# CHAPTER 4: DESIGN PHASE

## 4.1 Introduction

Having done the project analysis which was done in the previous chapter, chapter 3, the project moves on to the design phase where the designing of the automated fare collection system will be done. The design goal or the chapter goal is to answer the “how?” part of the development process. A context diagram and a dataflow diagram are the first tools the developer is going to implement to understand and help others understand the system. An architectural design, physical design, database design interface design and a program design shall also be carried out.

## 4.2 System design

System design in software engineering is the process of defining the modules, architectures, interfaces and data for an information system or software or application so as to satisfy the system objectives (Whitten et.al, 2004). It involves detailing how a system meets the requirements from the analysis phase in the previous chapter. This is achieved through the use of flowcharts, dataflow diagrams, architectural designs and other system design methodologies.

### 4.2.1 Description of the proposed system

The proposed automatic fare collection will be comprised of a device, which records GPS coordinates of places where a passenger drops off or boards a bus. Passengers swipe an issued unique RFID card or get a journey token on boarding a minibus and the GPS coordinates at a point at which a customer boarded the minibus will be recorded. On dropping off a passenger will swipe the card again and an amount will be deducted from the customer account which will be deduced by the system from the pick-up point GPS coordinates and drop off point GPS coordinates.

### 4.2.2 Context diagram of the proposed system

Figure 4.1 Context diagram

### 4.2.3 Dataflow diagram for the proposed system

Figure 4.2 Dataflow diagram of the proposed system

## 4.3 Architectural design

Architectural design in software engineering is the modelling of the technical environment in which an information system works in. It outlines the software and hardware elements which are needed for the information system to function properly. The architecture for the system in this project is described below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Automatic Fare Collection Device |  | Web Server | Middleware | DBMS |
|  |  |  |  |  |
|  |  | Web browser |  |  |

Fig 4.3 Architectural design

From the diagram the AFC device communicates with the web server. The web server is the software used to access web pages. Apache will be the used web server in this project. The web server then feeds data into the Database Management System (DBMS) through a middleware. The DBMS stores and manages the data. There are different DBMS that can be used and this project makes use of the MySQL DBMS. The Middleware is the backend programming or scripting language that is used to manipulate operations sent by the web server. There are different web applications middleware that include Python, ASP.Net, ColdFusion, Ruby on Rails and PHP which is the middleware to be used in this project.

## 4.4 Physical design

Physical design is the network arrangement of an information system. It is mapped by the use of a network model, which is a diagram that depicts all nodes for a network. A client-server network model shall be used in this project where each vehicle with an AFC device acts as a node communicating with one central server. There will also be computers acting as nodes. These computers will be those being used by transport managers to view operations and get reports.

### Figure 4.4 Physical design

## 4.5 Database design

Database design refers to the definition of data elements to be employed by the software application (Jardine, 1977). Three methods will be used to model the database design and these are database architecture, enhanced entity relationship modelling and the database tables.

### 4.5.1 Database Architecture

Database architecture is an abstract design for a database management system. The American National Standards Institute – Standards Planning and Requirements Committee (ANSI-SPARC) shall be used to describe the database architecture (Jardine, 1977). The ANSI-SPARC architecture employs a three-level architecture and the objective for this is to separate the user’s view as it allows independent customised user views. Each user of the application access data from the same location but each user has a different customised view of the data.

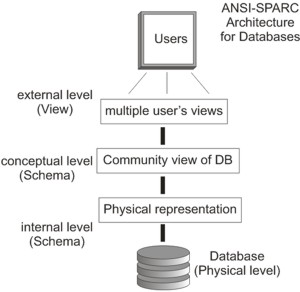


Fig 4.5 Database Architecture

### 4.5.2 Enhanced Entity Relationship Model

Enhanced Entity Relationship (EER) model is a conceptual view of the existisitng relationships between the tables of a database. This is expressed through an EER diagram. An EER diagram identifies all the tables of a database and their respective unique keys. Relationships between them are also displayed.

Draw me

Fig 4.6 EER Diagram

### 4.5.3 Database Tables

In DBMSs, a datababase table or just table is a set of data values with the same structure. A table has a specified number of columns and a number of rows. The columns are identified by a certain name and usually have an accepted data format. It is also advised that a table has one column as a unique key for each of the rows known as the primary key. Rows are the specific values for a particular record. Rows are also called tuples. Here are all the tables for this project and their columns and respective properties.

