Ambulance Arrival Timing
Architecture/Design Document

Table of Contents

1 INTRODUCTION 3

2 DESIGN GOALS 3

3 SYSTEM BEHAVIOR 3

4 LOGICAL VIEW 4

4.1 High-Level Design (Architecture) 4

4.2 Mid-Level Design 5

4.3 Detailed Class Design 6

5 PROCESS VIEW 7

6 USE CASE VIEW 7

Change History

Version: 0.3
Modifier: Md Enamul Sumon
Date: 14/9/2017
Description of Change: Added sequence diagrams for self-directed mode.

Version: 0.2
Modifier: Md Enamul Sumon
Date: 23/08/2017
Description of Change: Added process view and use case view. Also added sequence diagram to logical view.

Version: 0.1
Modifier: Md Enamul Sumon
Date: 13/08/2017
Description of Change: Initial rough draft. Contains logical view (high-level modules only) and development view.

---

1 Introduction

This document describes the architecture and design for the Ambulance Arrival Timing web application being developed for the La Trobe University—Bundoora. Ambulance Arrival Timing is a web directed application system that generates estimated arrival time (ETA) of the arrival of an ambulance for the hospital emergency center. Axicor Data Science Solutions proposed the project. User can upload data (ambulance movement timing report from coordinates to coordinates) in a correct format in a text file. The uploaded data will be analyzed, and machine learning will be performed. The user then can get a report which will be generated by the system. The report can be chosen within a certain period of time. User then can select a location and get the instant ETA to the hospital from that location on that time.

2 Design Goals

The priorities for the design that follows are:

- To minimize complexity and overall idea of the project development.
- To give good direction to the development teams about the process to build the system.
- To ensure future usability and extendibility so that the project can be extended if needed in future.

3 System Behavior

The overall system behavior is as follows.

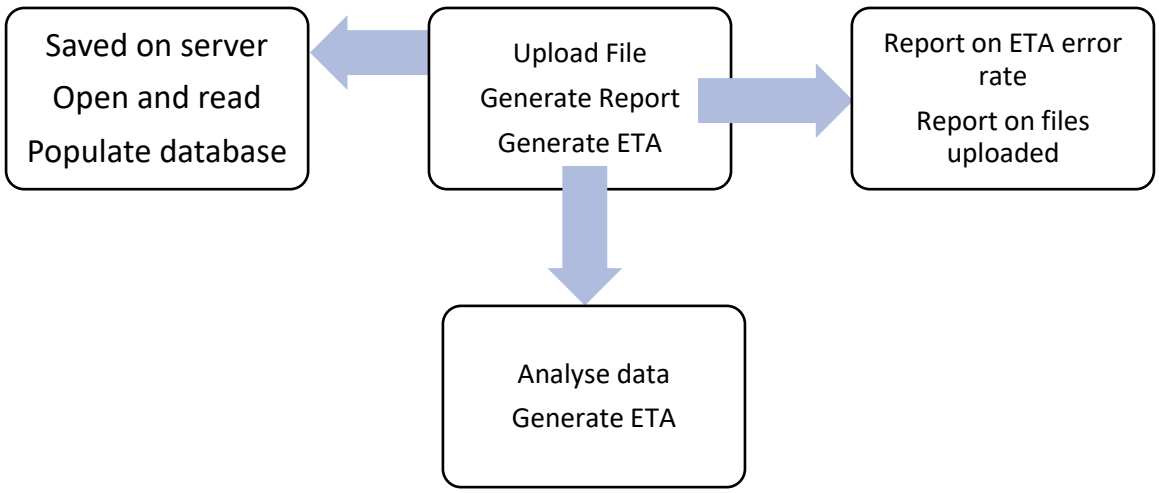


Figure 1 System Behavior

4 Logical View

4.1 High-Level Design (Architecture)

The high-level view or architecture consists of 5 major components:

