**Travel Ceylon - An Intelligent Trip Planner**

Andun Sameera

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 3.0

**Abstract**

This is a Travel planning application which is based on Sri Lanka. User can plan their own travel according to the following constrains, after processing all those constrains the Travel Ceylon app will suggest you are travel path to follow and list of interesting place which you can see on the path. I. Interest list of the traveler. II. Starting position. III. Time period of travel. IV. Special places to be included. V. Special places to be avoided. On the way of travel, if the traveler reaches a particular important place which was highlighted in the travel plan, app will pop up and give you the details about the arriving place and all the guidelines to arrive the place.

**In tro duction**

This document provides a high level overview of the software architecture for the Travel Ceylon Guiding System which is a navigational aid software application which process user’s inputs and gives user a good travel plan according to their choices and interests.

The document provides a high-level description of the goals of the architecture, the use cases support by the system and architectural styles and components that have been designed using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

**II. Purpose**

Travel Ceylon is truly a guiding system which is able to get user’s likes and dislikes as inputs. Then it can process them using the help of data store which held data about important places in Sri Lanka. After that it gives s travel plan to the user with important places marked in that, which they can visit. So do that task this system has to be architected well. This document will give the detailed description of the architecture which is designed for Travel Ceylon guiding system.

This document will give a description of the architecture in the best known way to represents which is the 4+1 model. It will give logic view, implementation view, deployment view and the process view of the architecture of the Travel Ceylon system. Also this document will give the use case which describes all these view points of the architecture.

**II. Scope**

1http://creativecommons.org/licenses/by/3.0/
The scope of this software architecture document is to depict the architecture of the Travel Ceylon Guiding system. This document describes the aspects of Travel Ceylon Guiding design that are considered to be architecturally significant; that is, those elements and behaviors that are most fundamental for guiding the construction of Travel Ceylon Guiding and for understanding this project as a whole. This will help anyone to understand the travel Ceylon guiding system and its functionality clearly.

II. Overview

Up to this point we have set up the basement for the Software Architecture Document of the Travel Ceylon System. In this section we will talk about the structure of this document beyond this point. It will be a help to understand the document easily.

- **Architectural Representation** of this document will give the architecture in 4 different perspectives. They are Logical view, Implementation View, Deployment View and Process view. At last we will give a brief idea of each of these through some use cases.
- The section **Architectural Goals and Constraints** will describe the goals and constrains of the system which related to the design of the architecture.
- **Use-Case View** of this document will describe the use cases which are incorporate with the Travel Ceylon system.
- **Logical view** of this document describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages.
- The **Process view** of the architecture document will describe the structure of the process which is parts of the development process of the Travel Ceylon system.
- **Deployment view** will describe how the Travel Ceylon will be deployed in the environment.
- **Implementation view** of the architecture will describe how the system will be implemented in architectural manner. Also this section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.
- **Data View** is a description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data or the translation between the Design Model and the Data Model is trivial.
- **Size and Performance** is a description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.
- **Quality** will be a description of how the software architecture contributes to all capabilities of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated.

---

 Architectural Representation

The architecture of Travel Ceylon Guiding system is described using the 4+1 model of the architecture representation.

- **Logical view** is for Design purpose. It will describe Functional Requirements: describes the design's object model. Also describes the most important use-case realizations and business requirements of the system.
- **Process view** is for Integration. It describes Non-functional requirements: describes the design's concurrency and synchronization aspects.
- **Implementation view** is for Programming. It will describe Software components: describes the layers and subsystems of the application.
- **Deployment view** is for do a good Deployment. It describes Topology: describes the mapping of the software onto the hardware and shows the system's distributed aspects. Describes potential deployment structures, by including known and anticipated deployment scenarios in the architecture we allow the implementers to make certain assumptions on network performance, system interaction and so forth.

http://cnx.org/content/m43405/1.1/
- Use Case view will describe the scenarios of the system. It describes the set of scenarios and/or use cases that represent some significant, central functionality of the system. Describes the actors and use cases for the system, this view presents the needs of the user and is elaborated further at the design level to describe discrete flows and constraints in more detail. This domain vocabulary is independent of any processing model or representational syntax.

- Data view is for doing the data handling of the Travel Ceylon. It Persistence: describes the architecturally significant persistent elements in the data model.

## Architectural Goals and Constraints

- There is a central data store which is accessed by Android Clients of Travel Ceylon system.
- Number of clients requires services from central data store at same time.
- Data Store should be up all the time.
- Clients should connect to the central data store via the Internet.
- Client Users and can update the central data store. (Updates should be approved by Admin)
- Any user can update the central data store via the web. (Updates should be approved by Admin)
- Central Data Store should have good performance and good response time.
- All the data in the data store should be stored in a relational database.
- Central data store should be secure enough.