**companies**

id int(11) PRIMARY KEY,

company varchar(50) ,

contact\_person varchar(40) ,

email varchar(50) ,

phone varchar(20) ,

address varchar(50) ,

users\_limit int(11)

**users**

id int(11) PRIMARY KEY,

fullname varchar(40) ,

company int(11) ,

email varchar(20) ,

password varchar(200) ,

change\_password int(11)

**vehicles**

id int(11) PRIMARY KEY,

company int(11) ,

reg varchar(8) ,

vehicle varchar(50) ,

journey\_cost decimal(10,0)

**afc\_ride**

id int(11) PRIMARY KEY,

passenger int(11) ,

vehicle int(11) ,

in\_lat varchar(30) ,

in\_lon varchar(30) ,

out\_lat varchar(30) ,

out\_lon varchar(30) ,

in\_time timestamp DEFAULT CURRENT\_TIMESTAMP,

out\_time datetime ,

amount double

**bt\_registered**

id int(11) PRIMARY KEY,

card varchar(30) ,

owner varchar(50) ,

email varchar(50) ,

phone varchar(20) ,

pin int(4) ,

balance double

**bt\_statement**

id int(11) PRIMARY KEY,

record\_time timestamp DEFAULT CURRENT\_TIMESTAMP,

passenger int(11) ,

company int(11) ,

vehicle int(11) ,

amount double ,

balance double

**tr\_notifications**

id int(11) PRIMARY KEY,

company int(11) ,

vehicle int(11) ,

record\_time timestamp DEFAULT CURRENT\_TIMESTAMP,

event varchar(50) ,

value varchar(50)

**tr\_notifications\_events**

id int(11) PRIMARY KEY,

event varchar(50)

**tr\_on\_off\_status**

id int(11) PRIMARY KEY,

vehicle int(11) ,

record\_time timestamp DEFAULT CURRENT\_TIMESTAMP,

status int(11)

**tr\_tracking\_data**

id int(11) PRIMARY KEY,

vehicle int(11) ,

record\_time timestamp DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

latitude varchar(50) ,

longitude varchar(50) ,

mass double ,

speed double ,

on\_off\_status int(11) ,

idling int(11) ,

battery int(11)

## 4.6 Program Design

Program design also known as software is the process by which a software engineer creates the specification of a software project so as to accomplish the objectives of the project (Suryanarayana and Girish), 2014. It involves conceptualization, framing, implementing the software project artifact as desired by the user requirements. It is a very critical task before coding. The developer shall make use of the model view controller (MVC) design pattern. The MVC pattern is a best program for web applications. The view part is the one that the user interacts with either to input or get output from the application. User’s input is handled by the controller which sends requests to the model, which communicates with the database.

## 4.7 Interface design

An interface is the means by which a user and a computer system interact, that is exchange of input and output. Interface design is therefore a very important stage of this project as the interface determines the user friendliness of an application.

### 4.7.1 Input Forms

These are forms that mainly take input from the user of the system.

#### 4.7.1.1 Login Form

This form allows users to login to the system and if a user tries to access the system with no user session, it will be the first form they encounter.

### 4.7.2 Output Forms

These are forms that mainly take input from the user of the system.

#### 4.7.1.2 Operation Report

This form allows management to have a real-time view of how operations are happening.

## 4.8 Pseudo Code

Pseudo code is a piece of code written in plain English with the intention of making the programmer get the logic and sense of the desired code output.

**vehicle tracking device**

device visits https://sirika.mupini.co.zw/track-vehicle.php with the following data every 1s

record\_time

latitude

longitude

mass

speed

on\_off\_status

idling

battery

record the data as a new row in a tracking data table

alert any set abnormalities

**fare collection device**

passenger taps a card

get the identity of the passenger from the card id

detect whether passenger is dropping off or getting in to a bus

if getting into bus

record location

deduct an amount set as the minimum to deduct

if dropping off

record location

retrieve getting into bus location

find difference in distance between the two locations

multiply the cost\_per\_km and the distance to get the actual amount to be deducted

update the amount to be deducted

**web application login**

visit sirika.mupini.co.zw

if not logged in

redirect user to the login page

allow user to enter email and password

if either of the two or both the records are wrong

alert user

redirect to login page

if true

redirect user to admin panel

**register new company**

to be done by admin

register company details

**register new user**

confirm if max number of users has not yet been reached

if reached

alert user

else

allow entereing staff details

confirm they meet the set validations

register

## 4.9 Security Design

Security design is to set mechanisms to protect information, data and information systems against unauthorised access of modification. There are three different aspects to consider in security design and these are physical security, network security and operational security.

### 4.9.1 Physical Security

Physical security is the protection of hardware, software, networks and even personnel from physical events or actions that causes damage or loss. These include theft, flooding, fires and vandalism. Personnel shall be tasked to protect the AFC devices from an physical security threats and will be held accountable if any action or event lead to the damage or loss of a device.

### 4.9.2 Network Security

Network security refers to the identification of all threats that can attack the system network intentionally or unintentionally. Powerful coding mechanisms will be employed and robust code will be built to protect the system network.

## 4.10 Conclusion

The design phase outlined the construction of the proposed system in light of the analysis that had been carried out in chapter 3. Design elements that include system design, physical design, database design, interface design, program design and security design were covered and considered understandable. This gives a stepping stone into the implementation phase where the actual coding, testing, documentation and conversion of the project will take place.