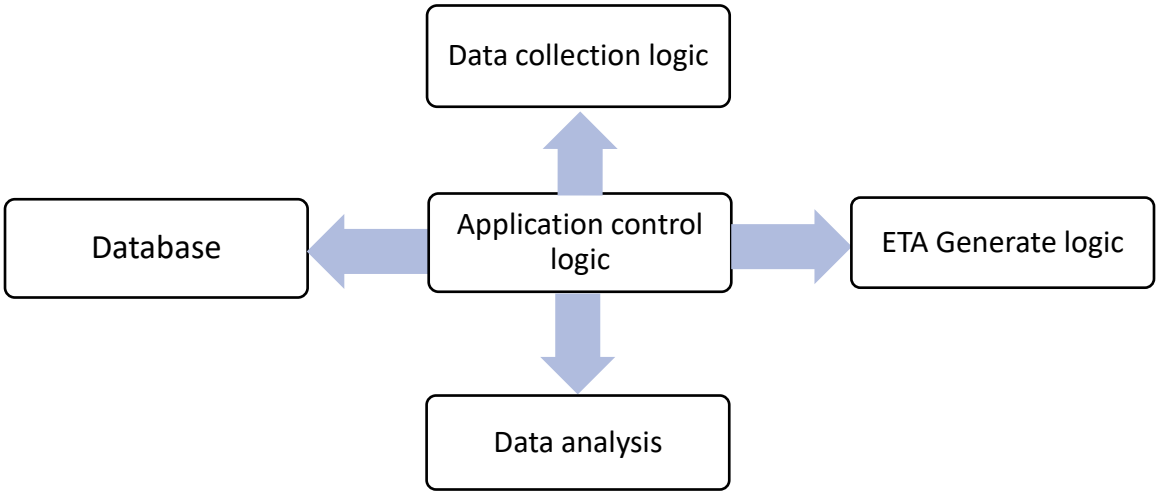


Figure 2 System Architecture

- The **Data Collection logic** is the controller for data collection and manipulation.
- The **Database** is a central repository for data on time estimation and way points.
- The **ETA Generate logic** is the function to generate ETA based on the data provided for the given route and time.
- The **Application Control Logic** is the main driver of the application. It presents information to the user and reacts to user inputs.

