A machine learning model that uses urban indicators to predict life expectancy
  + Slide bars that allows users to change values of the urban indicator values
  + A display panel that display the “predicted” life expectancy when urban indicator values are changed
* **Spatial Mapping:** mapping the health indicators to the geographic area information
  + **Geopandas:** Used to transform spatial data into a dataframe that can be manipulated and analyzed in python
    - **Inputs:** shape files from the census database
    - **Outputs:** a geographic data frame that can be passed as an object to other tools like folium to create maps or to other libraries for data analysis.
  + **Folium:** a python library that supports creation of interactive maps with connectivity to OpenStreetMap geographic information, and leaflet.js.
    - **Inputs:** geographic data frame, and map design and attribute information (e.g. type of map, coloring, zoom scale etc.)
    - **Outputs:** interactive map with the specified attributes combined with the spatial layering from the geographic data frame.
* **Data Analysis Visualization of Health Metrics:**
  + **Pandas:** open source python data analysis and manipulation tool
  + **Matplotlib:** used to create static and interactive python data visualization, helpful to create numeric plots and visualizations for health indicator metrics
* **UI Design:** User portal to specify geographic region of interest and choose relevant health indicators
  + UI features that will be involved
    - Drop down menu that allows the user to select a metropolitan area
    - A database that stores the census level data
    - Interactive map to display the shaded census tract shapes on top of a street map
    - A display panel that shows the average life expectancy in a metropolitan area
    - A display panel that pops up to display the life expectancy, percentile, and urban indicator values when a census tract is selected
    - A machine learning model that uses urban indicators to predict life expectancy
    - Slide bars that allows users to change values of the urban indicator values
    - A display panel that display the “predicted” life expectancy when urban indicator values are changed
  + Tools that support creation and implementation of the UI features:
    - **Folium:** a python library that supports creation of interactive maps with connectivity to OpenStreetMap geographic information, and leaflet.js. Advantages of using folium over other libraries like plotly is that it can integrate with html web development for rich visualizations.
      * **Inputs:** map interactivity attribute specification e.g. map coloring overlay, or click on chart, vs. hover infographic.
      * **Outputs:** described interaction relative to the intended geographic area
    - **Shiny Python :** an open source web framework to build web applications using python. This will be essential for the user interactivity component of our project. Once the data analysis, visualization and spatial mapping components are complete. Shiny python will be used to host our tool and for end users to interact with the tool directly.
      * **Inputs:** shiny app account info, rsconnect-python package installation, user interface (UI) specification, and server logic. See figure below.



Source: [Datacamp](https://www.datacamp.com/tutorial/mastering-shiny-for-python-a-beginners-guide-to-building-interactive-web-applications)

* + - * Outputs:
        + A cloud-hosted web application displaying interactive maps and data visualization plots for which users can specify feature parameters.

### **Interactions to accomplish use cases. Describe how the above software components interact to accomplish at least one of your use cases.**

* + **Use Case 1:** The user begins by selecting one out of ten metropolitan areas on the first landing page. After selecting the area, the next screen displays a dashboard created by Shiny Python that includes spatial mapping created by Geopandas and Folium from the Pandas library. This map showcases different coloring and zoom levels across various areas, shaded to represent life expectancy. At the top of the map, an information bar displays the average life expectancy for the selected metropolitan area, helping the user quickly access an overview of life expectancy levels across census tracts and relevant health indicators.
  + **Use Case 2:** For a deeper understanding of the census tract level, the user can click on the information bar to view the latest urban health indicators and data for a specific census tract of interest. Clicking on this bar opens a new display panel that shows life expectancy, percentiles, and urban indicator values for the selected census tract. The data visualization includes a life expectancy distribution graph for each city, city-level statistics, and census tract-level statistics. It also features a selection box for indicator values and a bar graph for the importance of each census tract indicator, displayed using the Matplotlib package in Python.
  + **Use Case 3:** Finally, the user can adjust urban indicator values to explore future scenarios and understand how policy changes might impact life expectancy. This feature is powered by a machine learning model that predicts life expectancy based on urban indicators. The display panel then shows the predicted changes in life expectancy within a given census tract as adjustments to urban infrastructure are simulated.

### **Preliminary plan. A list of tasks in priority order.**

* + Conduct data search and collection for each metropolitan area.
  + Clean each dataset to ensure accuracy and consistency.
  + Document the project overview, background, user profiles, and use cases.
  + Design a potential dashboard UI in Figma to visualize project features.
  + Build machine learning models using the cleaned dataset to predict future changes in life expectancy.
  + Explore various mapping tools to identify the best data visualization methods for the project.
  + Create spatial mappings using the Geopandas and Folium libraries.
  + Develop the Figma-designed UI using the Shiny Python package.
  + Generate data visualizations for life expectancy distribution by city, including city-level and census tract-level statistics, with selectable values and indicator options. Use the Matplotlib package to create a bar graph showcasing the importance of each census tract indicator.
  + Integrate all information and elements into a cohesive project structure.
  + Prepare for the final presentation.

### **Steps in Design:**

Figma = first page

User map

User interface design

1.**Identify the users**

–Describe who your users are and what they want from the tool at a high level

Citizens who want to know their neighborhood

(Policy makers and researchers who want to know the cities’ neighborhoods’ health conditions and exerts influence on it – if we have time)

**Stakeholders:**policymakers who want to better understand how a given neighborhood is influenced by various health indicators.

**Usecases:**

* **Research & Policy** - association of health indicators (in our case, life expectancy) with SDOH(social determinants of health) to understand sub-population resources development
* **Research & Policy of the community:** Ifexert certain influence on this neighborhood, how would the health evaluation be improved in general

2.**Functional design**:

–Describe what the system does (use cases)

* Maps a given geographic input - neighborhood name/zipcodes to census tracts
* Identifies ranked order of health indicators based on defined geographic region (zipcode/neighborhood)

3.**Component design**: Specify the components

–Each use case has a "Top level" component.

Need a mapping from the search terms a user types in to the census tract.

–Sub-components implements portions of the use case

User Story:

**Usecases:**

Description

Input

Output

**Components:**

What kind of components are having sub components?

User Interface

Authentication

**Tech review:**

Options

Why you need this tech,

Pros and cons