Your design document will need the following sections:

1. Executive Summary - a quick narrative (1 page at most) describing your design.
2. Special Terms/Definitions - define any terms and abbreviations that may not be common knowledge that you will use in your design (I usually do this section last even though it goes near the beginning).
3. Requirements Map - a re-statement of the business requirements (required and stretch) and a paragraph or two on how each will be satisfied in the design.
4. Class diagrams/UML - An actual UML diagram (using visio or whatever) that shows the classes that you intend to build, and their major properties, events, and methods (you do not need to catalog fields at this time).
5. Data flow diagram(s) - A more traditional flow chart showing the flow of data in your application.
6. Database/Storage (if applicable) - If you are using databases this should be a diagram of each table, field, type, etc.  If you are not then this should be a detailed design of any storage mechanism that you will use.
7. GUI/Screen Shots - If you are using a GUI you need screen shots/mock ups to show what you will build.
8. Test Cases - I haven't set a minimum number of cases, but be sure that all of your functionality is covered.

**Executive Summary**

This program will be a very basic chatbot. Essentially it will simulate very basic Artificial Intelligence (AI) through interaction with the user. The program will accept input in the form of grammatically correct English and respond appropriately—also in English.

The program will utilize dictionary files to detect words and determine the part of speech they belong to. (i.e. noun, verb, article, etc.) It will then attempt to construct a comprehensible sentence by analyzing how those parts of speech should interact based on a set of rules.

The program will then attempt to construct an appropriate response to the user’s input. It will use the words received from the user as well as words in its dictionary files to formulate a response in the form of a statement or question related to the user input.

The program will not be required to handle incorrect grammar, slang, profanity, obscure vocabulary or non-English words. Once the minimum requirements are met adding that functionality is optional.

**Business Requirements**

* Required
  + Must accept a sentence from the user
  + Must respond with a relevant statement or question
  + Must at least interact through the console
* Stretch
  + Can store and “remember” new words it learns through interaction with the user
  + Can detect and decipher part of speech an unknown word belongs to based on context
  + Can interact through a basic Graphical Interface (GUI)
  + Can detect and correct errors in spelling and grammar