4.2 Mid-Level Design

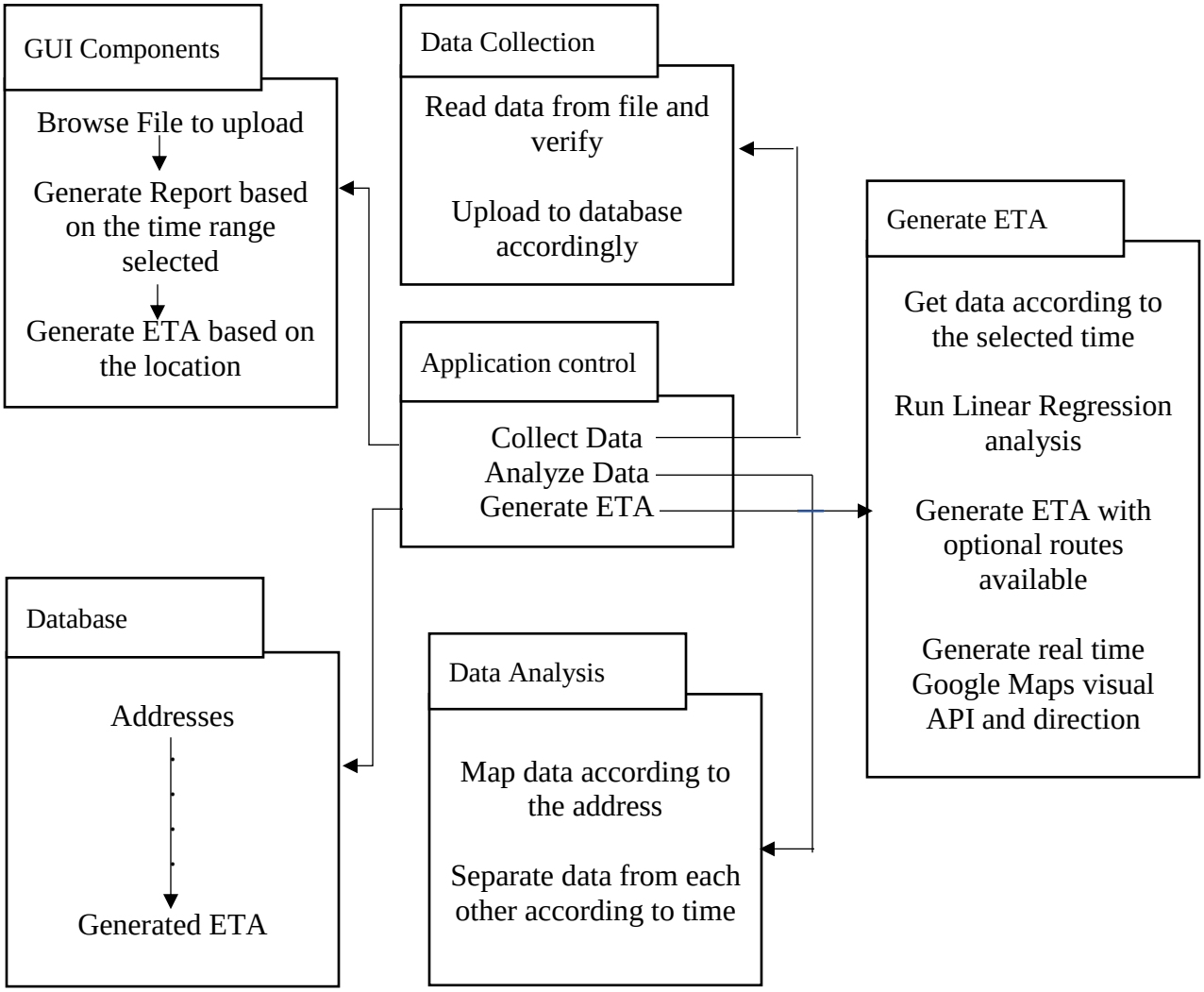
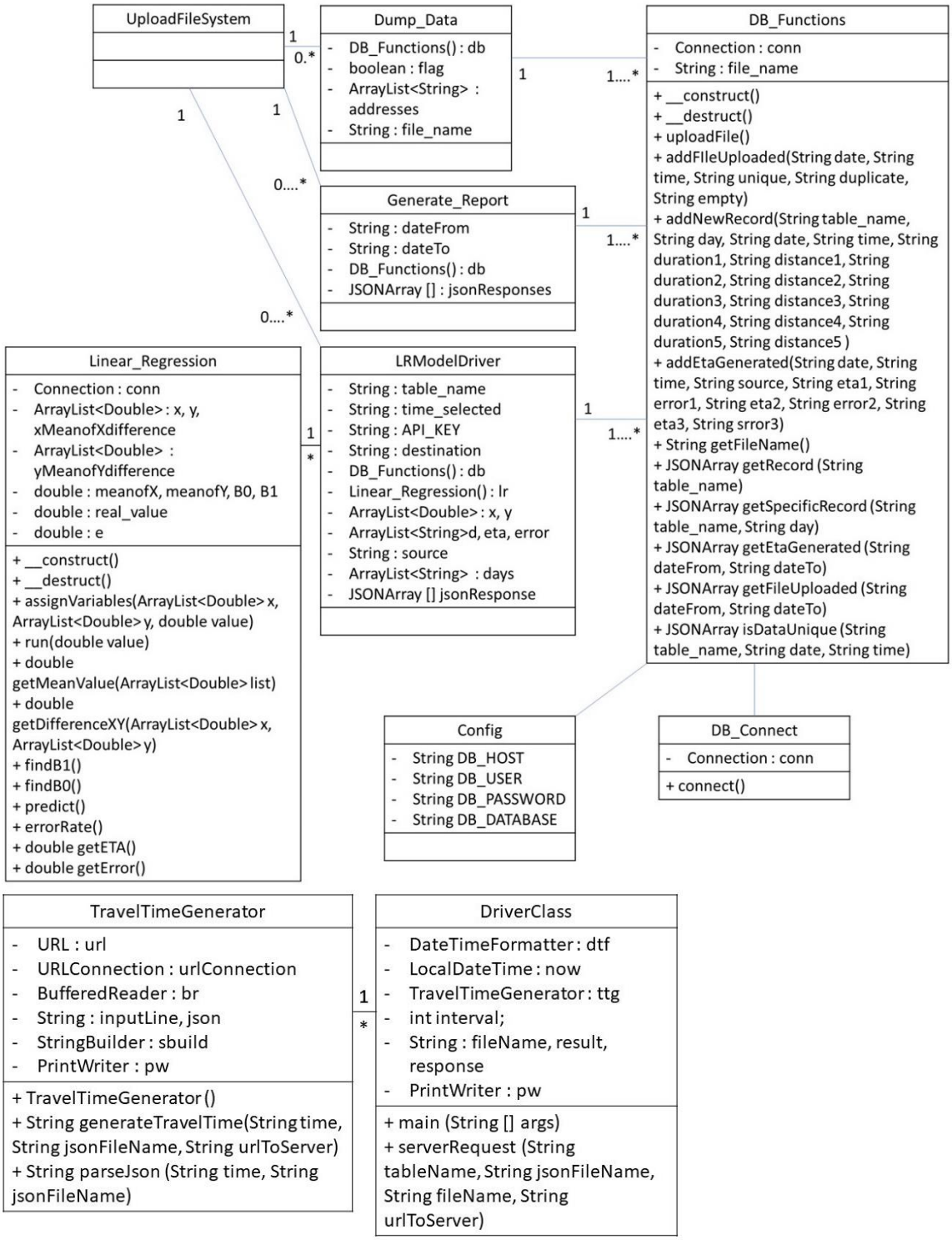