## Use-Case View

The Use Case describes the set of scenarios and/or use cases that represent some significant, central functionality. It also describes the set of scenarios and/or use cases that have a substantial architectural coverage or that stress or illustrate a specific, delicate point of the architecture. The use cases in this system with respect to actors of the system are listed below.

- Select Travel Planner or Place Data Insert option.
- Give the predefined list of details about the travel.
- Get and follow the travel plan.
- View a place highlight in the travel Plan.
- View a place pop ups on the Travel.
- Add details about a place.
- Verify details given by regular users.

### Use-Case Realizations

1. **Select Travel Planner or Place Data Insert option:**

   In the Travel Ceylon android client there are two major options. They are making a travel plan or add details about an important place in Sri Lanka. User can select any of these two choice single choices at once.

   - **Give the predefined list of details about the travel:**

     If the user of the Android client chooses Travel Planner option they are given list to fill, including potential destination of the visit, their interests, places like to visit, like to avoid, travel duration, starting place. Filling and submitting these will direct user to the next step.

   - **Get and follow the travel plan:**

     After giving the pre-defined inputs the travel plan will be shown. Then user can choose the follow option in the travel plan.

http://cnx.org/content/m43405/1.1/
**View a place highlight in the travel plan:**

In the travel plan user can view any place which are suggested to visit. They can see it all details by this option.

**View a place pop ups on the Travel:**

In the travel, if user reaches a place highlighted in the travel plan, Travel Ceylon will pop up this place and show its details to the user.

**Add details about a place:**

The use of the android client and the admin of the system can add new places to the central data store. They have to fill a pre-defined form to do this.

**Verify details given by regular users:**

Admin have to verify the places added by users before adding them to the central data store of Travel Ceylon.

**Logical View**

This application is a small one. So there are no special sub systems in this application. So Travel Ceylon will have following packages instead of sub systems. They are divided on the functionality it provides. These packages will have classes to accomplish the task assign to these packages.

**Data Access Server:**

This package will have classes to access the central database to add, get, update, and delete data which exits in the central data store.

**Data Manipulating Server:**

This package will have classes to format the data before they are sent to the Data Access. They will extract the data coming as SOAP messages and HTTP messages to the server. Also it encodes data to SOAP or HTTP messages before they are send to Communication.

**Communication Server:**

This will handle the communication between the clients of the Travel Ceylon system. They can communicate through SOAP or HTTP.

**Communication Android Client:**

This will handle communication between the central data store and Android client of the system. They are connecting the web service though SOAP.

**Data Manipulating Android Client:**

This package will have classes to format the data before they are sent to the Communication. They will extract the data coming as SOAP messages and HTTP messages from the server. Also it encodes data to SOAP or HTTP messages before they are send to Communication.

**Logic Android Client:**

This will handle the all the logic of the Travel Ceylon’s major functions. This will have classes to search for paths, show important places etc.

http://cnx.org/content/m43405/1.1/
· **User Interface: Android Client:**

This will handle the communication between the user and the Android Client.

11. Overview

At the above section we describe the major packages which are located in the Travel Ceylon system. Now we look in to them in different aspect considering the layers of the system.

- User Interface: Android Client
  - UI Layer
  - Logic: Android Client
  - Logic Layer
  - Communication: Server, Communication: Android Client
  - Communication Layer
  - Data Manipulating: Android Client, Data Manipulating: Server, Data Access: Server
  - Data Access Layer

---

ñ• Process View

In the Travel Ceylon guiding system, there will be one major process runs at the central data store. Because central data store will reply to any request from android clients, so it means only one process is needed. That will be the web service which provides city and place details. Other process will be the process runs at the Android client of the Travel Ceylon. It will process user inputs to generate a travel plan.

- Travel Ceylon Admin
- Travel Ceylon Web User
- Request for approval process
- Send the data which needed approval
- Input details about a place
- Travel Ceylon Web Server
- Travel Ceylon Web Server
- Query Approval Table
- Send Approval Table’s data
- Send to approval table
- This is also pipeline looking process which need single process to handle the entire task which performed by the system. Above diagrams and following diagram will show the view of process.

- Travel Ceylon User
- Travel Ceylon User
- Input Pre Defined Set of Inputs
- Give the Travel Plan
- Input details about a place
- Travel Ceylon Android Client
- Travel Ceylon Android Client
- Requires some data about cities
- Send data about cities
- Sends place data
- Travel Ceylon Web Service
- Travel Ceylon Web Service
- Query data about a city
- Fetch and give data about
- Send to approval table

ñ• Deployment View

http://cnx.org/content/m43406/1.1/
The Travel Ceylon system will have a central data store and Android mobile clients after successfully deployed. This Deployment diagram will show you those components of the system.

