

## 1.1 REQUIREMENTS & CONSTRAINTS

### Functional Requirements:

- The application should be able to display the determined route that fulfills the input of the user.
  - In case there are multiple routes, the application should clearly indicate them.
- The user should be able to easily input, view, and modify the current choices for points of interests and starting locations.
- Stretch goal: Incorporate other attributes in ranking the selections i.e., rank hotels by price or stars; rank museums by entry fee, etc.
- The application should allow the user to specify which algorithm they want their route to be determined by.
- The application should be scalable i.e., allow multiple users to use it.

### Nonfunctional Requirements:

- Design of the code should be extendable for easy addition of more areas/cities.
- Application should be able to reliably perform the task of input and creating routes.
- The user experience should be intuitive and clean.
- The code implementing every functionality should be designed with maintainability and extensibility in mind. For example,
  - UI components should be added over time
  - Data for additional cities should be easily adoptable

### Constraints:

- Route should be fully displayed within seconds after input submission.
- Route visualization should be accurate.
- Total cost of implementation should not exceed \$300.
- Time: 2 semesters

## 1.2 ENGINEERING STANDARDS

- IEEE Std 1012, Standard for Software Verification and Validation
  - “Verification and validation (V&V) processes are used to determine whether the development products of a given activity conform to the requirements of that activity and whether the product satisfies its intended use and user needs.” (1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation, 2016)
  - This standard will help this project to meet and or exceed the expectations and requirements of users and their needs.
- IEEE Std 1219, Standard for Software Maintenance
  - “provides the framework within which generic and specific software maintenance plans may be executed, evaluated, and tailored to the maintenance scope and magnitude of given software products.” (1219-1998 - IEEE Standard for Software Maintenance, 1995)
  - This standard will help this project to allow future developers to be able to easily maintain this software.
- IEEE/ISO/IEC 26512-2017, Requirements for acquirers and suppliers of information for users

- “This document was developed to assist users of ISO/IEC/IEEE 15288:2015 or ISO/IEC/IEEE 12207 to acquire or supply information for users as part of the system or life cycle processes. It defines the documentation process from the acquirer's standpoint and the supplier's standpoint.” (26512-2017 - ISO/IEC/IEEE International Standard - Systems and Software Engineering - Requirements for Acquirers and Suppliers of Information for Users, 2017)
- This standard will help this project to improve users quality and availability of information about the software.
- IEEE Std 982.1, Standard Dictionary of Measures to Produce Reliable Software
  - “This standard provides measures that are applicable for continual self-assessment and improvement of the software aspects of dependability.” (982.1-2005 - IEEE Standard Dictionary of Measures of The Software Aspects of Dependability, 2005)
  - This standard will help this project to create reliable and dependable software.
- IEEE/ISO/IEC 15288-2015, System life cycle processes
  - “This International Standard establishes a common framework of process descriptions for describing the life cycle of systems created by humans.” (15288-2015 - ISO/IEC/IEEE International Standard - Systems and Software Engineering -- System Life Cycle Processes, 2015)
  - This standard will aid us with definitions for the entire life cycle of the system, i.e., conception, development, production, utilization, support, and retirement.
- ACM 1.3
  - “Be honest and trustworthy.” (ACM Code of Ethics and Professional Conduct, n.d.)
  - This standard will aid this project developing software more ethically.

### 1.3 REFERENCES

982.1-2005 - IEEE Standard Dictionary of Measures of the Software Aspects of Dependability. (2005). IEEE Xplore. <https://ieeexplore.ieee.org/document/1634994>

15288-2015 - ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes. (2014). IEEE Xplore. <https://ieeexplore.ieee.org/document/7106435>

26512-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering - Requirements for acquirers and suppliers of information for users. (2017). IEEE Xplore. <https://ieeexplore.ieee.org/document/8288807>

1219-1998 - IEEE Standard for Software Maintenance. (1995). <https://ieeexplore.ieee.org/document/720567>

1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation. (2016). <https://ieeexplore.ieee.org/document/8055462>

ACM Code of Ethics and Professional Conduct. (n.d.). Association for Computing Machinery. <https://www.acm.org/code-of-ethics>