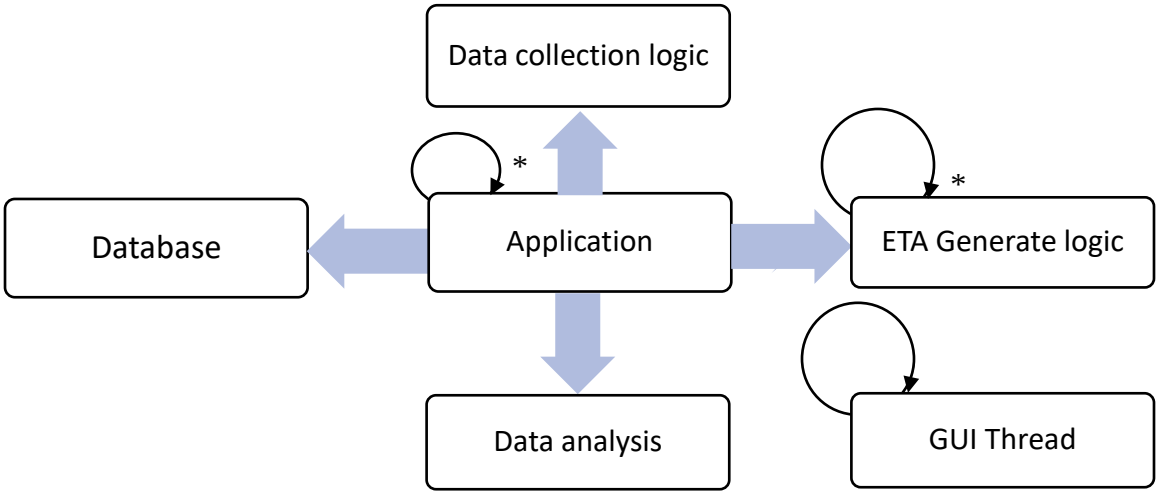
Figure 3 Mid-Level design components and their relationships

4.3 Detailed Class Design



5 Process View

Each loop represents a thread of control.



6 Use Case View

Upload File:
Goal: Successfully upload a text file
Actors: Active operator from operating a web browser
Main Flow:

1. User browse a text file from the hard disk
2. User clicks upload file button
3. System validates it's a text file
4. System opens the file to read data from it
5. System validates the line is correctly formatted
6. System validates the data line is not empty or blank
7. System validates the data on the line is unique
8. System records the unique data on the database
9. Repeat 5, 6, 7, 8 until system reaches the end of the file
10. System closes the file
11. System generates a summary of data recorded, discarded and empty lines

Extension:

3a) Uploaded file is not a text file
   1. System lets the user know
   2. System shuts the function and ready for a new start

5a) Line is not correctly formatted
   1. System lets the user know
   2. Closes the function

6a) Data line is empty or blank
   1. System records the incident to generate a report on it in the end

7a) Data is not unique

1. System records the incident for reporting

Generate Report:
Goal: Successfully generate report within for a time frame
Actor: Operator
Main Flow:

1. User selects a from a date to a date to generate report
2. System records the date
3. System goes through the database and locate the data in between that time frame
4. System gathers the data and generates a ETA generation report
5. System gathers the record for the number of files uploaded in the timeframe and generates a report
6. System gathers the error rate from the database according to the ETA generated and gives a report on it

Extension:

1a). User doesn't select any date

1. System goes with the default time frame and generates a overall report

Generate ETA:
Gola: Successfully generate ETA for a location to Austin Hospital
Actor: Operator
Main Flow:

1. User selects its current location
2. User enters a time manually or leaves is blank which means the current time on the user's device
3. System gathers the location and looks for data on the database
4. System looks for precious ETAs up to 30 mins around the time user selected on that given day
5. System validates at least 5 previous record around that time frame is in the database
6. System runs linear regression algorithm on those selected data
7. System predicts the ETA based on those record and generates ETA for multiple directions if available
8. System also generates a error rate to the ETA generated
9. System shows a real time Google Map interface with the Google Map prediction of the ETA.
10. System also shows the direction to the destination with instructions

Extension:

1a) Current location is not selected

1. System defaults to the first location available on the options menu

5a) Less than 5 records available

1. System lets the user know about that and doesn't provide any ETA hence returns to the main function