---

Implementation View

This will be same like the logic view of the Travel Ceylon System. Because this is a simple kind of software we use the implantation view same as the logic view have. The major reason for that is we don’t need a very complex logic view for this system.

1. Overview

These packages will be implemented in the Travel Ceylon system. They will look after all the functionality in the system.

- **Data Access Server**:

This package will have classes to access the central database to add, get, update, and delete data which exits in the central data store.

- **Data Manipulating Server**:

This package will have classes to format the data before they are sent to the Data Access. They will extract the data coming as SOAP messages and HTTP messages to the server. Also it encodes data to SOAP or HTTP messages before they are sent to Communication.

- **Communication Server**:

This will handle the communication between the clients of the Travel Ceylon system. They can communicate through SOAP or HTTP.

- **Communication Android Client**:

This will handle communication between the central data store and Android client of the system. They are connecting the web service though SOAP.

- **Data Manipulating Android Client**:

This package will have classes to format the data before they are sent to the Communication. They will extract the data coming as SOAP messages and HTTP messages from the server. Also it encodes data to SOAP or HTTP messages before they are send to Communication.

- **Logic Android Client**:

This will handle the all the logic of the Travel Ceylon’s major functions. This will have classes to search for paths, show important places etc.

- **User Interface Android Client**:

This will handle the communication between the user and the Android Client.

---

http://cnx.org/content/m43405/1.1/
The Travel Ceylon Guiding System’s core functionality has been design with aid of the central database of the system. It gets the data about places each time it creates a travel plan on the application. So the data store runs a major role in this Travel Ceylon Architecture.

The major task of the central database is to hold the interrelated data about important places which are located in Sri Lanka. So two major things will be stored in the central database about a place, they are the particular set of information about the place and interconnectivity to adjacent important places. So we will choose a graph representation of places to hold data. Considering that we will create our relational database to hold data. In the following diagram we will explain how that concept of customized version graph representation will record the data in Travel Ceylon System. This will show a example map with important places in Sri Lanka.

- Aukana
- Thissa Wawa
- Anuradapura
- Gal Wiharaya
- Habaran Jungle
- Polonnaruwa
- Parakrama Smaudraya
- Kandy
- Dalada Maligawa
- Hanthana
- Victoria Power Station

This diagram show that each major town or the city and its nearby important places. So we will maximize this kind of graph all around Sri Lanka. So we can add towns and important places which are near to them. After having such kind of representation we can design a traversal algorithm based on the users inputs. So we can have travel plan through that map. We can use a traversal algorithm like Dijkstra to traverse this graph representation of the map.

So we have a concept of representing the data about places, in the program. Then we have to design the database and its tables which really hold those data and this concept. So next we give an entity relationship diagram which will describe the data and entity relationship.

City

- Important Place

Also this relationship will not be enough to do the traversal of graph to find the best travel path for the user. For that purpose we need another relationship which represents the adjacency of cities. So apart from the above ER diagram we will have another table with adjacencies. So each city will have a table with its adjacent cities and distance to them. Following structure will be the relational tables in the Travel Ceylon main database.

Table: City
- City Name: String
- Longitude: Integer
- Latitude: Integer

Table: Important Places
- Place Name: String
- Category: Integer
- Description: String
- Longitude: Integer
- Latitude: Integer

Table: Close to
- City Name: String
- Place Name: String
- Distance: Integer

Table: City Graph of City
- XCity Name: String
- Distance: Integer

Table: Important Places for Approval
- Place Name: String
- Category: Integer
- Description: String
- Longitude: Integer
- Latitude: Integer

Size and Performance

Here is the list which we identified as the influences in software architecture.

- Travel Ceylon Should Handle 200 Online users at the same time
- The database should be large enough to hold all the adding data about important places.
- Each user of the Travel Ceylon should be provided with travel plan within 5 seconds after they input their interests.
- The travel plan should be optimized enough to cover all the liking of the user.
- Travel Ceylon Central data store should be up and running all the time.

- Quality
- Should be easy to use.
- Travel Ceylon Central Data System’s reaction should be Scalable when user demands increase. So there will be a mechanism of redundant web serveries servers to handle the android clients of the Travel Ceylon system.
- To guarantee the Reliability, Availability there will be a mechanism of redundant web serveries servers to handle the android clients of the Travel Ceylon system.
- To achieve good Security Authentication and authorization mechanisms will be added to the central data stores login.
- The Android Client of the Travel Ceylon should be designed in way with satisfying standard usability requirements.